



PART NUMBER  
**312596501**

VERSION NUMBER  
**4.1**

EDITION NUMBER  
**1**

# ASM

APPLICATION SOFTWARE MANAGER® (ASM) SOFTWARE

STORAGE AND ARCHIVE MANAGER

PRODUCT TYPE  
**SOFTWARE**



# StorageTek ASM Storage and Archive Management Guide

**Version 4.1**

Information contained in this publication is subject to change without notice.

We welcome your feedback. Please contact the Global Learning Solutions Feedback System at:

[GLSFS@StorTek.com](mailto:GLSFS@StorTek.com)

or

Global Learning Solutions  
Storage Technology Corporation  
One StorageTek Drive  
Louisville, CO 80028-3256  
USA

Please include the publication name, part number, and edition number in your correspondence if they are available.

### **Export Destination Control Statement**

These commodities, technology or software were exported from the United States in accordance with the Export Administration Regulations. Diversion contrary to U.S. law is prohibited.

### **Restricted Rights**

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) and (2) of the Commercial Computer Software - Restricted Rights at FAR 52.227-19 (June 1987), as applicable.

### **Limitations on Warranties and Liability**

Storage Technology Corporation cannot accept any responsibility for your use of the information in this document or for your use in any associated software program. You are responsible for backing up your data. You should be careful to ensure that your use of the information complies with all applicable laws, rules, and regulations of the jurisdictions in which it is used.

### **Proprietary Information Statement**

The information in this document, including any associated software program, may not be reproduced, disclosed or distributed in any manner without the written consent of Storage Technology Corporation.

Should this publication be found, please return it to StorageTek, One StorageTek Drive, Louisville, CO 80028-5214, USA. Postage is guaranteed.

**First Edition** (August 2004)

**Part Number** 312596501

**EC Number** 132167

This edition contains 204 pages.

StorageTek, the StorageTek logo, **Application Storage Manager™** is trademark or registered trademark of Storage Technology Corporation. Other products and names mentioned herein are for identification purposes only and may be trademarks of their respective companies.

©2004 by Storage Technology Corporation. All rights reserved.



# Contents

---

Organization	xvii
Additional Information	xviii
StorageTek's External Web Site	xviii
Customer Resource Center	xviii
e-Partners Site	xviii
Hardcopy Publications	xviii
Related Publications	xviii
<b>1: Overview</b>	<b>1</b>
Capabilities	1
Archiving	2
Releasing	2
Staging	2
Recycling	3
Storage Devices	3
Commands	4
User Commands	5
General System Administrator Commands	6
File System Commands	7
Automated Library Commands	9
Archiver Commands	10
Specialized Maintenance Commands	10
Site-Customizable Scripts	11
Application Programming Interface	12
Operational Utilities	12
<b>2: Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment<sup>13</sup></b>	
Conventions	13
Command Arguments	14
Terminology	15
Automated Library Operations	15
To Stop Removable Media Operations	16
To Start Removable Media Operations	16
To Turn On an Automated Library	17
To Turn Off an Automated Library	17
To Load a Cartridge Into an Automated Library	18
To Unload a Cartridge From a Drive	18

Labeling a Cartridge . . . . .	19
To Label or Relabel a Tape . . . . .	19
To Label or Relabel an Optical Disk . . . . .	20
To Audit a Volume . . . . .	21
To Audit an Automated Library (Direct-Attached Only) . . . . .	21
Using a Cleaning Cartridge . . . . .	22
To Reset the Number of Cleaning Cycles . . . . .	22
To Use a Cleaning Cartridge With a Barcode . . . . .	23
To Use a Cleaning Cartridge Without a Barcode . . . . .	24
To Clean a Tape Drive . . . . .	24
To Clear Media Errors . . . . .	25
To Remove a Stuck Cartridge From a Drive . . . . .	26
Catalog Operations, Importing Cartridges, and Exporting Cartridges . . . . .	28
Tracking Exported Media — The Historian . . . . .	29
Importing and Exporting from an Automated Library . . . . .	30
To Import a Cartridge from a Library With a Mailbox . . . . .	30
To Export a Cartridge from a Library With a Mailbox . . . . .	31
To Import a Cartridge from a Library Without a Mailbox . . . . .	31
To Export a Cartridge from a Library Without a Mailbox . . . . .	32
To Enable Load Notification . . . . .	32
Manually Loaded Drive Operations . . . . .	33
To Load a Cartridge . . . . .	33
To Unload a Cartridge . . . . .	33
To View a Library Catalog . . . . .	34
<b>3: Basic Operations for Libraries With Vendor-Specific Operational Procedures . . .</b>	<b>35</b>
ADIC/Grau Automated Libraries . . . . .	35
To Import a Cartridge . . . . .	36
To Export a Cartridge . . . . .	36
Fujitsu LMF Automated Libraries . . . . .	37
To Import a Cartridge . . . . .	37
To Export a Cartridge . . . . .	37
IBM 3584 UltraScalable Tape Libraries . . . . .	38
Importing Cartridges . . . . .	38
Cleaning Drives . . . . .	38
Partitioning . . . . .	39
To Remove a Cartridge . . . . .	39
IBM 3494 Libraries . . . . .	39
To Import a Cartridge . . . . .	39
To Export a Cartridge . . . . .	40
Sony Direct-Attached 8400 PetaSite Automated Libraries . . . . .	40
To Import Tapes . . . . .	40

Exporting Tapes .....	41
To Export a Tape Without Using the Mailbox Slots as Storage Slots .....	41
To Export a Tape Using Mailbox Slots as Storage Slots .....	42
How to Move a Cartridge to a Different Slot .....	42
Sony Network-Attached Automated Libraries .....	43
To Import a Cartridge .....	43
To Export a Cartridge .....	44
StorageTek ACSLS-Attached Automated Libraries .....	45
To Import Tapes .....	45
To Export Tapes Using a Mailbox .....	46
<b>4: Archiving .....</b>	<b>47</b>
Archiver – Theory of Operations .....	47
Archive Sets .....	48
Archiving Operations .....	49
Step 1: Identifying Files to Archive .....	50
Step 2: Composing Archive Requests .....	52
Step 3: Scheduling Archive Requests .....	53
Step 4: Archiving the Files in an Archive Request .....	55
Sample Default Output .....	56
Archiver Daemons .....	56
Archive Log Files and Event Logging .....	57
The <code>archiver.cmd</code> File .....	59
To Create or Modify an <code>archiver.cmd</code> File and Propagate Your Changes .....	59
The <code>archiver.cmd</code> File .....	60
An <code>archiver.cmd</code> File Example .....	61
The <code>archiver.cmd</code> Directives .....	62
Global Archiving Directives .....	63
The <code>archivemeta</code> Directive: Controlling Whether Metadata is Archived .....	63
The <code>archmax</code> Directive: Controlling the Size of Archive Files .....	64
The <code>bufsize</code> Directive: Setting the Archiver Buffer Size .....	64
The <code>drives</code> Directive: Controlling the Number of Drives Used for Archiving ..	65
The <code>examine</code> Directive: Controlling Archive Scans .....	66
The <code>interval</code> Directive: Specifying an Archive Interval .....	67
The <code>logfile</code> Directive: Specifying An Archiver Log File .....	68
To Back Up an Archiver Log File .....	68
The <code>notify</code> Directive: Renaming the Event Notification Script .....	68
The <code>ovflmin</code> Directive: Controlling Volume Overflow .....	69
The <code>wait</code> Directive: Delaying Archiver Startup .....	71
File System Directives .....	71
The <code>fs</code> Directive: Specifying the File System .....	71
Other File System Directives .....	71



Archive Set Assignment Directive .....	72
Assigning Archive Sets .....	72
File Size <i>search_criteria</i> : -access .....	74
File Size <i>search_criteria</i> : -minsize and -maxsize .....	74
Owner and Group <i>search_criteria</i> : -user and -group .....	75
File Name <i>search_criteria</i> Using Pattern Matching: -name <i>regex</i> .....	75
Release and Stage <i>file_attributes</i> : -release and -stage .....	77
Archive Set Membership Conflicts .....	78
Archive Copy Directives .....	79
Releasing Disk Space After Archiving: -release .....	80
Delaying Disk Space Release: -norelease .....	80
Setting the Archive Age .....	81
Unarchiving Automatically .....	81
Specifying More Than One Copy for Metadata .....	81
Archive Set Copy Parameters .....	82
Controlling the Size of Archive Files: -archmax .....	83
Setting the Archiver Buffer Size: -bufsize .....	83
Specifying the Number of Drives for an Archive Request: -drivemax, -drivem- in, and -drives .....	83
Maximizing Space on a Volume: -fillvsns .....	86
Specifying Archive Buffer Locks: -lock .....	86
Making Archive Copies of Offline Files: -offline_copy .....	86
Specifying Recycling .....	87
Associative Archiving: -join .....	87
Controlling Unarchiving .....	89
Controlling How Archive Files are Written: -tapenonstop .....	90
Reserving Volumes: -reserve .....	90
Setting Archive Priorities: -priority .....	93
Scheduling Archiving: -startage, -startcount, and -startsize ..	94
VSN Association Directives .....	96
VSN Pools Directives .....	98
Disk Archiving .....	99
Configuration Guidelines .....	100
Directives for Disk Archiving .....	101
To Enable Disk Archiving .....	102
Disk Archiving Examples .....	103
Example 1 .....	103
Example 2 .....	104
Example 3 .....	105
Archiver Examples .....	105
Example 1 .....	106
Example 2 .....	107

Example 3 .....	110
Example 4 .....	113
Archiver Guidelines .....	116
Troubleshooting the Archiver .....	117
Why Files Are Not Archiving .....	118
Additional Archiver Diagnostics .....	119
Why Files Are Not Releasing .....	120
<b>5: Releasing .....</b>	<b>121</b>
Releaser Overview .....	122
Theory of Operation .....	122
Definitions .....	122
Age .....	123
Candidate .....	123
Priority .....	124
Weight .....	124
Partial release .....	124
Partial Release and Partial Stage .....	124
System Administrator Option Summary .....	126
User Option Summary .....	127
The <code>releaser.cmd</code> File .....	128
Specifying Age-Related and Size-Related Release Priority Directives: <code>weight_age</code> , <code>weight_age_access</code> , <code>weight_age_modification</code> , and <code>weight_age_residence</code> .....	129
File Age .....	129
File Size .....	131
Specifying Directives for Individual File Systems: <code>fs</code> .....	132
Specifying Debugging Directives: <code>no_release</code> and <code>display_all_candidates</code> 132	
Specifying a Minimum Residence Time: <code>min_residence_age</code> .....	133
Specifying a Log File: <code>logfile</code> .....	133
Inhibiting Releasing for Rearchived Files: <code>rearch_no_release</code> .....	134
Adjusting the size of the Releaser Candidate List: <code>list_size</code> .....	135
The <code>archiver.cmd</code> File's Role in Releasing .....	135
Configuring the Releaser .....	136
Running the Releaser Manually .....	137
Troubleshooting the Releaser .....	137
<b>6: Staging .....</b>	<b>139</b>
The <code>stager.cmd</code> File .....	139
To Create or Modify a <code>stager.cmd</code> File and Propagate Your Changes .....	140
Specifying the Number of Drives .....	141
Setting the Stage Buffer Size .....	141

Specifying a Log File .....	142
Specifying the Number of Stage Requests .....	144
Example <code>stager.cmd</code> File .....	145
The <code>archiver.cmd</code> File's Role in Staging .....	145
Prioritizing Preview Requests Using the <code>preview.cmd</code> File .....	146
VSN and Age Directives (Global) .....	147
Watermark Directives (Global or File System Specific) .....	147
Calculating Total Preview Request Priority .....	149
How to Set Up a Preview Request Priority Scheme .....	149
Example 1: Enforcing Stage Requests .....	150
Example 2: Enforcing Archive Requests .....	151
Example 3: Prioritizing Requests by Media .....	151
Example 4: Complex Prioritization .....	151
<b>7: Recycling .....</b>	<b>153</b>
Recycler Overview .....	153
Recycling Directives .....	155
Specifying a Log File: the <code>logfile</code> Directive .....	155
Preventing Recycling: the <code>no_recycle</code> Directive .....	156
Specifying Recycling for an Entire Automated Library: the <code>Library</code> Directive .....	156
Configuring the Recycler .....	157
Step 1: Creating a <code>recycler.cmd</code> File (Optional) .....	158
Example <code>recycler.cmd</code> File .....	159
Step 2: Editing the <code>archiver.cmd</code> File (Optional) .....	161
Step 3: Running the Recycler .....	162
Step 4: Creating a <code>crontab</code> File for the Recycler (Optional) .....	163
Step 5: Removing <code>-recycle_ignore</code> and <code>ignore</code> Parameters .....	164
Step 6: Creating a <code>recycler.sh</code> File (Optional) .....	164
Troubleshooting the Recycler .....	164
<b>8: Upgrading the Hardware in Your Environment .....</b>	<b>167</b>
Adding Slots in an Automated Library .....	168
To Add Slots In a Library .....	168
Upgrading or Replacing a Library .....	169
To Replace or Upgrade a Library .....	169
Upgrading DLT Tape Drives .....	172
To Upgrade Tape Drives .....	173
<b>9: Advanced Topics .....</b>	<b>175</b>
Device Logging .....	175
When to Use the Device Log .....	176
Enabling the Device Log .....	176
To Enable the Device Log By Using the <code>samset(1M)</code> Command .....	177

Contents

To Enable the Device Log by Editing the <code>defaults.conf</code> File .....	177
Removable Media Files .....	178
To Create a Removable Media or Volume Overflow File .....	178
Segmented Files .....	180
Archiving .....	180
Disaster Recovery .....	181
System Error Facility Reporting .....	181
To Enable SEF Reporting .....	181
SEF Report Output .....	182
To Generate SEF Output .....	182
Managing the SEF Log File .....	185

# List of Figures

---

Figure 1. Output from the <code>archiver(1M) -l</code> Command	56
Figure 2. Archiver Log File Lines	57
Figure 3. <code>archiver.cmd</code> File Example	62
Figure 4. Archiver Log File Example	70
Figure 5. <code>sls(1M) -D</code> Command and Output	70
Figure 6. Archive Set Membership Directives	73
Figure 7. Archiving Directives that Prevent Archiving	74
Figure 8. Using the <code>-minsize</code> and <code>-maxsize</code> Directive Examples	75
Figure 9. Using the <code>-user</code> and <code>-group</code> Directive Examples	75
Figure 10. Regular Expression Example	76
Figure 11. Files not Archived (Assuming Directives Shown in <a href="#">Figure 10</a> )	76
Figure 12. Files Archived (Assuming Directives Shown in <a href="#">Figure 10</a> )	76
Figure 13. Example <code>archiver.cmd</code> File	77
Figure 14. Files Archived (Assuming Directives Shown in <a href="#">Figure 13</a> )	77
Figure 15. An <code>archiver.cmd</code> File With Possible Membership Conflicts	79
Figure 16. Corrected <code>archiver.cmd</code> File	79
Figure 17. An <code>archiver.cmd</code> File Using the <code>-release</code> Directive	80
Figure 18. An <code>archiver.cmd</code> File Using the <code>-norelease</code> Directive	80
Figure 19. An <code>archiver.cmd</code> File that Specifies the Archive Age	81
Figure 20. An <code>archiver.cmd</code> File that Specifies the Unarchive Age	81
Figure 21. An <code>archiver.cmd</code> File that Specifies Multiple Metadata Copies	81
Figure 22. Archive Set Copy Parameter Format	82
Figure 23. Formats for the <code>-drivemax</code> , <code>-drivemin</code> , and <code>-drives</code> Directives	84
Figure 24. Directives Used to Split an Archive Request Over Multiple Drives	85
Figure 25. Directives for Sorting an Archive Set	88
Figure 26. Directives to Control Unarchiving	89
Figure 27. Owner Forms for the <code>-reserve</code> Parameter	90
Figure 28. An <code>archiver.cmd</code> File With Reserved Volumes	91
Figure 29. Library Catalog Showing Reserved Volumes	91
Figure 30. Commands to Use to Display the Reserve Information	92
Figure 31. Reserving Volumes by Archive Set	92
Figure 32. VSN Specifications - Example 1	97
Figure 33. VSN Specifications - Example 2	97
Figure 34. VSN Specifications - Example 3	97
Figure 35. Example Showing VSN Pools	99
Figure 36. Format for the <code>-disk_archive</code> Parameter	101
Figure 37. Format for the <code>clients</code> and <code>endclients</code> Directive	102

Figure 38. The <code>diskvols.conf</code> File on <code>pluto</code> . . . . .	103
Figure 39. The <code>diskvols.conf</code> File on <code>mars</code> . . . . .	103
Figure 40. The <code>archiver.cmd</code> File on <code>pluto</code> . . . . .	103
Figure 41. A <code>diskvols.conf</code> File . . . . .	104
Figure 42. Directives in the <code>archiver.cmd</code> File that Pertain to Disk Archiving . . . . .	104
Figure 43. Output From <code>sls(1M)</code> . . . . .	104
Figure 44. The <code>diskvols.conf</code> File on <code>snickers</code> . . . . .	105
Figure 45. The <code>diskvols.conf</code> File on <code>mars</code> . . . . .	105
Figure 46. Directives in the <code>archiver.cmd</code> File that Pertain to Disk Archiving . . . . .	105
Figure 47. <code>archiver(1M) -lv</code> Example Output Part One . . . . .	106
Figure 48. <code>archiver(1M) -lv</code> Example Output Part Two . . . . .	106
Figure 49. <code>archiver(1M) -lv</code> Example Output Part Three . . . . .	107
Figure 50. <code>archiver(1M) -lv</code> Example Output Part Four . . . . .	107
Figure 51. <code>archiver(1M) -lv</code> Output Part One Showing the <code>archiver.cmd</code> File . . . . .	108
Figure 52. <code>archiver(1M) -lv</code> Output Part Two Showing Media and Drives . . . . .	108
Figure 53. <code>archiver(1M) -lv</code> Output Part Three Showing File System Organization . . . . .	109
Figure 54. <code>archiver(1M) -lv</code> Output Part Four Showing Archive Sets and Removable Media . . . . .	109
Figure 55. <code>archiver(1M) -lv -c</code> Command Output . . . . .	110
Figure 56. <code>archiver.cmd</code> File and Archiver Output . . . . .	113
Figure 57. Age Priority Directive Formats . . . . .	130
Figure 58. Priority Calculation . . . . .	131
Figure 59. <code>releaser.cmd</code> File Fragment . . . . .	131
Figure 60. <code>releaser.cmd</code> File . . . . .	131
Figure 61. Releaser Log File Example . . . . .	133
Figure 62. Mathematical Relationships From the <code>---after scan---</code> Line in <a href="#">Figure 61</a> . . . . .	134
Figure 63. Directives to Release the Largest Files First . . . . .	136
Figure 64. Directives to Release Oldest-Modified Files First . . . . .	136
Figure 65. <code>mcf</code> File Used in this Chapter's Examples . . . . .	140
Figure 66. Stager Log File Example . . . . .	143
Figure 67. Example <code>stager.cmd</code> File . . . . .	145
Figure 68. <code>wm_priority</code> Calculation . . . . .	148
Figure 69. Settings for Going Below the LWM . . . . .	149
Figure 70. Example <code>preview.cmd</code> File . . . . .	150
Figure 71. Directives . . . . .	152
Figure 72. The <code>preview.cmd</code> File . . . . .	152
Figure 73. A <code>recycler.cmd</code> File Example . . . . .	159
Figure 74. Sample Recycling Messages . . . . .	160
Figure 75. Disk Archiving Specifications in the <code>archiver.cmd</code> File . . . . .	161
Figure 76. Recycler Log File Example for Removable Media Cartridges . . . . .	162
Figure 77. Recycler Log File Example for Disk Archive Files . . . . .	163
Figure 78. Recycler Messages . . . . .	165

Figure 79. Example `mcf` File ..... 175  
Figure 80. The `devlog` File ..... 175  
Figure 81. `sef.output` Contents ..... 183

## List of Figures



# List of Tables

---

Table 1. Automated Library Daemons	4
Table 2. User Commands	5
Table 3. General System Administrator Commands	6
Table 4. File System Commands	7
Table 5. Automated Library Commands	9
Table 6. Archiver Commands	10
Table 7. Specialized Maintenance Commands	10
Table 8. Site-Customizable Scripts	11
Table 9. Operational Tools	12
Table 10. Command Arguments	14
Table 11. Terminology	15
Table 12. Arguments for <code>samcmd(1M) load</code>	18
Table 13. Arguments for <code>tplabel(1M)</code>	19
Table 14. Arguments for <code>odlabel(1M)</code>	20
Table 15. Arguments for <code>auditslot(1M)</code>	21
Table 16. Arguments for <code>chmed(1M)</code>	23
Table 17. Arguments for <code>chmed(1M)</code>	25
Table 18. Arguments for <code>auditslot(1M)</code>	26
Table 19. Arguments for <code>chmed(1M)</code>	27
Table 20. Arguments for <code>samexport(1M)</code>	31
Table 21. Arguments for the <code>import(1M) Command</code>	36
Table 22. Arguments for the <code>samexport(1M) Command</code>	36
Table 23. Arguments for the <code>import(1M) Command</code>	37
Table 24. Arguments for the <code>samexport(1M) Command</code>	38
Table 25. Arguments for the <code>move(1M) Command</code>	41
Table 26. Arguments for the <code>move(1M) Command</code>	43
Table 27. Arguments for the <code>import(1M) Command</code>	44
Table 28. Arguments for the <code>samexport(1M) Command</code>	44
Table 29. Arguments for the <code>import(1M) Command</code>	46
Table 30. Arguments for the <code>samexport(1M) Command</code>	46
Table 31. Archiver Log File Fields	58
Table 32. The <code>archiver.cmd</code> File Directive Units	61
Table 33. Arguments for the <code>archmax</code> Directive	64
Table 34. Arguments for the <code>bufsize</code> Directive	65
Table 35. Arguments for the <code>drives</code> Directive	66
Table 36. Values for the <code>examine</code> Directive's <i>method</i> argument	67
Table 37. Arguments for the <code>ovflmin</code> Directive	69

Table 38. Arguments for the Archive Set Assignment Directive .....	73
Table 39. <code>-access age</code> Suffixes .....	74
Table 40. <code>-minsize</code> and <code>-maxsize size</code> Suffixes .....	74
Table 41. The <code>-release</code> Option .....	78
Table 42. The <code>-stage</code> Directive's <i>attributes</i> .....	78
Table 43. Arguments for the Archive Set Copy Parameters .....	82
Table 44. Arguments for the <code>-drivemax</code> , <code>-drivemin</code> , and <code>-drives</code> Parameters .....	84
Table 45. Archive Set Example Split .....	85
Table 46. Values for the <code>-offline_copy</code> Directive's <i>method</i> argument .....	87
Table 47. Archive Set Form Examples .....	92
Table 48. Owner Set Form Examples .....	92
Table 49. File System Form Examples .....	93
Table 50. Archive Priorities .....	94
Table 51. The <code>-startage</code> , <code>-startcount</code> , and <code>-startsize</code> Directive Formats .....	95
Table 52. Arguments for the VSN Association Directive .....	96
Table 53. Arguments for the VSN Pools Directive .....	99
Table 54. Directory Structure Example .....	105
Table 55. Mount Options for Partial Release .....	126
Table 56. User Release Options .....	128
Table 57. Archive Set Assignment <i>directives</i> .....	135
Table 58. Arguments for the <i>drives</i> Directive .....	141
Table 59. Arguments for the <i>bufsize</i> Directive .....	142
Table 60. Keywords for the <i>event</i> Argument .....	143
Table 61. Stager Log File Fields .....	144
Table 62. Staging <i>directives</i> that can Appear in the <code>archiver.cmd</code> File .....	145
Table 63. Watermark Priority Directives .....	148
Table 64. Request Priority Example .....	150
Table 65. Recycling Methods and Media Types .....	154
Table 66. Arguments for the <code>no_recycle</code> Directive .....	156
Table 67. Library Directive <i>parameter</i> Values .....	157
Table 68. Archive Set Recycling Directives .....	161
Table 69. Arguments for the <code>request(1)</code> Command .....	179

# Preface

---

This manual, the StorageTek ASM Storage and Archive Management Guide, describes the storage and archive management software supported in the StorageTek ASM 4.1 release. The StorageTek ASM software automatically copies files from online disk to archive media. The archive media can consist of either online disk or removable media cartridges.

This manual is written for the system administrators who are responsible for configuring and maintaining StorageTek ASM software. You, the system administrator, are assumed to be knowledgeable about Solaris OS procedures, including creating accounts, performing system backups, and other basic Solaris system administration tasks.

## ■ Organization

This manual has nine chapters, and an index:

- Chapter 1** “Overview” provides overview information
- Chapter 2** Explains basic operations. The information in this chapter applies to most automated libraries and manually loaded devices.
- Chapter 3** Explains how to manage cartridges in libraries with operational instructions that pertain only to that kind of library. This chapter describes these libraries and their library-specific basic operational procedures.
- Chapter 4** Explains the archiving process
- Chapter 5** Explains the releasing process
- Chapter 6** explains the staging process
- Chapter 7** Explains the recycling process
- Chapter 8** describes upgrade procedures that are specific to StorageTek ASM environments.
- Chapter 9** Describes advanced topics in StorageTek ASM operations
- Index** The Index helps you locate information.
- Glossary**

## ■ Additional Information

StorageTek offers several methods for you to obtain additional information.

### StorageTek's External Web Site

StorageTek's external Web site provides marketing, product, event, corporate, and service information. The external Web site is accessible to anyone with a Web browser and an Internet connection.

The URL for the StorageTek external Web site is <http://www.storagetek.com>

### Customer Resource Center

StorageTek's CRC is a Web site that enables members to resolve technical issues by searching code fixes and technical documentation. CRC membership entitles you to other proactive services, such as HIPER subscriptions, technical tips, answers to frequently asked questions, addenda to product documentation books, and online product support contact information. Customers who have a current warranty or a current maintenance service agreement may apply for membership by clicking on the Request Password button on the CRC home page. StorageTek employees may enter the CRC through PowerPort.

The URL for the CRC is <http://www.support.storagetek.com>.

### e-Partners Site

StorageTek's e-Partners site is a Web site that provides information about products, services, customer support, upcoming events, training programs, and sales tools to support StorageTek's e-Partners. Access to this site, beyond the e-Partners Login page, is restricted. On the e-Partners Login page, StorageTek employees and current partners who do not have access can request a login ID and password and prospective partners can apply to become StorageTek resellers.

The URL for the e-Partners site is <http://members.storagetek.com>.

### Hardcopy Publications

You may order paper copies of publications listed on the CRC. Send e-mail to [DistrL@louisville.stortek.com](mailto:DistrL@louisville.stortek.com).

## ■ Related Publications

You can find additional information in the following publications:

<b>Publication</b>	<b>Part Number</b>
StorageTek ASM, ASM-QFS, and ASM/QFS- Standalone Disaster Recovery Guide	312520501
StorageTek QFS and StorageTek ASM File System Administration Guide	312593801
StorageTek QFS and StorageTek ASM Software Installation and Configuration Guide	312593901
StorageTek QFS and StorageTek ASM 4.1 Release Notes	312594001



# Overview

---

The StorageTek ASM environment provides a configurable file system with storage, archive management, and retrieval capabilities. The StorageTek ASM software archives files by copying the files from online disk cache to archive media. The archive media can consist of disk slices in another file system or it can consist of removable tape or magneto-optical cartridges in automated or manually loaded storage devices. In addition, the StorageTek ASM software automatically maintains online disk space at site-specified usage thresholds. It releases disk space associated with archived file data and restores the files to online disk when they are needed.

This chapter provides a technical overview of the StorageTek ASM components. The following topics are presented:

- [“Capabilities” on page 1](#)
- [“Storage Devices” on page 3](#)
- [“Commands” on page 4](#)

## ■ Capabilities

The StorageTek ASM environment includes a file system and the storage and archive management software. The StorageTek ASM-QFS environment includes the StorageTek QFS file system. Both file systems are high-performance UNIX file systems that reside in the server’s disk cache. The major difference between these file systems is that the StorageTek ASM-QFS file system offers more high-performance features, including a distributed shared file system. For more information about the file systems themselves, see the Storagetek QFS and StorageTek ASM File System Administration Guide.

The other components that reside in the StorageTek ASM environment are as follows:

- The archiver, which automatically copies online disk cache files to archive media. The archive media can consist of either online disk files or removable media cartridges.
- The releaser, which automatically maintains the file system’s online disk cache at site-specified percentage usage thresholds by freeing disk blocks occupied by eligible archived files.

- The stager, which restores file data to the disk cache. When a user or process requests file data that has been released from disk cache, the stager automatically copies the file data back to the online disk cache.
- The recycler, which clears archive volumes of expired archive copies and makes volumes available for reuse.

The following sections briefly describe each of these capabilities. You can find more information about these capabilities in subsequent chapters.

## Archiving

By default, the archiver automatically creates one archive copy of all files in a StorageTek ASM file system and it writes the archive copy to archive media. You can configure the archiver to create up to four archive copies on a variety of archive media. If a file is *segmented*, each segment is treated as a file, and each segment is archived separately. The archiving process is initiated after disk-based files match a site-definable set of selection criteria.

For more information about the archiver, see [“Archiving” on page 47](#). For more information about segmented files, see [“Segmented Files” on page 180](#).

## Releasing

*Releasing* is the process of freeing primary (disk) storage that is used by an archived file’s data. Two threshold values, both expressed as a percentage of total disk space, are used to manage online disk cache free space. These thresholds are the high water mark and the low water mark. When online disk consumption exceeds the high water mark, the system automatically begins releasing the disk space occupied by eligible archived files. Disk space occupied by archived file data is released until the low water mark is reached. Files are selected for release depending on the file’s size and age. Optionally, the first portion of a file can be retained on disk for speedy access and for masking staging delays. If a file has been archived in segments, portions of the file can be released individually. For more information about the releaser, see [“Releasing” on page 121](#).

## Staging

When a file whose data blocks have been released is accessed, the stager automatically stages the file or file segment data back to online disk cache. The read operation tracks along directly behind the staging operation, allowing the file to be immediately available to an application before the entire file is completely staged.

The StorageTek ASM software processes stage request errors automatically. If a stage error is returned, the system attempts to find the next available archive copy of the file. Stage errors that can be automatically processed include media errors, unavailability of media, unavailability of an automated



library, and others. For more information about staging, see [“Staging” on page 139](#).

## Recycling

As users modify files, archive copies associated with the old versions of these files are considered to be *expired* on their archive media. Such copies are no longer needed, so they can be purged from the system. The recycler identifies the archive volumes with the largest proportions of expired archive copies and preserves the nonexpired copies by moving them to separate volumes.

If a removable media volume contains only expired copies, you can take one of the following actions:

- You can relabel the volume for immediate reuse.
- You can export the volume to offsite storage as an historical record of file changes. You can use standard UNIX utilities to restore previous versions of files from expired archive copies.

The recycling process is transparent to end users as it relates to their data files. For more information about recycling, see [“Recycling” on page 153](#).

## ■ Storage Devices

The StorageTek ASM environment supports a wide variety of tape storage and magneto-optical devices. The automated libraries that StorageTek ASM supports can be divided into the following groups depending on how they are attached to the environment:

- A direct attachment. A direct-attached library is connected directly to the host system using a small computer system interface (SCSI). This can be either a direct connection or a Fibre Channel connection. For example, a direct attachment is used for StorageTek libraries. The StorageTek ASM system controls these libraries directly using the SCSI standard for automated libraries.
- A network attachment. The StorageTek ASM software can be configured as a client of the library's host system. The network-attached libraries include some of the StorageTek, ADIC/Grau, IBM, and Sony libraries. These libraries use a software package supplied by the vendor. In these cases, the StorageTek ASM software interfaces with the vendor software using a daemon specifically designed for the automated library.

Table 1 shows the daemons specific to various automated libraries.

**Table 1. Automated Library Daemons**

Daemon	Description
sam-robotd	Monitors the execution of robot control daemons. The sam-robotd daemon is started automatically by the sam-amld daemon.
sam-genericd	Controls direct-attached libraries and media changers. Also controls ADIC libraries through the DAS interface.
sam-stkd	Controls the StorageTek media changers through the ACSAPI interface.
sam-ibm3494d	Controls the IBM 3494 tape libraries through the Imcpd interface.
sam-sonyd	Controls the Sony network-attached automated libraries through the DZC-8000S interface.

For a list of supported storage devices, contact your StorageTek sales representative or your authorized service provider (ASP).

The relationships between the devices managed within the StorageTek ASM environment are defined in the master configuration file, `/etc/opt/SUNWsamfs/mcf`. The `mcf` file specifies the removable media devices, libraries, and file systems included in the StorageTek ASM environment. Each piece of equipment is assigned a unique equipment identifier in the `mcf` file. Entries in the `mcf` also define manually mounted archiving devices and automated library catalog files.

When possible, the system uses the standard Solaris disk and tape device drivers. For devices not directly supported in the Solaris operating system (OS), such as certain library and optical disk devices, special device drivers are included in the StorageTek ASM software packages.

## ■ Commands

The StorageTek QFS and StorageTek ASM environments consist of a file system, daemons, processes, various types of commands (user, administrator, and so on), and tools. This section describes the commands that are included in the StorageTek QFS and StorageTek ASM software distributions.

The StorageTek QFS and StorageTek ASM commands operate in conjunction with the standard UNIX file system commands. Some commands are specific to only one product. All the commands are documented in UNIX `man(1)` pages.

The Storagetek QFS and StorageTek ASM File System Administration Guide contains overview information for the daemons, but individual daemons are described throughout the documentation set where appropriate.

This section introduces the commands and indicates which commands you can use within the StorageTek QFS or StorageTek ASM file systems. See the man pages that are included in the software distribution for more information.

This section contains the following topics:

- [“User Commands” on page 5](#)
- [“General System Administrator Commands” on page 6](#)
- [“File System Commands” on page 7](#)
- [“Automated Library Commands” on page 9](#)
- [“Archiver Commands” on page 10](#)
- [“Specialized Maintenance Commands” on page 10](#)
- [“Site-Customizable Scripts” on page 11](#)
- [“Application Programming Interface” on page 12](#)
- [“Operational Utilities” on page 12](#)

## User Commands

By default, file system operations are transparent to the end user. Depending on your site practices, however, you might want to make some commands available to users at your site to fine-tune certain operations. [Table 2](#) summarizes these commands.

**Table 2. User Commands**

Command	Description	Used By
archive(1)	Archives files and sets archive attributes on files.	StorageTek ASM
release(1)	Releases disk space and sets release attributes on files.	StorageTek ASM
request(1)	Creates a removable media file.	StorageTek ASM
sdu(1)	Summarizes disk usage. The sdu(1) command is based on the GNU version of the du(1) command.	StorageTek QFS StorageTek ASM
segment(1)	Sets segmented file attributes.	StorageTek ASM
setfa(1)	Sets file attributes.	StorageTek QFS StorageTek ASM

**Table 2. User Commands (Continued)**

<b>Command</b>	<b>Description</b>	<b>Used By</b>
sfind(1)	Searches for files in a directory hierarchy. The sfind(1) command is based on the GNU version of the find(1) command and contains options for searching based on StorageTek QFS and StorageTek ASM file attributes.	StorageTek QFS StorageTek ASM
sls(1)	Lists contents of directories. The sls(1) command is based on the GNU version of the ls(1) command and contains options for displaying file system attributes and information.	StorageTek QFS StorageTek ASM
squota(1)	Reports quota information.	StorageTek QFS StorageTek ASM
ssum(1)	Sets the checksum attributes on files.	StorageTek ASM
stage(1)	Sets stage attributes on files and copies offline files to disk.	StorageTek ASM

## General System Administrator Commands

[Table 3](#) summarizes the commands that you can use to maintain and manage the system.

**Table 3. General System Administrator Commands**

<b>Command</b>	<b>Description</b>	<b>Used By</b>
samcmd(1M)	Executes one samu(1M) operator interface utility command.	StorageTek QFS StorageTek ASM
samd(1M)	Starts or stops robotic and removable media daemons.	StorageTek ASM

**Table 3. General System Administrator Commands (Continued)**

Command	Description	Used By
samexplorer(1M)	Generates a StorageTek QFS or StorageTek ASM diagnostic report script	StorageTek QFS StorageTek ASM
samset(1M)	Changes StorageTek ASM settings.	StorageTek ASM
samu(1M)	Invokes the full-screen, text-based operator interface. This interface is based on the curses(3CURSES) software library. The samu utility displays the status of devices and allows the operator to control automated libraries.	StorageTek QFS StorageTek ASM

## File System Commands

[Table 4](#) summarizes the commands that you can use to maintain the file system.

**Table 4. File System Commands**

Commands	Description	Used By
mount(1M)	Mounts a file system. The man page name for this command is mount_samfs(1M).	StorageTek QFS StorageTek ASM
qfsdump(1M) qfsrestore(1M)	Creates or restores a dump file containing the file data and metadata associated with a StorageTek QFS file system.	StorageTek QFS
sambcheck(1M)	Lists block usage for a file system.	StorageTek QFS StorageTek ASM
samchaid(1M)	Changes file admin set ID attribute. For use with quotas.	StorageTek QFS StorageTek ASM
samfsck(1M)	Checks and repairs metadata inconsistencies in a file system and reclaims allocated, but unused, disk space.	StorageTek QFS StorageTek ASM

**Table 4. File System Commands (Continued)**

<b>Commands</b>	<b>Description</b>	<b>Used By</b>
samfsconfig(1M)	Displays configuration information.	StorageTek QFS StorageTek ASM
samfsdump(1M) samfsrestore(1M)	Creates or restores a dump file of the metadata associated with a StorageTek ASM file system.	StorageTek ASM StorageTek ASM-QFS
samfsinfo(1M)	Displays information about the layout of a StorageTek QFS or StorageTek ASM file system.	StorageTek QFS StorageTek ASM
samfstyp(1M)	Determines the StorageTek QFS or StorageTek ASM file system type.	StorageTek QFS StorageTek ASM
samgrowfs(1M)	Expands a file system by adding disk devices.	StorageTek QFS StorageTek ASM
sammkfs(1M)	Initializes a new file system from disk devices.	StorageTek QFS StorageTek ASM
samncheck(1M)	Returns a full directory path name given the mount point and inode number.	StorageTek QFS StorageTek ASM
samquota(1M)	Reports, sets, or resets quota information.	StorageTek QFS StorageTek ASM
samquotastat(1M)	Reports on active and inactive file system quotas.	StorageTek QFS StorageTek ASM
samsharefs(1M)	Manipulates the StorageTek QFS shared file system configuration information.	StorageTek QFS
samtrace(1M)	Dumps the trace buffer.	StorageTek QFS StorageTek ASM
samunhold(1M)	Releases SANergy file holds.	StorageTek QFS StorageTek ASM
trace_rotate(1M)	Rotates trace files.	StorageTek QFS StorageTek ASM

## Automated Library Commands

[Table 5](#) summarizes the automated library commands that you can use to configure, initialize, and maintain the automated libraries and devices within the StorageTek ASM environment.

**Table 5. Automated Library Commands**

<b>Command</b>	<b>Description</b>
auditslot(1M)	Audits a single media cartridge slot within a specified automated library.
build_cat(1M)	Builds a media catalog file for an automated library. Optionally, can also populate the catalog file.
chmed(1M)	Sets or clears library catalog flags and values on a specific cartridge.
cleandrive(1M)	Requests that a tape drive be loaded with a cleaning tape.
dump_cat(1M)	Displays the content of a binary catalog file in various ASCII formats.
import(1M) samexport(1M)	Imports or exports cartridges from a library by placing it in the mailbox. For network-attached libraries, this command updates the library catalog, but it does not physically move cartridges.
samload(1M) unload(1M)	Loads or unloads a cartridge for a specified device.
move(1M)	Moves a cartridge from one slot to another.
odlabel(1M)	Labels optical disks for use with the StorageTek ASM system.
samdev(1M)	Adds /dev/samst logical device entries. Used to communicate automated library, optical disk, and tape drive information.
tplabel(1M)	Labels tapes for use with the StorageTek ASM system.

## Archiver Commands

[Table 6](#) summarizes the commands that control the archiver's actions within the StorageTek ASM environment.

**Table 6. Archiver Commands**

Command	Description
archiver(1M)	Evaluates the archiver commands file for syntax completeness and semantic accuracy.
archiver.sh(1M)	Logs exceptional archiver events.
showqueue(1M)	Displays the content of an archiver queue file.
reserve(1M)	Reserves and unreserves volumes.
unreserve(1M)	

## Specialized Maintenance Commands

[Table 7](#) summarizes the various maintenance commands you can use in a StorageTek ASM environment.

**Table 7. Specialized Maintenance Commands**

Command	Description
archive_audit(1M)	Generates a report of all archived files on each cartridge.
dmpshm(1M)	Dumps the shared memory segments.
exarchive(1M)	Manipulates (exchanges) archive copies.
itemize(1M)	Catalogs an optical disk.
rearch(1M)	Marks or unmarks archive entries to be rearchived.
unrearch(1M)	
sam-recycler(1M)	Reclaims space used by expired archive copies from archive media.
sam-releaser(1M)	Releases disk space from online disk cache file systems.
samdev(1M)	Creates symbolic links in the /dev/samst directory that point to the actual devices to be used by the StorageTek ASM file system. This command is similar in function to the UNIX makedev(1M) command.
samset(1M)	Changes or displays variables used in StorageTek ASM operations.



**Table 7. Specialized Maintenance Commands (Continued)**

<b>Command</b>	<b>Description</b>
set_admin(1M)	Adds or removes permission for an administrator group to execute administrator commands.
set_state(1M)	Sets the state of a StorageTek ASM device.
stageback.sh(1M)	Stages files from StorageTek ASM or StorageTek ASM QFS archive tapes
star(1M)	Creates tape archives and adds or extracts files. This is a GNU version of the tar(1) command, and it has been extended for use with the StorageTek ASM file system. You can use this command in a disaster recovery situation if you need to read data from archive tapes.
tapealert(1M)	Decodes TapeAlert events.
unarchive(1M)	Deletes archive entries for one or more files.
undamage(1M)	Marks an archive entry for one or more files or directories as undamaged.

## Site-Customizable Scripts

[Table 8](#) summarizes the site-customizable scripts that you can use to monitor and control the StorageTek ASM environment. By default, the software installs these scripts in `/opt/SUNWsamfs/examples`. You can move these scripts from `/opt/SUNWsamfs/examples` to `/etc/opt/SUNWsamfs/scripts` and modify them to perform the desired action for your site. For more information about these scripts, see their respective man pages.

**Table 8. Site-Customizable Scripts**

<b>Script</b>	<b>Description</b>
dev_down.sh(1M)	Sends email to root when a device is marked down or off.
load_notify.sh(1M)	Notifies the operator when the StorageTek ASM software requests a cartridge that resides outside the library.
log_rotate.sh(1M)	Rotates log files.
recover.sh(1M)	Recovers files archived after the last <code>samfsdump(1M)</code> was taken.

**Table 8. Site-Customizable Scripts (Continued)**

Script	Description
restore.sh(1M)	Restores files to their online or partially online status.
stageback.sh(1M)	Stages files from archive media.
tarback.sh(1M)	Reloads files from archive media.

## Application Programming Interface

You can use the application programming interface (API) to make file system requests from within a user application. The requests can be made locally or remotely to the machine on which the file system is running. The API consists of the `libsam` and `libsamrpc` libraries. These libraries contain library routines for obtaining file status; for setting archive, release, and stage attributes for a file; and for manipulating the library catalog of an automated library. The `sam-rpcd` remote procedure call daemon handles remote requests. To automatically start the `sam-rpcd` daemon, set `samrpc=on` in the `defaults.conf` file.

For more information about the API, see the `intro_libsam(3)` man page. This man page provides overview information for using the library routines in `libsam` and `libsamrpc`.

## Operational Utilities

Within the StorageTek ASM environment, you can use the `samu(1M)` operator utility and ASM QFS Manager to perform basic operations. [Table 9](#) summarizes the operational tools.

**Table 9. Operational Tools**

GUI Tools	Description
ASM QFS Manager	Provides a web-based graphical user interface to the StorageTek QFS and StorageTek ASM software. You can use this interface to configure, control, monitor, and reconfigure the components of your StorageTek QFS and StorageTek ASM environment. For information on installing ASM QFS Manager, see the StorageTek QFS and StorageTek ASM Software Installation and Configuration Guide. For information on using the ASM QFS Manager, see its online help.
<code>samu(1M)</code>	Provides the starting point for accessing the <code>samu(1M)</code> operator utility.

# Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment

## 2

---

An *automated library* is a robotically controlled device designed to load and unload removable cartridges without operator intervention. Cartridges are imported to and exported from the library. They are loaded and unloaded automatically. The archiving and staging processes use a site-defined scheme for allocating the number of drives to use. Automated libraries are also known as media changers, jukeboxes, robots, libraries, or media libraries.

The following sections describe aspects of using libraries in a StorageTek ASM environment. The StorageTek QFS and StorageTek ASM Software Installation and Configuration Guide provides initial configuration instructions, and this chapter provides operational instructions for automated libraries and manually loaded drives. In addition, this chapter describes the operator-oriented load notification facility that alerts an operator when a requested volume is not in a library.

**Note:** The StorageTek ASM software interoperates with automated libraries from many manufacturers. Contact StorageTek customer support for information pertinent to library model numbers, firmware levels, and other compatibility information.

Some automated libraries have features that cause certain operations to differ from the ones described in this chapter. To determine whether your automated library has additional, vendor-specific operating instructions when used in a StorageTek ASM environment, check [“Basic Operations for Libraries With Vendor-Specific Operational Procedures”](#) on page 35.

This chapter contains the following topics:

- [“Conventions”](#) on page 13
- [“Automated Library Operations”](#) on page 15
- [“Manually Loaded Drive Operations”](#) on page 33

## ■ Conventions

The procedures for performing the basic operations described in this chapter typically show how to use the `samcmd(1M)` command, the `samu(1M)` operator utility, and the following commands:

- `tplabel(1M)`
- `odlabel(1M)`
- `auditslot(1M)`
- `cleandrive(1M)`
- `chmed(1M)`
- `import(1M)`
- `set_state(1M)`
- `sameexport(1M)`

In many cases, however, there is more than one way to perform the task described. You can perform many of these tasks from within ASM QFS Manager, which is the web-based graphical user interface (GUI) to the StorageTek QFS and StorageTek ASM software. You can use this interface to configure, control, monitor, and reconfigure the components of your StorageTek QFS and StorageTek ASM environment. For information on installing ASM QFS Manager, see the StorageTek QFS and StorageTek ASM Software Installation and Configuration Guide. For information on using the ASM QFS Manager, see its online help.

## Command Arguments

Many of the commands accept a common set of arguments. [Table 10](#) shows these arguments.

**Table 10. Command Arguments**

Argument	Meaning
<i>eq</i>	The Equipment Ordinal of the device being addressed as defined in the mcf file. The device that is identified can be an automated library, a drive, or a file system.
<i>slot</i>	The number of a storage slot in an automated library as recognized in the library catalog.
<i>partition</i>	A side of a magneto-optical disk. The partition must be 1 or 2.
<i>media_type</i>	The media type. For a list of valid media types, see the mcf(4) man page.
<i>vsn</i>	The volume serial name assigned to the volume.

Some commands accept various combinations of arguments depending on your circumstances. For example, from the `samu(1M)` operator utility, the `load` command has the following two formats:

```
:load eq:slot
:load media_type.vsn
```

Note the following:

- The first form uses a colon (:) to separate *eq* and *slot*.
- The second form uses a period (.) to separate *media\_type* and *vsn*.

## Terminology

Certain terms used throughout this chapter might be new to you. [Table 11](#) shows some of the most commonly used terms and their meanings.

**Table 11. Terminology**

Term	Meaning
<i>Automated library</i>	An automated device for storing tape and optical cartridges.
<i>Cartridge</i>	A tape or magneto-optical cartridge. A magneto-optical cartridge can contain one or more volumes or partitions.
<i>Partition</i>	An entire tape or one side of a magneto-optical disk. A partition can contain only one volume.
<i>Volume</i>	A named area on a cartridge for storing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side. A volume serial name (VSN) identifies a volume.

## ■ Automated Library Operations

Several basic operations are essentially the same on all automated libraries. The following basic operations are explained in this section:

- [“To Start Removable Media Operations” on page 16](#)
- [“To Stop Removable Media Operations” on page 16](#)
- [“To Turn On an Automated Library” on page 17](#)
- [“To Turn Off an Automated Library” on page 17](#)
- [“To Load a Cartridge Into an Automated Library” on page 18](#)
- [“To Unload a Cartridge From a Drive” on page 18](#)
- [“Labeling a Cartridge” on page 19](#)
- [“To Audit a Volume” on page 21](#)
- [“To Audit an Automated Library \(Direct-Attached Only\)” on page 21](#)
- [“Using a Cleaning Cartridge” on page 22](#)

- [“To Clean a Tape Drive” on page 24](#)
- [“To Clear Media Errors” on page 25](#)
- [“To Remove a Stuck Cartridge From a Drive” on page 26](#)
- [“Catalog Operations, Importing Cartridges, and Exporting Cartridges” on page 28](#)
- [“To Enable Load Notification” on page 32](#)

## To Stop Removable Media Operations

It is possible to stop removable media operations and leave the StorageTek ASM system mounted. You might do this, for example, if you want to manually manipulate cartridges in a library. When operations are resumed, pending stages are reissued and archiving is resumed.

- To stop removable media operations, use the `samcmd(1M) idle` and `samd(1M) stop` commands.

Use these commands in the following formats:

```
samcmd idle eq
samd stop
```

For *eq*, enter the Equipment Ordinal of the equipment being addressed as defined in the `mcf` file. To idle the drives, enter a `samcmd idle eq` command for each *eq* configured in your `mcf` file.

You can also idle the drives by using the `samu(1M)` operator utility or by using ASM QFS Manager.

**Note:** The drives in your StorageTek ASM environment should be idled prior to issuing the `samd(1M) stop` command. This allows the archiver, stager, and other processes to complete current tasks. Failure to issue the `samd(1M) stop` command can cause unexpected results when archiving, staging, and other activities are resumed.

## To Start Removable Media Operations

Typically, removable media operations commence when a StorageTek ASM file system is mounted.

- To start removable media operations manually, without mounting any file systems, enter the `samd(1M) start` command.

Use this command in the following format:

```
# samd start
```

If removable media operations are already running when the preceding command is entered, the following message is generated:

```
SAM-FS sam-amld daemon already running
```

For more information about the `samd(1M)` command, see the `samd(1M)` man page.

## To Turn On an Automated Library

When a library is in the `on` state, it is under the control of the StorageTek ASM system and can proceed with general operations. When you turn on a library, the StorageTek ASM software performs the following actions:

- Queries the device regarding its internal state. It discovers where tapes are, whether or not barcodes are used, and so on.
- Updates the catalog and other internal structures.
- Use the `samcmd(1M) on` command to turn on an automated library.

Use this command in the following format:

```
samcmd on eq
```

For `eq`, specify the Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

## To Turn Off an Automated Library

Placing a library in the `off` state stops I/O operations and removes the automated library from StorageTek ASM control. No automatic movement of cartridges occurs. Note that the drives in the automated library remain in the `on` state. You might want to turn an automated library off to perform the following tasks:

- To stop StorageTek ASM operations for this automated library only.
- To power down the automated library.
- Use the `samcmd(1M) off` command to turn off an automated library.

Use this command in the following format:

```
samcmd off eq
```

For `eq`, specify the Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

## To Load a Cartridge Into an Automated Library

Loading a cartridge into a drive occurs automatically when a VSN is requested for archiving or staging. Loading refers to moving a cartridge from a storage slot to a drive and making it ready.

- Use the `samcmd(1M) load` command to manually load a cartridge.

You can use this command even if the drive is in `unavail` status. This command has the following two possible formats:

```
samcmd load eq:slot[:partition]
samcmd load media_type.vsn
```

**Table 12. Arguments for `samcmd(1M) load`**

Argument	Meaning
<i>eq</i>	The Equipment Ordinal of the drive being addressed as defined in the mcf file.
<i>slot</i>	The number of a storage slot in an automated library as recognized in the library catalog.
<i>media_type</i>	The media type. For a list of valid media types, see the <code>mcf(4)</code> man page.
<i>partition</i>	A side of a magneto-optical disk. The partition must be 1 or 2. This argument is not applicable to tape cartridges.
<i>vsn</i>	The volume serial name assigned to the volume.

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

When you manually load a cartridge, it is generally loaded in the next available drive in the library. If you want to make a drive unavailable for this purpose, use the `samu(1M)` utility's `:unavail` command or change the state of the device using ASM QFS Manager. You might do this, for example, during a disaster recovery operation or to analyze a tape.

## To Unload a Cartridge From a Drive

Unloading a cartridge occurs automatically when a volume is no longer needed. You can also manually unload a drive. Unloading refers to removing a cartridge from a drive.

- Use the `samcmd(1M) unload` command to manually load a cartridge.



This command can be used even if the drive is in `unavail` status. Use this command in the following format:

```
samcmd unload eq
```

For `eq`, specify the Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

## Labeling a Cartridge

The procedure for labeling a cartridge differs depending on whether you are labeling a tape or optical cartridge. The following two sections describe these procedures.

**CAUTION:** Labeling and relabeling a cartridge makes the data currently on the cartridge inaccessible to any software. Relabel a cartridge only if you are certain that you do not need the data that is stored on the cartridge.

### To Label or Relabel a Tape

The following `tplabel(1M)` command line format shows the options most commonly used when labeling or relabeling a tape:

```
tplabel [ -new | -old vsn ] -vsn vsn eq:slot
```

**Table 13. Arguments for `tplabel(1M)`**

Argument	Meaning
<code>vsn</code>	A volume serial name. If you are relabeling, the new VSN name can be identical to the old VSN name.
<code>eq</code>	The Equipment Ordinal of the automated library or manually loaded drive being addressed as defined in the <code>mcf</code> file.
<code>slot</code>	The number of a storage slot in an automated library as recognized in the library catalog. This argument is not applicable for manually loaded drives.

- To label a new tape, use the `tplabel(1M)` command.

Use this command in the following format:

```
tplabel -new -vsn vsn eq:slot
```

- To relabel an existing tape, use the `tplabel(1M)` command.

Use this command in the following format:

```
tplabel -old vsn -new -vsn vsn eq:slot
```

After issuing the command to label or relabel a tape, the tape is loaded and positioned, and the tape label is written. For more information about the `tplabel(1M)` command, see the `tplabel(1M)` man page.

You can also perform this task by using ASM QFS Manager.

## To Label or Relabel an Optical Disk

The following `odlabel(1M)` command line format shows the options most commonly used when labeling or relabeling an optical disk:

```
odlabel [ -new | -old vsn ] -vsn vsn eq:slot:partition
```

**Table 14. Arguments for `odlabel(1M)`**

Argument	Meaning
<i>vsn</i>	A volume serial name. If you are relabeling, the new VSN name can be identical to the old VSN name.
<i>eq</i>	The Equipment Ordinal of the automated library or manually loaded drive being addressed as defined in the mcf file.
<i>slot</i>	The number of a storage slot in an automated library as recognized in the library catalog. This argument is not applicable to manually loaded drives.
<i>partition</i>	A side of a magneto-optical disk. The partition must be 1 or 2. This argument is not applicable to tape cartridges.

- To label a new optical disk, use the `odlabel(1M)` command.

Use this command in the following format:

```
odlabel -new -vsn vsn eq:slot:partition
```

- To relabel an existing optical disk, use the `odlabel(1M)` command.

Use this command in the following format:

```
odlabel -old vsn -vsn vsn eq:slot:partition
```

After issuing the command to label or relabel an optical disk, the optical disk is loaded and positioned, and the optical disk label is written. For

more information about the `odlabel(1M)` command, see the `odlabel(1M)` man page.

You can also perform this task by using ASM QFS Manager.

## To Audit a Volume

Occasionally the reported space remaining on a tape or optical cartridge might need to be updated in the library catalog. The `auditslot(1M)` command loads the cartridge containing the volume, reads the label, and updates the library catalog entry for the slot.

- Use the `auditslot(1M)` command to audit a volume.

Use this command in the following format:

```
auditslot [-e] eq:slot[:partition]
```

**Table 15. Arguments for `auditslot(1M)`**

Argument	Meaning
<code>-e</code>	If the <code>-e</code> option is specified and the media is tape, the remaining space is updated. Otherwise, it is not changed.
<code>eq</code>	The Equipment Ordinal of the automated library or manually loaded drive being addressed as defined in the mcf file.
<code>slot</code>	The number of a storage slot in an automated library as recognized in the library catalog. This argument is not applicable to manually loaded drives.
<code>partition</code>	A side of a magneto-optical disk. The partition must be 1 or 2. This argument is not applicable to tape cartridges.

For more information about the `auditslot(1M)` command, see the `auditslot(1M)` man page.

You can also perform this task by using the `samu(1M)` utility's `:audit` command or by using ASM QFS Manager.

## To Audit an Automated Library (Direct-Attached Only)

**Note:** This task cannot be performed on a network-attached automated library.

A full audit loads each cartridge into a drive, reads the label, and updates the library catalog. A library should be audited in the following situations:

- After moving cartridges in the automated library without using StorageTek ASM commands.

- If you are in doubt about the status of the library catalog and would like to update it (for example, after a power outage).
- If you have added, removed, or moved cartridges in an automated library that has no mailbox.
- Use the `samcmd(1M) audit` command to perform a full audit on an automated library.

Use this command in the following format:

```
samcmd audit eq
```

For `eq`, specify the Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

You can also perform this task by using the `samu(1M)` utility's `:audit` command or by using ASM QFS Manager.

## Using a Cleaning Cartridge

The StorageTek ASM systems allow you to import a cleaning cartridge to clean tape drives. This procedure differs depending on whether or not the cleaning cartridge is barcoded. The following sections explain various aspects of using a cleaning cartridge.

Cleaning practices differ from manufacturer to manufacturer. If you are having trouble with this, see [“Basic Operations for Libraries With Vendor-Specific Operational Procedures” on page 35](#) to determine if specialized procedures are recommended for your equipment.

**Note:** This task cannot be performed on a network-attached automated library.

## To Reset the Number of Cleaning Cycles

Cleaning tapes are useful only for a limited number of cleaning cycles. You can view the number of remaining cycles with the `samu(1M)` utility's `:v` display or from within ASM QFS Manager.

The StorageTek ASM systems track the number of cleaning cycles used for each cleaning tape and eject the tape when the remaining cycles equal zero. For example, a DLT cleaning tape has 20 cycles, and an Exabyte cleaning tape has 10 cycles. Each time a cleaning tape is imported, the cleaning cycle is reset to the highest number of cycles for that type of tape.

If automatic cleaning is available on your system but all cleaning tapes in the automated library have a count of zero, the drive is set to off and a message is issued in the StorageTek ASM log.

- Use the `chmed(1M)` command to reset a cleaning tape with a count of zero.

Use this command in the following format:

```
chmed -count count media_type.vsn
```

**Table 16. Arguments for `chmed(1M)`**

Argument	Meaning
<i>count</i>	The number of cleaning cycles to which you want the cleaning tape reset.
<i>media_type</i>	The media type. For a list of valid media types, see the <code>mcf(4)</code> man page.
<i>vsn</i>	The volume serial name assigned to the volume.

## To Use a Cleaning Cartridge With a Barcode

If the cleaning cartridge is barcoded, you can import it using the `import(1M)` command.

1. Make sure that the cleaning cartridge has a barcode of `CLEAN` or starts with the letters `CLN`.
2. Use the `import(1M)` command to import the cleaning cartridge.

Use this command in the following format:

```
import eq
```

For *eq*, specify the Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

The StorageTek ASM system moves the cartridge from the mailbox to a storage slot and updates the library catalog for each cartridge. In addition, after this command is issued, the cleaning media flag is set, and the access count is set to the appropriate number of cleaning cycles, based on the media type. Each time the media is used to clean a drive, it decrements the access count.

For example, the following command imports a cleaning tape into the automated library that is numbered `50` in your `mcf` file:

```
# import 50
```

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

## To Use a Cleaning Cartridge Without a Barcode

If the cartridge is not barcoded, you must import it first. It does not become marked as a cleaning cartridge. Perform the following steps:

1. Import the cartridge using the `import(1M)` command.

Use this command in the following format:

```
import eq
```

For `eq`, specify the Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

2. Use the `chmed(1M)` command to change the type to a cleaning cartridge.

You must know the Equipment Ordinal of the automated library and the slot into which the cleaning cartridge is loaded.

In the following example command line, the automated library is Equipment Ordinal 50 and the cleaning cartridge is in slot 77:

```
# chmed +C 50:77
```

The preceding command changes the cartridge type to that of a cleaning cartridge.

3. Use `chmed(1M)` again to set the cleaning cycle count.

The following example command sets the count on the cartridge used in the preceding step:

```
# chmed -count 20 50:77
```

For more information about the `chmed(1M)` command, see the `chmed(1M)` man page.

## To Clean a Tape Drive

**Note:** The StorageTek ASM systems does not support automatic cleaning on network-attached libraries. You should use the vendor's library manager software for automatic cleaning.

The StorageTek ASM environment supports the use of cleaning tapes if cleaning tapes are supported by the hardware. If a tape drive requests cleaning, the system automatically loads a cleaning tape.

If your system uses barcoded labels, cleaning tapes must have a VSN of `CLEAN` or a VSN starting with the letters `CLN` in the barcode label. Alternatively, you can use the `chmed(1M)` command to mark a VSN as a

cleaning tape and set the count. Multiple cleaning tapes are allowed in a system.

**Note:** Certain drive errors can result in cleaning cartridges being loaded repeatedly until all cleaning cycles are consumed. You can prevent this situation by using the `chmed(1M)` command to limit the number of cleaning cycles on cleaning cartridges. For example:

```
# chmed -count 20 50:77
```

When automatic cleaning is not available and the system uses barcodes, perform the following procedure to request manually that a drive be cleaned:

- Use the `cleandrive(1M)` command.

Use this command in the following format:

```
cleandrive eq
```

For *eq*, specify the Equipment Ordinal of the automated library being addressed as defined in the `mcf` file. This is the drive to be loaded with the cleaning cartridge.

## To Clear Media Errors

When a hardware or software error is encountered on a cartridge, the StorageTek ASM system sets the *media error* flag in the VSN catalog. On any given cartridge that generates a *media error* signal, you can use the `chmed(1M)` command to clear the error, and you can attempt to use the cartridge. The *media error* flag is displayed in the `samu(1M)` utility's `v` display and in ASM QFS Manager.

1. Use the `chmed(1M)` command to clear the `media error` flag.

Use this command in the following format to clear the `media error` flag:

```
chmed -E media_type.vsn
```

**Table 17. Arguments for `chmed(1M)`**

Argument	Meaning
<i>media_type</i>	The media type. For a list of valid media types, see the <code>mcf(4)</code> man page.
<i>vsn</i>	The volume serial name assigned to the volume.

2. Issue the `auditslot(1M)` command to update the space remaining information.

Use this command in the following format:

```
auditslot -e eq:slot[:partition]
```

**Table 18. Arguments for `auditslot(1M)`**

Argument	Meaning
<code>-e</code>	If the <code>-e</code> option is specified and the media is tape, the remaining space is updated. Otherwise, it is not changed.
<code>eq</code>	The Equipment Ordinal of the automated library or manually loaded drive being addressed as defined in the <code>mcf</code> file.
<code>slot</code>	The number of a storage slot in an automated library as recognized in the library catalog. This argument is not applicable to manually loaded drives.
<code>partition</code>	A side of a magneto-optical disk. The partition must be 1 or 2. This argument is not applicable to tape cartridges.

For more information about the `auditslot(1M)` command, see the `auditslot(1M)` man page.

You can also perform this task by using the `samu(1M)` utility's `:audit` command or by using ASM QFS Manager.

## To Remove a Stuck Cartridge From a Drive

If a cartridge becomes stuck in a drive, follow these steps.

1. Use the `samcmd(1M) off` command to turn off the drives in the automated library.

Use this command in the following format:

```
samcmd off eq
```

For `eq`, specify the Equipment Ordinal of the drive being addressed as defined in the `mcf` file.

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

2. Use the `samcmd(1M) off` command to turn off the automated library.

Use this command in the following format:

```
samcmd off eq
```



For *eq*, specify the Equipment Ordinal of the library being addressed as defined in the `mcf` file.

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

3. Physically remove the cartridge from the drive.

Make sure you do not damage either the cartridge or the drive.

4. Use the `samcmd(1M) on` command to turn on the automated library and the drive.

Issue this command once for the drive and once for the library. Use this command in the following format:

```
samcmd on eq
```

For *eq*, specify the Equipment Ordinal of the library or drive being addressed as defined in the `mcf` file.

If the automated library performs an audit when it is turned on, you are done. If it does not, perform the next step.

5. If you put the cartridge back into its storage slot, adjust the library catalog to set the occupied flag for the damaged tape by using the `chmed(1M)` command.

Use this command in the following format:

```
chmed +o eq:slot
```

**Table 19. Arguments for `chmed(1M)`**

Argument	Meaning
<i>eq</i>	The Equipment Ordinal of the automated library or drive being addressed as defined in the <code>mcf</code> file.
<i>slot</i>	The number of a storage slot in a library as recognized in the library catalog. This argument is not applicable for manually loaded drives.

For more information about the `chmed(1M)` command, see the `chmed(1M) man` page.

If you keep the cartridge out, and you want to put it back in later, you must import the cartridge into the automated library.

## Catalog Operations, Importing Cartridges, and Exporting Cartridges

The physical addition (import) and removal (export) of cartridges from an automated library enables you to perform several functions, including the following:

- You can replace cartridges.
- You can relocate cartridges to off-site storage to use later for disaster recovery purposes. If this is your task, you can use the `-I` option on the `chmed(1M)` command to specify additional information such as the storage location of the cartridge.

When you import and export cartridges, you also update the library catalog. Within the StorageTek ASM system, you accomplish these tasks by using the `import(1M)` and `sameexport(1M)` commands. You can also perform these tasks by using ASM QFS Manager.

A library *catalog* is the central repository of all information that the StorageTek ASM environment needs for finding cartridges in an automated library. The library catalog file is a binary UFS-resident file that contains information about each slot in an automated library. The information in this file includes the one or more volume serial names (VSNs) associated with the cartridge stored in the slot, the capacity and space remaining on that cartridge, and flags indicating read-only, write-protect, recycling, and other status information for the cartridge.

The StorageTek ASM environment treats catalogs differently depending on how the automated library is attached to the server, as follows:

- If the automated library is direct-attached, the library catalog is a one-to-one mapping between library catalog entries and physical slots in the automated library. The first entry in the library catalog is for the first slot in the automated library. When a cartridge is needed, the system consults the library catalog to determine which slot contains the VSN, and it issues a command to load the cartridge from that slot into a drive.
- If the automated library is network-attached, the library catalog is not a direct mapping to the slots. It is a list of the VSNs known to be present in the automated library. When a cartridge is requested, the system sends a request to the vendor's software to load the VSN into a drive. The vendor's software locates the VSN's storage slot.

Each automated library handles cartridge import and export differently due to system characteristics and the vendor-supplied software. For example, on the ACL 4/52 library, you need to issue a `move` command to move cartridges into the import/export unit before exporting cartridges from the automated library.

**Note:** Network-attached automated libraries import and export cartridges using their own utilities, so the `import(1M)` and `sameexport(1M)` commands only update the library catalog entries used by the StorageTek ASM systems. If you have a network-attached library, see [“Basic Operations for Libraries With Vendor-Specific Operational Procedures” on page 35](#) for information about importing and exporting cartridges.

## Tracking Exported Media — The Historian

The StorageTek ASM historians keep track of cartridges exported from an automated library or a manually mounted device. The historian acts like a virtual library, but it has no defined hardware devices. Like an automated library, it is configured in the `mcf` file, has a catalog that records entries for all cartridges associated with it, can import and export cartridges, and appears in ASM QFS Manager as another automated library.

You can configure the historian in the `mcf` file by using a device type of `hy`. If you do not configure the historian in the `mcf` file, it is created as follows:

```
historian n+1 hy - on /var/opt/SUNWsamfs/catalog/historian
```

In the preceding entry, `n+1` is the last Equipment Ordinal in the `mcf` file plus 1. If you want to use a different Equipment Ordinal or path name for the catalog, you need only to define the historian in the `mcf`.

The historian library catalog is initialized with 32 entries when the historian first starts. Make sure that the catalog resides on a file system large enough to hold the entire catalog. Your site might want to track existing StorageTek ASM cartridges that have been exported from the library. In this case, you need to build a historian catalog from the existing cartridges as described in the `build_cat(1M)` man page.

The following two configuration directives in the `defaults.conf` file affect the behavior of the historian:

- If the `exported_media = unavailable` directive appears, then any cartridges exported from an automated library are flagged as unavailable to the historian. Requests for cartridges flagged as unavailable generate an EIO error.
- If the `attended = no` directive appears, it declares to the historian that no operator is available to handle load requests. Requests to load cartridges known to the historian, and not already loaded, generate an EIO error.

For more configuration information, see the `historian(7)` and `defaults.conf(4)` man pages.

## Importing and Exporting from an Automated Library

A *mailbox* is an area in an automated library for adding and removing cartridges from the automated library. The `import(1M)` command moves a cartridge from the mailbox to a storage slot. The `samexport(1M)` command moves the cartridge from a storage slot to the mailbox. For most libraries, if a cartridge is present in the mailbox at the time that the StorageTek ASM software is started, the software imports the cartridge automatically upon startup.

Importing and exporting practices differ from manufacturer to manufacturer. If you are having trouble with this, see [“Basic Operations for Libraries With Vendor-Specific Operational Procedures” on page 35](#) to determine if specialized procedures are recommended for your equipment.

The following sections describe importing and exporting cartridges:

- [“To Import a Cartridge from a Library With a Mailbox” on page 30](#)
- [“To Export a Cartridge from a Library With a Mailbox” on page 31](#)
- [“To Import a Cartridge from a Library Without a Mailbox” on page 31](#)
- [“To Export a Cartridge from a Library Without a Mailbox” on page 32](#)

### To Import a Cartridge from a Library With a Mailbox

To import cartridges into an automated library that uses a mailbox, follow these steps.

1. Open the mailbox using the manufacturer’s suggested operation.

There is usually a button near the mailbox. Sometimes the mailbox is a one-slot mailbox referred to as a *mail slot* in the vendor’s documentation.

2. Manually place the cartridge into the mailbox.
3. Close the mailbox.
4. Use the `import(1M)` command to import the cartridge.

Use this command in the following format:

```
import eq
```

For *eq*, specify the Equipment Ordinal of the library being addressed as defined in the `mcf` file.

The system moves the cartridge from the mailbox to a storage slot and updates the library catalog for each cartridge.

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

## To Export a Cartridge from a Library With a Mailbox

This procedure moves a cartridge from a storage slot to a mailbox or mail slot. To export (eject) cartridges from a library that uses a mailbox, follow these steps.

1. Use the `samexport(1M)` command to move a cartridge from a storage slot to the mailbox.

Use this command in one of the following formats:

```
samexport eq:slot
samexport media_type.vsn
```

**Table 20. Arguments for `samexport(1M)`**

Argument	Meaning
<i>eq</i>	The Equipment Ordinal of the automated library being addressed as defined in the <code>mcf</code> file.
<i>slot</i>	The number of a storage slot in an automated library as recognized in the library catalog.
<i>media_type</i>	The media type. For a list of valid media types, see the <code>mcf(4)</code> man page.
<i>vsn</i>	The volume serial name assigned to the volume.

You can also perform this step by using `samu(1M)` or ASM QFS Manager.

2. Open the mailbox or mail slot using the manufacturer's suggested operation.

There is usually a button near the mailbox.

## To Import a Cartridge from a Library Without a Mailbox

1. Use the `samcmd(1M)` `unload` command.

Use this command in the following format:

```
samcmd unload eq
```

For *eq*, specify the Equipment Ordinal of the library being addressed as defined in the `mcf` file.

Wait until the system completes its current task, sets the status to off, and transfers the current active catalog to the historian.

2. Unlock and open the door to the automated library.
3. Load cartridges into the available slots.

4. Close and lock the door to the automated library.

The automated library reinitializes and scans the cartridges in the library. The StorageTek ASM software updates the library catalog by adding the VSNs of the imported cartridges to the catalog. The automated library state is set to `on`.

## To Export a Cartridge from a Library Without a Mailbox

1. Use the `samcmd(1M) unload` command.

Use this command in the following format:

```
samcmd unload eq
```

For `eq`, specify the Equipment Ordinal of the library being addressed as defined in the `mcf` file.

Wait until the system completes its current task, sets the status to off, and transfers the current active catalog to the historian.

2. Unlock and open the door to the automated library.
3. Remove the cartridges from their respective slots.
4. Close and lock the door to the automated library.

The automated library reinitializes and scans the cartridges in the automated library. The system updates the library catalog with the VSNs of the cartridges currently in library slots. The VSNs of the removed cartridges are removed from the library catalog and are now recorded only in the historian file. The automated library state is set to `on`.

## To Enable Load Notification

The StorageTek ASM software requests cartridges to be loaded regularly to satisfy archiving and staging needs. If the request is for a cartridge that resides inside a library, the request is handled automatically. If the request is for a cartridge that resides outside the library, operator action is required. If enabled, the `load_notify.sh(1M)` script sends email when a cartridge needs to be obtained from outside the library.

1. Become superuser.
2. Use the `cp(1)` command to copy the load notification script from its installed location to its operable location.

For example:

```
# cp /opt/SUNWsamfs/examples/load_notify.sh /etc/opt/SUNWsamfs/  
scripts/load_notify.sh
```

- Use `more(1)` or another command to examine the `defaults.conf` file.

Make sure that the following directives are in the file and appear as follows:

- `exported_media=available`
- `attended=yes`

These directives are set by default. If the load notification capability is to be enabled, ensure that these directives have not been changed.

- Modify the `load_notify.sh` script to send notices to the operator.

By default, the script sends email to `root`, but it can be edited to send email to another person, to dial a pager, or to provide some other means of notification.

## ■ Manually Loaded Drive Operations

This section describes operations that differ if you have a manually loaded, standalone drive and not an automated library. Each manually loaded drive has its own one-slot library catalog.

### To Load a Cartridge

- To load a cartridge into a manually loaded device, place the cartridge in the drive according to the manufacturer's instructions.

The StorageTek ASM system recognizes that the cartridge is loaded, reads the label, and updates the manual, one-slot catalog. No further action is necessary.

### To Unload a Cartridge

- Use `samcmd(1M) idle` command to idle the drive.

This command ensures that no archive or stage processes are active. Use this command in the following format:

```
samcmd idle eq
```

For *eq*, specify the Equipment Ordinal of the drive being addressed as defined in the `mcf` file.

The drive switches from `idle` to `off` when all I/O activity is complete, and the tape ejects.

If this is a tape, the tape rewinds, and the cartridge is ready to be removed. An optical cartridge ejects automatically. See the manufacturer's instructions for removing the specific cartridge.

You can also perform this task by using `samu(1M)` or ASM QFS Manager.

## To View a Library Catalog

- Use the `samu(1M)` utility's `:v` command.

Use this command in the following format:

```
:v eq
```

For *eq*, specify the Equipment Ordinal of the library being addressed as defined in the `mc f` file.



# Basic Operations for Libraries With Vendor-Specific Operational Procedures

## 3

---

You can include libraries from many different manufacturers in a StorageTek ASM environment. For most libraries, you should use the operational procedures described in [“Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment”](#) on page 13. Some libraries, however, have vendor-specific operational procedures, and these are described in this chapter.

**Note:** The StorageTek ASM software is compatible with automated libraries from many manufacturers. Consult the your StorageTek sales representative or your authorized service provider for information pertinent to library model numbers, firmware levels, and other compatibility information.

This chapter describes the following automated libraries:

- [“ADIC/Grau Automated Libraries”](#) on page 35
- [“Fujitsu LMF Automated Libraries”](#) on page 37
- [“IBM 3584 UltraScalable Tape Libraries”](#) on page 38
- [“IBM 3494 Libraries”](#) on page 39
- [“Sony Direct-Attached 8400 PetaSite Automated Libraries”](#) on page 40
- [“Sony Network-Attached Automated Libraries”](#) on page 43
- [“StorageTek ACSLS-Attached Automated Libraries”](#) on page 45

## ■ ADIC/Grau Automated Libraries

If you have an ADIC/Grau automated library, use the procedures in this section for importing and exporting cartridges. These procedures differ from those described in [“Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment”](#) on page 13.

Because you use vendor-supplied utilities to physically add and remove cartridges in the ADIC/Grau automated library, the StorageTek ASM interface (`import(1M)`, `samexport(1M)`, and ASM QFS Manager) affects only the library catalog.

## To Import a Cartridge

To import a cartridge, perform the following steps.

1. Use ADIC/Grau commands to physically move the cartridge into the library.
2. Use the StorageTek ASM `import(1M)` command to update the library catalog.

Use this command in the following format:

```
import -v volser eq
```

**Table 21. Arguments for the `import(1M)` Command**

Argument	Meaning
<i>volser</i>	The <i>volser</i> to be added. The grauaci interface verifies that the ADIC/Grau automated library has the <i>volser</i> information before updating the library catalog with the new entry.
<i>eq</i>	The Equipment Ordinal of the device being addressed as defined in the mcf file.

## To Export a Cartridge

To export a cartridge, perform the following steps.

1. Use the StorageTek ASM `samexport(1M)` command to remove the entry from the library catalog.

Use this command in one of the following formats:

```
samexport eq:slot  
samexport media_type.vsn
```

**Table 22. Arguments for the `samexport(1M)` Command**

Argument	Meaning
<i>eq</i>	The Equipment Ordinal of the device being addressed as defined in the mcf file.
<i>slot</i>	The number of a storage slot in an automated library as recognized in the library catalog.
<i>media_type</i>	The media type. For a list of valid media types, see the mcf(4) man page.
<i>vsn</i>	The volume serial name assigned to the volume.

The `samexport(1M)` command updates the library catalog as each VSN is exported, and it moves the library catalog entry for each VSN from the library catalog to the historian.

2. Use ADIC/Grau commands to physically move the cartridge out of the library.

## ■ Fujitsu LMF Automated Libraries

If you have a Fujitsu LMF automated library, use the procedures in this section for importing and exporting cartridges. These procedures differ from those described in [“Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment”](#) on page 13.

Because you use vendor-supplied utilities to physically add or remove cartridges in the Fujitsu LMF automated library, the StorageTek ASM interface (`import(1M)`, `samexport(1M)`, and ASM QFS Manager) affects only the library catalog.

### To Import a Cartridge

To import a cartridge, perform the following steps.

1. Use Fujitsu commands to physically move the cartridge into the library.
2. Use the StorageTek ASM `import(1M)` command to update the library catalog.

Use this command in the following format:

```
import -v volser eq
```

**Table 23. Arguments for the `import(1M)` Command**

Argument	Meaning
<i>volser</i>	The <i>volser</i> to be added. The <code>fujitsulmf</code> interface verifies that the LMF automated library has the <i>volser</i> information before updating the library catalog with the new entry.
<i>eq</i>	The Equipment Ordinal of the device being addressed as defined in the <code>mcf</code> file.

### To Export a Cartridge

To export a cartridge, perform the following steps.

1. Use the StorageTek ASM `samexport(1M)` command to remove the entry from the library catalog.

Use this command in one of the following formats:

```
samexport eq:slot
samexport media_type.vsn
```

**Table 24. Arguments for the `samexport(1M)` Command**

Argument	Meaning
<code>eq</code>	The Equipment Ordinal of the device being addressed as defined in the mcf file.
<code>slot</code>	The number of a storage slot in an automated library as recognized in the library catalog.
<code>media_type</code>	The media type. For a list of valid media types, see the mcf(4) man page.
<code>vsn</code>	The volume serial name assigned to the volume.

The `samexport(1M)` command updates the library catalog as each VSN is exported, and it moves the library catalog entry for each VSN from the StorageTek ASM library catalog to the StorageTek ASM historian.

2. Use Fujitsu commands to physically move the cartridge out of the library.

## ■ IBM 3584 UltraScalable Tape Libraries

The IBM 3584 UltraScalable Tape Libraries are supported in the StorageTek ASM environments. The following sections describe aspects of this library's operations that differ from the procedures described in ["Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment"](#) on page 13.

### Importing Cartridges

When the StorageTek ASM software is started, cartridges that are in the mailbox are not automatically imported.

### Cleaning Drives

To use this library in a StorageTek ASM environment, disable automatic cleaning and enable hosted cleaning. This process is described in the *IBM 3584 UltraScalable Tape Library Planning and Operator Guide*, IBM publication GA32-0408-01. This is also described in the `ibm3584(7)` man page.

## Partitioning

This library accommodates several tape drives. If you are using multiple drives, it is possible to divide this one physical library into two, three, or four logical libraries. If you have divided your library into two or more logical libraries, be sure that these logical libraries are operating properly before you add the IBM 3584 library to the StorageTek ASM environment.

When a cartridge is exported from a partitioned library, only the logical library from which it was exported can access that drawer slot. If the cartridge is removed and reinserted manually, it is accessible to any and all logical partitions.

### To Remove a Cartridge

The following steps describe the act of removal as used in this situation:

1. Open the door.
2. Remove the cartridge(s).
3. Close the door.
4. Wait for the door to lock and then unlock.
5. Open the door.
6. Replace the cartridge(s).
7. Close the door.

For more information on using this library as a logically partitioned library in a StorageTek ASM environment, see your IBM documentation or the [i.bm3584\(7\) man page](#).

## ■ IBM 3494 Libraries

The IBM 3494 libraries are supported in the StorageTek ASM environments. The following sections describe aspects of this library's operations that differ from the procedures described in [“Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment” on page 13](#).

### To Import a Cartridge

To import a cartridge, perform the following steps.

1. Place the new media into the I/O slots.
2. Close the door.

The library locks the door and moves the media into the storage area. You can import only 100 volumes at one time.

If the library is configured with `access=private`, this is the last step you need to perform. The library informs the daemon as the media is moved, and the media is added to the catalog.

3. Use the `import(1M)` command to add the media to the catalog. (Optional)

Perform this step only if the library is configured with `access=shared`.

If the library is configured with `access=shared`, issue the `import(1M)` command to add the media to the catalog.

## To Export a Cartridge

1. Use the `export(1M)` command to export cartridges.

This command moves the media to the I/O area and turns on the output mode light. on the operator panel.

2. Physically remove the media from the I/O area.

## ■ Sony Direct-Attached 8400 PetaSite Automated Libraries

The Sony 8400 PetaSite Series automated library is different from other Sony models because it has an eight-slot import and export mailbox (slots 400–407). For this reason, the import and export operations are more straightforward for this system. This automated library uses a barcode reader.

Because the mailbox slots can be used as storage slots, the StorageTek ASM library catalog keeps track of the mailbox slots.

**Note:** The information in this section applies only to Sony direct-attached 8400 PetaSite automated libraries. This information does not pertain to the Sony direct- attached B9 and B35 automated libraries, nor does it pertain to the [“Sony Network-Attached Automated Libraries” on page 43.](#)

## To Import Tapes

To import tapes, follow these steps.

1. Open the door of the automated library by pushing the open/close button on the front panel of the automated library.
2. Load the cartridges into the mailbox slots.
3. Push the open/close button on the front panel of the automated library and manually close the door to the mailbox.

The automated library checks the mailbox slots for the cartridge barcodes after the door is closed. If there is a problem with the barcodes, both the `in` and `out` lights flash for that slot.

4. Use the `import(1M)` command to enable the StorageTek ASM system to recognize the imported cartridges.

Use this command in the following format:

```
import eq
```

For `eq`, specify the Equipment Ordinal of the device being addressed as defined in the `mcf` file.

You can also perform this step by using ASM QFS Manager.

## Exporting Tapes

The procedure for exporting tape cartridges differs depending on whether or not you are using the mailbox slots as storage slots.

### To Export a Tape Without Using the Mailbox Slots as Storage Slots

Use the following procedure to export a cartridge when you are not using the mailbox slots as storage slots.

1. Issue the `move(1M)` command to move the cartridge to a mailbox slot (slots 400-407).

Use this command in the following format:

```
move source_slot destination_slot eq
```

**Table 25. Arguments for the `move(1M)` Command**

Argument	Meaning
<code>source_slot</code>	The number of the slot in which the cartridge currently resides.
<code>destination_slot</code>	The number of the slot into which the cartridge should be moved.
<code>eq</code>	The Equipment Ordinal of the device being addressed as defined in the <code>mcf</code> file.

2. Push the open/close button on the front panel of the automated library.  
The door opens.

3. Remove the cartridge from the mailbox slot.
4. Push the open/close button on the front panel of the automated library and manually close the door to the mailbox.
5. Issue the `samexport(1M)` command to enable the StorageTek ASM system to recognize the exported cartridge.

Use this command in the following format:

```
samexport eq
```

For *eq*, specify the Equipment Ordinal of the device being addressed as defined in the `mcf` file.

You can also perform this step by using ASM QFS Manager.

## To Export a Tape Using Mailbox Slots as Storage Slots

Use the following procedure to export a cartridge when you are using the mailbox slots as storage slots and the cartridge you want to export is in one of the mailbox slots.

1. Push the open/close button on the front panel of the automated library.  
The door opens.
2. Remove the cartridge from the mailbox slot.
3. Push the open/close button on the front panel of the automated library and manually close the mailbox door.
4. Issue the `samexport(1M)` command to enable the StorageTek ASM system to recognize the exported cartridge.

Use this command in the following format:

```
samexport eq
```

For *eq*, specify the Equipment Ordinal of the device being addressed as defined in the `mcf` file.

You can also perform this step by using ASM QFS Manager.

## How to Move a Cartridge to a Different Slot

To move a cartridge to a different slot, follow these steps:

1. Make sure that the source slot is occupied and that the destination slot is empty.
2. Issue the `move(1M)` command.



Use this command in the following format:

```
move eq:source_slot destination_slot
```

**Table 26. Arguments for the `move(1M)` Command**

Argument	Meaning
<code>eq</code>	The Equipment Ordinal of the device being addressed as defined in the mcf file.
<code>source_slot</code>	The number of the slot in which the cartridge currently resides.
<code>destination_slot</code>	The number of the slot into which the cartridge should be moved.

You can also perform this step by using ASM QFS Manager.

## ■ Sony Network-Attached Automated Libraries

If you have a Sony network-attached automated library, use the procedures in this section for importing and exporting cartridges. These procedures differ from those described in [“Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment”](#) on page 13.

Because you use vendor-supplied utilities to physically add and remove cartridges in the Sony automated library, the StorageTek ASM interfaces (`import(1M)`, `samexport(1M)`, and ASM QFS Manager) affect only the library catalog.

### To Import a Cartridge

To import a cartridge, perform the following steps.

1. Use Sony commands to physically move the cartridge into the library.
2. Use the `import(1M)` command to update the library catalog.

Use this command in the following format:

```
import -v [ " ] volser [ " ] eq
```

**Table 27. Arguments for the `import(1M)` Command**

Argument	Meaning
" "	Quotation marks. The <i>volser</i> must be enclosed in quotation marks if it contains spaces.
<i>volser</i>	The <i>volser</i> to be added. The PSC API interface verifies that the Sony automated library has the <i>volser</i> information before updating the library catalog with the new entry. If the cartridge does not physically exist in the library, the entry is placed in the historian catalog.
<i>eq</i>	The Equipment Ordinal of the library being addressed as defined in the <i>mcf</i> file.

## To Export a Cartridge

To export a cartridge, perform the following steps.

1. Use the `samexport(1M)` command to remove the entry from the library catalog.

Use this command in one of the following formats:

```
samexport eq:slot
samexport media_type.vsn
```

**Table 28. Arguments for the `samexport(1M)` Command**

Argument	Meaning
<i>eq</i>	The Equipment Ordinal of the device being addressed as defined in the <i>mcf</i> file.
<i>slot</i>	The number of a storage slot in an automated library as recognized in the library catalog.
<i>media_type</i>	The media type. For a list of valid media types, see the <i>mcf(4)</i> man page.
<i>vsn</i>	The volume serial name assigned to the volume.

The `samexport(1M)` command updates the library catalog as each VSN is exported, and it moves the library catalog entry for each VSN from the library catalog to the historian.

2. Use Sony commands to physically move the cartridge out of the library.

## ■ StorageTek ACSLS-Attached Automated Libraries

If you have a StorageTek ACSLS-attached automated library, use the procedures in this section for importing and exporting cartridges. These procedures differ from those described in [“Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment”](#) on page 13.

A *mailbox* is an area used for putting cartridges into and removing cartridges from the automated library. Some StorageTek automated libraries import and export only one cartridge at a time. Examples of StorageTek automated libraries with a mailbox that are supported within the StorageTek ASM environment includes the StorageTek 9714 and the StorageTek 9710. The StorageTek 9730 uses a mailslot. In StorageTek documentation, the mailbox and mailbox slot are often referred to as the *CAP*.

When importing and exporting cartridges from an ACSLS-attached automated library, be aware of the following:

- When importing cartridges, StorageTek ASM commands affect only the library catalog. The `import(1M)` command does not insert cartridges into the automated library physically. You must use ACSLS commands to physically import cartridges.
- When exporting cartridges StorageTek ASM commands affect only the library catalog unless you also use the `-f` option on the `samexport(1M)` command. Using the `-f` option directs the StorageTek ASM system to put the volume in the Cartridge Access Port (CAP) and to update the catalog accordingly. If you do not specify the `-f` option, the catalog is updated, but because the volume is not put in the CAP, you must still use ACSLS commands to physically export the cartridges.

It is your responsibility to keep the ACSLS inventory and the StorageTek ASM catalog in agreement.

You can also perform the import and export procedures by using `samu(1M)` or ASM QFS Manager.

### To Import Tapes

- To import tape cartridges, use the `import(1M)` command.

Use this command in the following format:

```
import -v vsn eq
```

**Table 29. Arguments for the `import(1M)` Command**

Argument	Meaning
<i>vsn</i>	The volume serial name assigned to the volume.
<i>eq</i>	The Equipment Ordinal of the device being addressed as defined in the mcf file.

The `import(1M)` command causes the new VSN to appear in the library catalog. If the VSN was in the historian, the StorageTek ASM software moves the VSN information from the historian to the library catalog.

## To Export Tapes Using a Mailbox

You can export tape cartridges by slot or by VSN.

- To export tape cartridges, use the `samexport(1M)` command.

Use this command in one of the following formats:

```
samexport [-f] eq:slot
samexport [-f] media_type.vsn
```

**Table 30. Arguments for the `samexport(1M)` Command**

Argument	Meaning
<code>-f</code>	Directs the StorageTek ASM system to put the volume in the Cartridge Access Port (CAP) and to update the catalog accordingly.
<i>eq</i>	The Equipment Ordinal of the device being addressed as defined in the mcf file.
<i>slot</i>	The number of a storage slot in an automated library as recognized in the library catalog.
<i>media_type</i>	The media type. For a list of valid media types, see the <code>mcf(4)</code> man page.
<i>vsn</i>	The volume serial name assigned to the volume.

The `samexport(1M)` command updates the library catalog as each VSN is exported, and it moves the library catalog entry for each VSN from the library catalog to the historian.

# Archiving

---

*Archiving* is the process of copying a file from a StorageTek ASM file system to a volume that resides on a removable media cartridge or on a disk partition of another file system. Throughout this chapter, the term *archive media* is used to refer to the various cartridges or disk slices to which archive volumes are written. The StorageTek ASM archiving capabilities include many features, such as those that you can use to specify that files be archived immediately, to specify that files never be archived, and to perform other tasks.

This chapter describes the archiver's theory of operations, provides general guidelines for developing archive policies for your site, and explains how to implement policies by creating an `archiver.cmd` file.

The following topics are presented:

- [“Archiver – Theory of Operations” on page 47](#)
- [“The `archiver.cmd` File” on page 59](#)
- [“The `archiver.cmd` Directives” on page 62](#)
- [“Disk Archiving” on page 99](#)
- [“Archiver Examples” on page 105](#)
- [“Archiver Guidelines” on page 116](#)
- [“Troubleshooting the Archiver” on page 117](#)

## ■ Archiver – Theory of Operations

The archiver automatically writes StorageTek ASM files to archive media. Operator intervention is not required to archive and stage the files. Files are archived to a volume on the archive media, and each volume is identified by a unique identifier called a *volume serial name* (VSN). Archive media can contain one or more volumes. To identify an individual volume, the media type and VSN must be specified.

The archiver starts automatically when a StorageTek ASM file system is mounted. You can customize the archiver's operations for your site by inserting archiving directives into the following file:

```
/etc/opt/SUNWsamfs/archiver.cmd
```

The `archiver.cmd` file does not need to be present for archiving to occur. In the absence of this file, the archiver uses the following defaults:

- All files are archived to available volumes.
- The *archive age* for all files is 4 minutes. The archive age is the amount of time since a file's last modification.
- The *archive interval* is 10 minutes. The archive interval is the amount of time that elapses between complete archiving processes.

The following sections describe the concept of an archive set and explain the operations performed during the archiving process.

## Archive Sets

An *archive set* identifies a group of files to be archived. Archive sets can be defined across any group of file systems. Files in an archive set share common criteria that pertain to the size, ownership, group, or directory location. The archive sets control the destination of the archive copy, how long to keep the copy archived, and how long to wait before archiving the data. All files in an archive set are copied to the volumes associated with that archive set. A file in the file system can be a member of one and only one archive set.

As files are created and modified, the archiver copies them to archive media. Archive files are compatible with the standard UNIX `tar(1)` format. This ensures data compatibility with the Sun Solaris operating system (OS) and other UNIX systems. This format includes the file access data (inode information) and the path to the file. If a complete loss of your StorageTek ASM environment occurs, the `tar(1)` format allows file recovery using standard UNIX tools and commands. The archiving process also copies the data necessary for StorageTek ASM file system operations. This data consists of directories, symbolic links, the index of segmented files, and archive media information.

In the remainder of this section, the term *files* refers to both file data and metadata. The terms *file data* and *metadata* are used only when a distinction is required. The term *file system* refers to a mounted StorageTek ASM file system.

Archive set names are determined by the administrator and are virtually unlimited with the following exceptions:

- There are two reserved archive set names: `no_archive` and `allsets`.

The `no_archive` archive set is defined by default. Files selected to be in this archive set are never archived. Files in a temporary directory, such as `/sam1/tmp`, for example, might be included in the `no_archive` archive set.

The `allsets` archive set is used to define parameters that apply to all archive sets.

- Archive sets named for each StorageTek ASM file system are reserved for control structure information. StorageTek ASM file systems provide a default archive set for each file system. For each file system, both the metadata and data files are archived. The file system archive set encompasses the directory and link information and any files that are not included in another archive set. The default archive sets are given the names of their associated file systems and cannot be changed. For example, `samfs1` would be the archive set name for a file system configured and named as `samfs1`.
- Archive set names are limited to 29 characters. The characters are limited to the 26 uppercase and lowercase letters, the digits 0 through 9, and the underscore character (`_`).

## Archiving Operations

By default, the archiver makes one copy of each archive set, but you can request up to four archive copies for each archive set. An archive set and a copy number become a synonym for a collection of volumes. The archive copies provide duplication of files on separate volumes.

To ensure that files are complete before archiving, the archiver waits a specified period of time after the file is modified before archiving it. As mentioned previously, this period of time is called the *archive age*.

The data in a file must be modified before the file is considered to be a candidate for archiving or rearchiving. A file is not archived if it is only accessed. For example, issuing a `touch(1)` or an `mv(1)` command on a file does not cause it to be archived or rearchived. Issuing an `mv(1)` command alters the file name but not the file data, and this can have ramifications in a disaster recovery situation if you are restoring from `tar(1)` files. For more information on disaster recovery, see the *StorageTek ASM, StorageTek QFS, and StorageTek SM QFS Disaster Recovery Guide QFS, StorageTek ASM, and StorageTek ASM QFS Disaster Recovery Guide*.

Files are selected for archiving based on their archive age. The archive age can be defined for each archive copy.

Users can change the default time references on their files to values far in the past or future by using the `touch(1)` command. This can cause unexpected archiving results, however. To avoid such problems, the archiver adjusts the references so that they are always in the following range:

*creation\_time < time\_ref < time\_now*

The following sections describe the steps taken by the archiver from the initial file scan to the file copy process.

## Step 1: Identifying Files to Archive

There is a separate `sam-arfind` process for each mounted file system. The `sam-arfind` process monitors each file system to determine the files that need archiving. The file system notifies its `sam-arfind` process whenever a file is changed in a manner that would affect its archival state. Examples of such changes are file modification, rearchiving, unarchiving, and renaming. When notified, the `sam-arfind` process examines the file to determine the archive action required.

The `sam-arfind` process determines the archive set to which the file belongs by using the file properties descriptions. The characteristics used for determining a file's archive set include the following:

- The directory path portion of the file's name and, optionally, the complete file name using a regular expression
- The user name of the file's owner
- The group name of the file's owner
- A minimum file size
- A maximum file size

If the archive age of the file for one or more copies has been met or exceeded, `sam-arfind` adds the file to one or more archive requests for the archive set. The *archive request* is the collection of files that all belong to the same archive set. Separate archive requests are used for files being rearchived. This allows scheduling to be controlled independently for files not yet archived and for those being rearchived. The archive request is a file that resides in the following directory:

```
/var/opt/SUNWsamfs/archiver/file_sys/ArchReq
```

The files in this directory are binary files, and you can display them by using the `showqueue(1M)` command.

The archive request is sometimes referred to as an *ArchReq*.

If the archive age of the file for one or more copies has not been met, the directory in which the file resides and the time at which the archive age is reached is added to a scan list. Directories are scanned as the scan list times are reached. Files that have reached their archive age are added to archive requests.

If a file is offline, the `sam-arfind` process selects the volumes to be used as the source for the archive copy. If the file copy is being rearchived, the `sam-arfind` process selects the volume containing the archive copy that is being rearchived.



If a file is segmented, only those segments that have changed are selected for archival. The index of a segmented file contains no user data, so it is treated as a member of the file system archive set and is archived separately.

The archive priority is computed from file property characteristics and from file property multipliers associated with the archive set. Essentially, the computation is as follows:

$$\text{archive\_priority} = \text{the sum of } (\text{file\_property\_value} * \text{property\_multiplier})$$

Most *file\_property\_value* numbers are 1 or 0, as the property is `TRUE` or `FALSE`. For instance, the value of the property copy 1 is 1 if archive copy 1 is being made. The values of copy 2, copy 3, and copy 4 are, therefore, 0.

Others, such as archive age and file size, can have values other than 0 or 1.

The *property\_multiplier* values are determined from the `-priority` parameters for the archive set. Various aspects of a file, such as age or size, can be given values so that your site can alter the archive request's priority. For more information on the `-priority` parameter, see the `archiver.cmd(4)` man page.

The *archive\_priority* and the property multipliers are floating-point numbers. The default value for all property multipliers is 0.0. The archive request is set to the highest file priority in the archive request.

There are two methods by which files are marked for archiving: continuous archiving and scanning. With continuous archiving, the archiver works with the file system to determine which files need to be archived. With scanning, the archiver periodically peruses the file systems and selects files for archiving. The following sections describe these methods.

### **Continuous Archiving**

Continuous archiving is the default archiving method (`examine=noscan`). With continuous archiving, you can specify scheduling start conditions for an archive set by using the `-startage`, `-startcount`, and `-startsize` parameters. These conditions allow you to optimize archive timeliness versus archive work done.

- Example 1. If it takes an hour to create files that should be archived together, you can set the `-startage` parameter to 1 hour (`-startage 1h`) to ensure that all files are created before scheduling the Archive Request.
- Example 2. You can specify a `-startsize` of 150 gigabytes (`-startsize 150g`) to direct the archiver to wait until 150 gigabytes of data are ready to be archived.
- Example 3. If you know that 3000 files will be generated for archival, then specify `-startcount 3000` to ensure that the files get archived together.

When any of the scheduling start conditions are reached, the `sam-arfind` process sends each archive request to the archiver daemon, `sam-archiverd`, to be scheduled for file copying to archive media.

### **Scanned Archiving**

As an alternative to continuous archiving, you can specify `examine=scan` to direct `sam-arfind` to examine files for archival using scanning. Files needing archival are placed into archive requests. The `sam-arfind` process scans each file system periodically to determine which files need archiving. The first scan that `sam-arfind` performs is a directory scan. During this scan, `sam-arfind` descends recursively through the directory tree. Each file is examined, and the file status flag `archdone` is set if the file does not need archiving. During successive scans, the `.inodes` file is scanned. Only those inodes with the `archdone` flag not set are examined.

When the file system scanning has been completed, the `sam-arfind` process sends each archive request to the archiver daemon, `sam-archiverd`, to be scheduled for file copying to archive media. The `sam-arfind` process then sleeps for the duration specified by the `interval=time` directive. At the end of the interval, the `sam-arfind` process resumes scanning.

## **Step 2: Composing Archive Requests**

When archive requests are received by the `sam-archiverd` daemon, they are *composed*. This step describes the composition process.

All the files in an archive request might not be archived at one time. This can be caused by the capacity of the archive media or by the controls specified in the archiver command file. *Composing* is the process of selecting the files to be archived from the archive request at one time. When the archive copy operation has been completed for an archive request, the archive request is recomposed if files remain to be archived.

The `sam-archiverd` daemon orders the files in the archive requests according to certain default and site-specific criteria. The default operation is to archive all the files in an archive request to the same archive volumes in the order that they were found during the file system scan. The site-specific criteria allow you to control the order in which files are archived and how they can be distributed on volumes.

These criteria are called *archive set parameters*, and the order in which they are evaluated is as follows:

`-reserve`, `-join`, `-sort`, `-rsort` (performs a reverse sort), and `-drives`. For more information on these parameters, see the `archiver.cmd(4)` man page.

If the archive request belongs to an archive set that has `-reserve owner` specified, the `sam-archiverd` daemon orders the files in the archive request according to the file's directory path, user name, or group name. This action is controlled by the `-reserve` parameter for the archive set. The files belonging

to the first *owner* are selected for archiving. The remaining files are archived later.

If the archive request belongs to an archive set that has the `-join method` specified, the `sam-archiverd` daemon groups the files together according to the `-join method` specified. If a `-sort` or `-rsort method` is also specified, then the `sam-archiverd` daemon sorts the files within each group according to the `-sort` or `-rsort method`. The archive request is joined and sorted.

Each group of joined files is treated as if it were a single file for the remainder of the composing and scheduling processes.

If the archive request belongs to an archive set that has a `-sort` or `-rsort method` specified, the `sam-archiverd` daemon sorts the files according to the sort method specified on the `-sort` or `-rsort` parameter. Depending on the sort method, the `sam-archiverd` daemon tends to keep files together based on the sort method, age, size, or directory location. By default, the archive requests are not sorted, so the files are archived in the order in which they are encountered during the file system scan.

The `sam-archiverd` daemon determines whether the files are online or offline. If both online and offline files are in the archive request, the online files are selected for archiving first.

If the archive request was not required to be joined or sorted by a sort method, the offline files are ordered by the volume upon which the archive copies reside. This ensures that all files (within each archive set) on the same volume are staged at the same time in the order in which they were stored on the media. When more than one archive copy of an offline file is being made, the offline file is not released until all required copies are made. All the files to be staged from the same volume as the first file are selected for archiving.

Note that using the `-join`, `-sort`, or `-rsort` parameters can have a negative effect on performance when archiving offline files. This is due to the possibility that the order of the files to be archived does not match the order of the volumes needed for the offline files.

It is recommended that you use the `-join`, `-sort`, or `-rsort` parameters only for the first archive copy to be made. Other copies will most likely maintain the order of the first copy if enough archive media is available when the copies are started.

The archive requests are entered in the `sam-archiverd` daemon's scheduling queue.

### Step 3: Scheduling Archive Requests

The scheduler in the `sam-archiverd` daemon executes on demand when the following conditions exist:

- An archive request is entered in the scheduling queue.
- The archiving for an archive request has been completed.

- A change in media status is received from the catalog server.
- A message is received that changes the state of the archiver.

The archive requests in the scheduling queue are ordered by priority. Each time the scheduler executes, all archive requests are examined to determine if they can be assigned to a `sam-arcopy` process to have the files copied to archive media.

There must be drives available to use for making file copies. There must be volumes available that can be used by the archive set and have sufficient space to hold the files in the archive request.

### **Drives**

If the archive set has the `-drives` parameter specified, the `sam-archiverd` daemon divides the selected files in the archive request among multiple drives. If the number of drives available at this time is less than that specified by the `-drives` parameter, the smaller number is used.

If the total size of files in the archive request is less than the `-drivemin` value, only one drive is used. The `-drivemin` value is either the value specified by the `-drivemin` parameter or it is the `archmax` value.

The `archmax` value is specified by the `-archmax` parameter or the value defined for the media. For more information on the `-archmax` parameter and the `archmax=` directive, see the `archiver.cmd(4)` man page.

If the total size of files in the archive request is more than the `-drivemin` value, then the following value is computed:  $drive\_count = total\_size / drivemin$ . If  $drive\_count$  is less than the number of drives specified by the `-drives` parameter, then  $drive\_count$  becomes the number of drives to use.

Drives can take differing amounts of time to archive files. You can use the `-drivemax` parameter to obtain better drive utilization. The `-drivemax` parameter requires you to specify the maximum number of bytes to be written to a drive before rescheduling that drive for more data.

### **Volumes**

There must be a volume, or volumes, with enough space to hold at least some of the files in the archive request. The volume that has most recently been used for the archive set is used if there is enough space. Also, the volume must not be in use by the archiver.

If a volume usable for the archive set is presently busy, another is selected. This is true unless the `-fillvsns` parameter is specified. In this case, the archive request is not schedulable.

If an archive request is too big for one volume, the files that can fit on the volume are selected to be archived to the volume. If the archive request contains files that are too big to fit on one volume, and volume overflow for the

archive request is not selected, the files cannot be archived. An appropriate message for this condition is sent to the log.

You can specify volume overflow for the archive set (by using the `-ovflmin` parameter) or for the media (by using the `ovflmin=` directive). For more information on the `-ovflmin` parameter and the `ovflmin=` directive, see the `archiver.cmd(4)` man page. This specification, `ovflmin`, determines the minimum size for files to overflow media. An `ovflmin` specified for the archive set takes precedence over a media-defined `ovflmin`. If the size of the files is less than `ovflmin`, the files cannot be archived. An appropriate message for this condition is sent to the log.

If the size of the files is more than `ovflmin`, then additional volumes are assigned as required. The additional volumes are selected in order of decreasing size in order to minimize the number of volumes required for the file.

If no usable volumes can be found for the archive request, the archive request waits.

Certain properties, such as whether or not the file is online or offline, are used in conjunction with the archive priority (computed in Step 1) when determining the scheduling priority for a particular archive request. For more information on customizing the property multiplier, see the `-priority` parameters described on the `archiver.cmd(4)` man page.

For each archive request, the `sam-archiverd` daemon computes the scheduling priority by adding the archive priority to multipliers associated with various system resource properties. These properties are associated with the number of seconds that the archive request has been queued, whether or not the first volume to be used in the archiving process is loaded into a drive, and so on.

Using the adjusted priorities, the `sam-archiverd` daemon assigns each ready archive request to be copied.

## Step 4: Archiving the Files in an Archive Request

When an archive request is ready to be archived, the `sam-archiverd` daemon steps through each archive request to mark the archive file (tarball) boundaries so that each archive file's size is less than the `-archmax target_size` specification. If a single file is larger than `target_size`, it becomes the only file in an archive file.

For each archive request and each drive to be used, the `sam-archiverd` daemon assigns the archive request to a `sam-arcopy` process to copy the files to the archive media. If a single file is larger than `target_size`, it becomes the only file in an archive file. The archive information is entered into the inode.

If archive logging is enabled, an archive log entry is created.

If the file was staged, the disk space is released. This process continues until all files in the list have been archived.

A variety of errors and file status changes can prevent a file from being successfully copied. This can include read errors from the cache disk and write errors to the volumes. Status changes include modification since selection, file open for write, and file removed.

When the `sam-arcopy` process exits, the `sam-archiverd` daemon examines the archive request. If any files have not been archived, the archive request is recomposed.

## Sample Default Output

Figure 1 shows sample output is from running the `archiver(1M) -l` command.

**Figure 1. Output from the `archiver(1M) -l` Command**

```
# archiver

Archive media:
default:mo
media:mo archmax:5000000
media:lt archmax:50000000
Archive devices:
device:mo20 drives_available:1 archive_drives:1
device:lt30 drives_available:1 archive_drives:1
Archive file selections:
Filesystem samfs1:
samfs1 Metadata
    copy:1 arch_age:240
big path:.. minsize:512000
    copy:1 arch_age:240
all path:
    copy:1 arch_age:30
Archive sets:
all
    copy:1 media:mo
big
    copy:1 media:lt
samfs1
    copy:1 media:mo
```

## Archiver Daemons

The `sam-archiverd` daemon schedules the archiving activity. The `sam-arfind` process assigns files to be archived to archive sets. The `sam-arcopy` process copies the files to be archived to the selected volumes.

The `sam-archiverd` daemon is started by `sam-fsd` when StorageTek ASM activity begins. The `sam-archiver` daemon executes the `archiver(1M)`

command to read the `archiver.cmd` file and builds the tables necessary to control archiving. It starts a `sam-arfind` process for each mounted file system; likewise, if a file system is unmounted, the associated `sam-arfind` process is stopped. The `sam-archiverd` process then monitors `sam-arfind` and processes signals from an operator or other processes.

## Archive Log Files and Event Logging

The `sam-arfind` and `sam-arcopy` processes produce a log file that contains information about each archived or automatically unarchived file. The log file is a continuous record of archival action. You can use the log file to locate earlier copies of files for traditional backup purposes.

This file is not produced by default. You can use the `logfile=` directive in the `archiver.cmd` file to specify that a log file be created and to specify the name of the log file. You determine the name of this file. For more information on the log file, see the “[The archiver.cmd Directives](#)” on page 62 in this chapter and see the `archiver.cmd(4)` man page.

The archiver logs warnings and informational messages in the log file using the `syslog` facility and `archiver.sh`.

[Figure 2](#) shows sample lines from an archiver log file with definitions for each field.

**Figure 2. Archiver Log File Lines**

```
A 2001/03/23 18:42:06 mo 0004A arset0.1 9a089.1329 samfs1
118.51162514 t0/fdn f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.1 samfs1 189.53
1515016 t0/fae f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.b92 samfs1 125.53
867101 t0/fai f 0 56
A 2001/03/23 19:13:09 lt SLOT22 arset0.2 798.1 samfs1 71531.14
1841087 t0/fhh f 0 51
A 2001/03/23 19:13:10 lt SLOT22 arset0.2 798.e0e samfs1 71532.12
543390 t0/fhg f 0 51
A 2003/10/23 13:30:24 dk DISK01/d8/d16/f216 arset4.1 810d8.1 qfs2
119571.301 1136048 t1/fileem f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.8ad
qfs2 119573.295 1849474 t1/fileud f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.16cb
qfs2 119576.301 644930 t1/fileen f 0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.1bb8
qfs2 119577.301 1322899 t1/fileeo f 0 0
```

Reading left to right, the fields in the previous listing have the content shown in [Table 31](#).

**Table 31. Archiver Log File Fields**

Field	Content
1	Archive activity, as follows: <ul style="list-style-type: none"> <li>• A for archived.</li> <li>• R for rearchived.</li> <li>• U for unarchived.</li> </ul>
2	Date of archive action in <i>yyyy/mm/dd</i> format.
3	Time of archive activity in <i>hh:mm:ss</i> format.
4	Archive media type. For information on media types, see the <i>mcf(4)</i> man page.
5	VSN. For removable media cartridges, this is the volume serial name. For disk archives, this is the disk volume name and archive <i>tar(1)</i> file path.
6	Archive set and copy number.
7	Physical position of start of archive file on media ( <i>tar(1)</i> file) and file offset within the archive file in hexadecimal.
8	File system name.
9	Inode number and generation number. The generation number is an additional number used in addition to the inode number for uniqueness since inode numbers get re-used.
10	Length of file if file is written on only 1 volume. Length of section if file is written on multiple volumes.
11	Path and name of file relative to the file system's mount point.
12	Type of file, as follows: <ul style="list-style-type: none"> <li>• d for directory.</li> <li>• f for regular file.</li> <li>• l for symbolic link.</li> <li>• R for removable media file.</li> <li>• I for segment index.</li> <li>• S for data segment.</li> </ul>
13	Section of an overflowed file or segment. If the file is an overflowed file, the value is nonzero. The value is zero for all other file types.
14	Equipment Ordinal of the drive upon which the file was archived.



## ■ The archiver.cmd File

The `archiver.cmd` file controls the archiver's behavior. By default, the archiver runs whenever `sam-fsd` is started and a StorageTek ASM file system is mounted. The default actions that the archiver takes are as follows:

- Archive all files to all available volumes.
- The archive age for all files is four minutes.
- The archive interval is 10 minutes.

It is likely that you will customize the actions of the archiver to meet the archiving requirements of your site. These actions are controlled by directives located in the archiver command file (`archiver.cmd`).

### To Create or Modify an archiver.cmd File and Propagate Your Changes

1. Decide whether you want to edit the `archiver.cmd` file or if you want to edit a temporary `archiver.cmd` file. (Optional)

Perform this step if you have an `/etc/opt/SUNWsamfs/archiver.cmd` file and your system is already archiving files. Consider copying your `archiver.cmd` file to temporary location where you can edit and test it before putting it into production.

2. Use `vi(1)` or another editor to edit your `archiver.cmd` file or the temporary file.

Add the directives you need in order to control archiving at your site. For information on the directives you can include in this file, see [“The archiver.cmd Directives” on page 62](#) and [“Disk Archiving” on page 99](#).

3. Save and close the `archiver.cmd` file or the temporary file.
4. Use the `archiver(1M) -lv` command to verify the correctness of the file.

Whenever you make changes to the `archiver.cmd` file, you should check for syntax errors using the `archiver(1M)` command. Specifying the `archiver(1M)` command as follows evaluates an `archiver.cmd` file against the current StorageTek ASM system:

```
# archiver -lv
```

The preceding command lists all options and writes a listing of the `archiver.cmd` file, volumes, file system content, and errors to the standard output file (`stdout`). Errors prevent the archiver from running.

By default, the `archiver(1M)` command evaluates file `/etc/opt/SUNWsamfs/archiver.cmd` for errors. If you are working

with a temporary `archiver.cmd` file prior to putting it into production, you can use the `-c` option on the `archiver(1M)` command and supply this temporary file's name.

5. Repeat Step 2, Step 3, and Step 4 until your file is free from errors.

You must correct all errors before you move onto the next step. The archiver does not archive any files if it finds errors in the `archiver.cmd` file.

6. Move the temporary file to `/etc/opt/SUNWsamfs/archiver.cmd`. (Optional)

Perform this step only if you are working with a temporary file.

7. Save and close the `archiver.cmd` file.
8. Use the `samd(1M) config` command to propagate the file changes and restart the system.

```
# samd config
```

## The `archiver.cmd` File

The `archiver.cmd` file consists of the following types of directives:

- General directives
- Archive set assignment directives
- Archive set directives
- VSN pool directives
- VSN association directives

The directives consist of lines of text read from the `archiver.cmd` file. Each directive line contains one or more fields separated by spaces or tabs. Any text that appears after the pound sign character (`#`) is treated as a comment and is not examined. Lines can be continued onto the next line by ending the line with a backslash (`\`).

Certain directives in the `archiver.cmd` file require you to specify a unit of time or a unit in bytes. To specify these units, use one of the following letters in [Table 32](#) as a suffix to the number that signifies the unit.

**Table 32. The `archiver.cmd` File Directive Units**

Unit Suffixes	Significance
<b>Time Suffixes:</b>	
s	Seconds.
m	Minutes. 60 seconds.
h	Hours. 3,600 seconds.
d	Days. 86,400 seconds.
w	Weeks. 604,800 seconds.
y	Years. 31,536,000 seconds.
<b>Size Suffixes:</b>	
b	Bytes.
k	Kilobytes. $2^{10}$ , or 1,024, bytes.
M	Megabytes. $2^{20}$ , or 1,048,576, bytes.
G	Gigabytes. $2^{30}$ , or 1,073,741,824, bytes.
T	Terabytes. $2^{40}$ , or 1,099,511,627,776, bytes.
P	Petabytes. $2^{50}$ , or 1,125,899,906,842,624 bytes.
E	Exabytes. $2^{60}$ , or 1,152,921,504,606,846,976 bytes.

## An `archiver.cmd` File Example

[Figure 3](#) shows a sample `archiver.cmd` file. The comments at the right indicate the various types of directives.

**Figure 3. archiver.cmd File Example**

```

interval = 30m                # General directives
logfile = /var/opt/SUNWsamfs/archiver/archiver.log

fs = samfs1                    # Archive Set Assignments
no_archive tmp
work work
    1 1h
    2 3h
images images -minsize 100m
    1 1d
    2 1w
samfs1_all .
    1 1h
    2 1h

fs = samfs2                    # Archive Set Assignments
no_archive tmp
system . -group sysadmin
    1 30m
    2 1h
samfs2_all .
    1 10m
    2 2h

params                          # Archive Set Directives
allsets -drives 2
images.1 -join path -sort size
endparams

vsns                             # VSN Associations
samfs1.1 mo optic-2A
samfs1.2 lt TAPE01
work.1 mo optic-[3-9][A-Z]
work.2 lt .*
images.1 lt TAPE2[0-9]
images.2 lt TAPE3[0-9]
samfs1_all.1 mo.*
samfs1_all.2 lt.*
samfs2.1 mo optic-2A
samfs2.2 lt TAPE01
system.1 mo optic08a optic08b
system.2 lt ^TAPE4[0-1]
samfs2_all.1 mo.*
samfs2_all.2 lt.*
endvsns

```

## ■ The archiver.cmd Directives

The following sections explain the archiver.cmd directives. They are as follows:

- [“Global Archiving Directives” on page 63](#)
- [“File System Directives” on page 71](#)
- [“Archive Set Assignment Directive” on page 72](#)
- [“Archive Copy Directives” on page 79](#)
- [“Archive Set Copy Parameters” on page 82](#)
- [“VSN Association Directives” on page 96](#)
- [“VSN Pools Directives” on page 98](#)

## Global Archiving Directives

Global directives control the overall archiver operation. These global directives in an `archiver.cmd` file can be identified either by the equal sign (=) in the second field or by the absence of additional fields. These directives allow you to optimize archiver operations for your site's configuration.

Global directives must be specified prior to any `fs=` directives in your `archiver.cmd` file. The `fs=` directives are those that pertain to specific file systems. The archiver issues a message if it detects a global directive after an `fs=` directive.

### The `archivemeta` Directive: Controlling Whether Metadata is Archived

The `archivemeta` directive controls whether or not file system metadata is archived. If files are often moved around and there are typically many changes to the directory structures in your file system, you would want to archive your metadata. If, however, the directory structures are very stable, you can disable metadata archiving and reduce the actions performed by your removable media drives as cartridges are loaded and unloaded to archive metadata. By default, metadata is archived.

This directive has the following format:

```
archivemeta = state
```

For *state*, specify either `on` or `off`. The default is `on`.

Metadata archiving differs depending on whether you are using a Version 1 or a Version 2 superblock, as follows:

- For Version 1 file systems, the archiver archives directories, removable media files, segment index inodes, and symbolic links as metadata.
- For Version 2 file systems, removable media files and symbolic links are stored in inodes rather than in data blocks. They are not archived. Only

directories and segment index inodes are archived as metadata. Symbolic links are archived as data.

## The `archmax` Directive: Controlling the Size of Archive Files

The `archmax` directive specifies the maximum size of an archive file. User files are combined to form the archive file. No more user files are added to the archive file after the `target_size` is met. Large user files are written in a single archive file.

To change the defaults, use the following directive:

```
archmax=media target_size
```

**Table 33. Arguments for the `archmax` Directive**

Argument	Meaning
<code>media</code>	The media type. For the list of valid media types, see the <code>mcf(4)</code> man page.
<code>target_size</code>	Specifies the maximum size of the archive file. The maximum size of an archive file is media-dependent. By default, archive files written to optical disks are no larger than 5 megabytes. The default maximum archive file size for tapes is 512 megabytes.

There are advantages and disadvantages to setting large or small sizes for archive files. For example, if you are archiving to tape and `archmax` is set to a large size, the tape drive stops and starts less often. However, when writing large archive files, there is the possibility that when an end-of-tape is reached prematurely, a large amount of tape can be wasted. As a rule, `archmax` should not be set to more than 5 percent of the media capacity. For example, you can use the following `archmax` directive for a 20 gigabyte tape:

```
archmax=sg 1G
```

The `archmax` directive can also be set for an individual archive set.

## The `bufsize` Directive: Setting the Archiver Buffer Size

By default, a file being archived is copied to archive media using a memory buffer. You can use the `bufsize` directive to specify a nondefault buffer size and, optionally, to lock the buffer. These actions can improve performance, and you can experiment with different `buffer_size` values.

This directive has the following format:

```
bufsize=media buffer_size [ lock ]
```

**Table 34. Arguments for the `bufsize` Directive**

Argument	Meaning
<i>media</i>	The media type. For the list of valid media types, see the <code>mcf(4)</code> man page.
<i>buffer_size</i>	Specify a number from 2 through 32. The default is 4. This value is multiplied by the <code>dev_blksize</code> value for the media type, and the resulting buffer size is used. The <code>dev_blksize</code> can be specified in the <code>defaults.conf</code> file. For more information on this file, see the <code>defaults.conf(4)</code> man page.
lock	<p>The lock argument indicates whether or not the archiver should use locked buffers when making archive copies. If lock is specified, the archiver sets file locks on the archive buffer in memory for the duration of the <code>sam-arcopy(1M)</code> operation. This avoids the overhead of locking and unlocking the buffer for each I/O request and can result in a reduction in system CPU time.</p> <p>The lock argument should be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition.</p> <p>The lock argument is effective only if direct I/O is enabled for the file being archived. By default, lock is not specified and the file system sets the locks on all direct I/O buffers, including those for archiving. For more information on enabling direct I/O, see the <code>setfa(1)</code> man page, the <code>sam_setfa(3)</code> library routine man page, or the <code>-O forcedirectio</code> option on the <code>mount_samfs(1M)</code> man page.</p>

For example, this directive can be specified in the `archiver.cmd` file in a line like the following:

```
bufsize=od 7 lock
```

You can specify a buffer size and a lock on an archive set basis by using the `-bufsize` and `-lock` archive set copy parameters. For more information, see [“Archive Set Copy Parameters” on page 82](#).

## The `drives` Directive: Controlling the Number of Drives Used for Archiving

By default, the archiver uses all of the drives in an automated library for archiving. To limit the number of drives in an automated library used by the archiver, use the `drives` directive.

This directive has the following format:

```
drives=auto_lib count
```

**Table 35. Arguments for the `drives` Directive**

Argument	Meaning
<code>auto_lib</code>	The Family Set name of the automated library as defined in the mcf file.
<code>count</code>	The number of drives to be used for archiving activities.

Also see the `-drivemax`, `-drivemin`, and `-drives` archive set copy parameters described in [“Specifying the Number of Drives for an Archive Request: `-drivemax`, `-drivemin`, and `-drives`”](#) on page 83.

## The `examine` Directive: Controlling Archive Scans

New files and files that have changed are candidates for archiving. The archiver finds such files by implementing one of the following methods:

- Continuous archiving. When continuous archiving is implemented, the archiver works with the file system to detect file changes immediately after they occur.
- Scan-based archiving. With scan-based archiving, the archiver scans the file system periodically looking for files that need to be archived.

The `examine` directive controls whether the archiver performs continuous or scan-based archiving, as follows:

```
examine=method
```



Specify one of the keywords shown in [Table 36](#) for *method*.

**Table 36. Values for the `examine` Directive's *method* argument**

<i>method</i> Value	Meaning
noscan	Specifies continuous archiving. After the initial scan, directories are scanned only when the content changes and archiving is required. Directory and inode information is not scanned. This archiving method provides better performance than scan-based archiving, particularly for file systems with more than 1,000,000 files. Default.
scan	Specifies scan-based archiving. The initial file system scan is a directory scan. Subsequent scans are inode scans.
scandirs	Specifies scan-based archiving on directories only. When specified, if the archiver finds a directory with the <code>no_archive</code> attribute set, that directory is not scanned. Files that do not change can be placed in such a directory, and this can dramatically reduce the amount of time spent on archiving scans.
scaninodes	Specifies scan-based archiving on inodes only.

## The `interval` Directive: Specifying an Archive Interval

The archiver executes periodically to examine the status of all mounted StorageTek ASM file systems. The timing is controlled by the archive interval. The *archive interval* is the time between scan operations on each file system. To change the time, use the `interval` directive.

**Note:** The `interval` directive is effective only if the `examine=scan` directive is also specified in the `archiver.cmd` file.

This directive has the following format:

```
interval=time
```

For *time*, specify the time, in seconds, between scan operations on a file system. By default, *time* is interpreted in seconds. By default, `interval=600`, which is 10 minutes. You can specify a unit of time, such as minutes, hours, and so on. For information on specifying a unit of time, see [Table 32](#).

If the archiver receives the `samu(1M)` utility's `:arrun` command, it begins scanning all file systems immediately. If the `examine=scan` directive is also specified in the `archiver.cmd` file, a scan is performed after `:arrun` or `:arscan` is issued.

If the `hwm_archive` mount option is set for the file system, the archive interval can be shortened automatically. This mount option specifies that the archiver commence its scan when the file system is filling up and the high water mark is crossed. The `high=percent` mount option sets the high water mark for the file system.

For more information on specifying the archive interval, see the `archiver.cmd(4)` man page. For more information on setting mount options, see the `mount_samfs(1M)` man page.

## The `logfile` Directive: Specifying An Archiver Log File

The archiver can produce a log file that contains information about each file that is archived, rearchived, or automatically unarchived. The log file is a continuous record of archival action. To specify a log file, use the `logfile` directive. This directive has the following format:

```
logfile=pathname
```

For *pathname*, specify the absolute path and name of the log file. By default, this file is not produced.

**Example.** Assume that you want to back up the archiver log file every day by copying the previous day's log file to an alternate location. This can be accomplished if you make sure that the copy is performed when the archiver log file is closed. In other words, you must not perform the copy operation while the archiver log file is open for a write operation.

## To Back Up an Archiver Log File

The steps you need to take are as follows:

1. Use the `mv(1)` command to move the archiver log file within UFS.  
This gives any `sam-arfind(1M)` or `sam-arcopy(1M)` operations time to finish writing to the archiver log file.
2. Use the `mv(1)` command to move the previous day's archiver log file to the StorageTek ASM file system.

The `logfile` directive can also be set for an individual file system.

## The `notify` Directive: Renaming the Event Notification Script

The `notify` directive sets the name of the archiver's event notification script file to *filename*. This directive has the following format:

```
notify=filename
```

For *filename*, specify the name of the file containing the archiver event notification script or the full path to this file.

The default file name is as follows:

```
/etc/opt/SUNWsamfs/scripts/archiver.sh
```

The archiver executes this script to process various events in a site-specific manner. The script is called with a keyword for the first argument. The keywords are as follows: *emerg*, *alert*, *crit*, *err*, *warning*, *notice*, *info*, and *debug*.

Additional arguments are described in the default script. For more information, see the *archiver.sh(1M)* man page.

## The `ovflmin` Directive: Controlling Volume Overflow

*Volume overflow* is the process of allowing archived files to span multiple volumes. Volume overflow is enabled when you use the `ovflmin` directive in the `archiver.cmd` file. When a file size exceeds the `ovflmin` directive's *minimum\_file\_size* argument, the archiver writes another portion of this file to another available volume of the same type, if necessary. The portion of the file written to each volume is called a *section*.

**Note:** Before using volume overflow, make sure that you understand the concept. Use volume overflow with caution only after thoroughly assessing the effect on your site. Disaster recovery and recycling are much more difficult with files that span volumes.

The archiver controls volume overflow through the `ovflmin` directive. The `ovflmin` directive specifies the minimum size file that is allowed to overflow a volume. By default, volume overflow is disabled.

This directive has the following format:

```
ovflmin = media minimum_file_size
```

**Table 37. Arguments for the `ovflmin` Directive**

Argument	Meaning
<i>media</i>	The media type. For a list of valid media types, see the <i>mcf(4)</i> man page.
<i>minimum_file_size</i>	Specify the minimum size of the file to overflow

**Example 1.** Assume that many files exist with a length that is a significant fraction (say 25 percent) of an `MO` media cartridge. These files partially fill the volumes and leave unused space on each volume. To get better packing of

the volumes, set `ovflmin` for `mo` media to a size slightly smaller than the size of the smallest file. The following directive sets it to 150 megabytes:

```
ovflmin=mo 150m
```

Note that enabling volume overflow in this example also causes two volumes to be loaded for archiving and staging the file.

The `ovflmin` directive can also be set for an individual archive set.

**Example 2.** The `sls(1)` command lists the archive copy showing each section of the file on each VSN. [Figure 4](#) shows the archiver log file and the `sls -D` command output for a large file named `file50` that spans multiple volumes.

**Figure 4. Archiver Log File Example**

```
A 97/01/13 16:03:29 lt DLT000 big.1 7eed4.1 samfs1 13.7
477609472 00 big/file50 0 0

A 97/01/13 16:03:29 lt DLT001 big.1 7fb80.0 samfs1 13.7
516407296 01 big/file50 0 1

A 97/01/13 16:03:29 lt DLT005 big.1 7eb05.0 samfs1 13.7
505983404 02 big/file50 0 2
```

[Figure 4](#) shows that `file50` spans three volumes with VSNs of `DLT000`, `DLT001`, and `DLT005`. The position on the volume and the size of each section is indicated in the seventh and tenth fields respectively, and matches the `sls -D` output also shown. For a complete description of the archiver log entry, see the `archiver(1M)` man page.

[Figure 5](#) shows the `sls -D` command and output.

**Figure 5. `sls(1M) -D` Command and Output**

```
# sls -D file50
file50:
 mode: -rw-rw---- links: 1 owner: gmm group: sam
 length: 1500000172 admin id: 7 inode: 1407.5
 offline; archdone; stage -n
 copy1: ---- Jan 13 15:55 lt
 section 0: 477609472 7eed4.1 DLT000
 section 1: 516407296 7fb80.0 DLT001
 section 2: 505983404 7eb05.0 DLT005
 access: Jan 13 17:08 modification: Jan 10 18:03
 changed: Jan 10 18:12 attributes: Jan 13 16:34
 creation: Jan 10 18:03 residence: Jan 13 17:08
```

Volume overflow files do not generate checksums. For more information on using checksums, see the `ssum(1)` man page.

**Note:** When you use the volume overflow feature, be aware that it is difficult to retrieve volume overflow data in the event of a disaster. For information on how to retrieve such a file, see the examples in the *StorageTek ASM, StorageTek QFS, and StorageTek SM QFS Disaster Recovery Guide QFS, StorageTek ASM, and StorageTek ASM QFS Disaster Recovery Guide*. For more information, see the `request(1)` man page.

## The `wait` Directive: Delaying Archiver Startup

The `wait` directive causes the archiver to wait for a start signal from `samu(1M)` or ASM QFS Manager. When this signal is received, typical archiver operations are begun. By default, the archiver begins archiving when started by `sam-fsd(1M)`. To delay archiving, use the `wait` directive. This directive has the following format:

```
wait
```

The `wait` directive can also be set for an individual file system.

## File System Directives

You can use the `fs=` directive to include directives specific to a particular file system in the `archiver.cmd` file after the general directives. After an `fs=` directive is encountered, the archiver assumes that all subsequent directives specify actions to be taken only for individual file systems.

### The `fs` Directive: Specifying the File System

By default, archiving controls apply to all file systems. However, you can confine some controls to an individual file system. To specify an individual file system, use the `fs` directive. This directive has the following format:

```
fs=fsname
```

For *fsname*, specify the file system name as defined in the `mcf` file.

The general directives and archive set association directives that occur after these directives apply only to the specified file system until another `fs=` directive is encountered. For instance, you can use this directive to specify a different log file for each file system.

### Other File System Directives

Several directives can be specified both as global directives for all file systems and as directives that are specific to one file system. Their effects are the same regardless of where they are specified. These directives are as follows:

- The `interval` directive. For more information on this directive, see [“The interval Directive: Specifying an Archive Interval” on page 67](#).
- The `logfile` directive. For more information on this directive, see [“The logfile Directive: Specifying An Archiver Log File” on page 68](#).
- The `wait` directive. For more information on this directive, see [“The wait Directive: Delaying Archiver Startup” on page 71](#).

## Archive Set Assignment Directive

By default, files are archived as part of the archive set named for the file system. In ASM QFS Manager, an *archive policy* defines archive sets. However, you can specify archive sets to include files that share similar characteristics. If a file does not match one of the specified archive sets, it is archived as part of the default archive set named for the file system.

### Assigning Archive Sets

The archive set membership directives assign files with similar characteristics to archive sets. The syntax of these directives is patterned after the `find(1)` command. Each archive set assignment directive has the following format:

```
archive_set_name path [search_criteria1 search_criteria2 ... ] [file_attributes]
```

**Table 38. Arguments for the Archive Set Assignment Directive**

Argument	Meaning
<i>archive_set_name</i>	A site-defined name for the archive set. Must be the first field in the archive set assignment directive. An archive set name is usually indicative of the characteristics of the files belonging to the archive set. Archive set names are restricted to the letters in the alphabet, numbers, and the underscore character ( <code>_</code> ). No other special characters or spaces are allowed. The first character in the archive set name must be a letter.  To prevent archiving for various files, specify <code>no_archive</code> as the <i>archive_set_name</i> .
<i>path</i>	A path relative to the mount point of the file system. This allows an archive set membership directive to apply to multiple StorageTek ASM file systems. If the path is to include all of the files in a file system, use a period ( <code>.</code> ) for the path field. A leading slash ( <code>/</code> ) is not allowed in the path. Files in the directory specified by <i>path</i> , and its subdirectories, are considered for inclusion in this archive set.
<i>search_criteria1</i> <i>search_criteria2</i>	Zero, one, or more <i>search_criteria</i> arguments can be specified. Search criteria can be specified to restrict the archive set according to file size, file ownership, and other factors. For information on possible <i>search_criteria</i> arguments, see the following sections.
<i>file_attributes</i>	Zero, one, or more <i>file_attributes</i> can be specified. These file attributes are set for files as the <code>sam-arcfind</code> process scans a file system during archiving.

**Example 1.** [Figure 6](#) shows typical archive set membership directives.

**Figure 6. Archive Set Membership Directives**

```

hmk_files    net/home/hmk    -user hmk
datafiles    xray_group/data -size 1M
system      .

```

**Example 2.** You can suppress the archiver by including files in an archive set named `no_archive`. [Figure 7](#) shows lines that prevent archiving of files in a

`tmp` directory, at any level, and regardless of the directory in which the `tmp` directory resides within the file system.

**Figure 7. Archiving Directives that Prevent Archiving**

```
fs = samfs1
no_archive tmp
no_archive . -name */tmp/
```

The following sections describe the *search\_criteria* that you can specify.

### File Size *search\_criteria*: `-access`

You can use the `-access age` characteristic to specify that the age of a file be used to determine archive set membership. When you use this *search\_criteria*, files with access times older than *age* are rearchived to different media. For *age*, specify an integer number followed by one of the suffixes shown in [Table 39](#).

**Table 39. `-access age` Suffixes**

Letter	Meaning
s	Seconds
m	Minutes
h	Hours
d	Days
w	Weeks
y	Years

For example, you can use this directive to specify that files that have not been accessed in a long time be rearchived to cheaper media.

### File Size *search\_criteria*: `-minsize` and `-maxsize`

The size of a file can be used to determine archive set membership using the `-minsize size` and `-maxsize size` characteristics. For *size*, specify an integer followed by one of the letters shown in [Table 40](#).

**Table 40. `-minsize` and `-maxsize size` Suffixes**

Letter	Meaning
b	Bytes
k	Kilobytes
M	Megabytes
G	Gigabytes



**Table 40.** `-minsize` and `-maxsize` *size* Suffixes

Letter	Meaning
T	Terabytes
P	Petabytes
E	Exabytes

**Example.** The lines in this example specify that all files of at least 500 kilobytes, but less than 100 megabytes, belong to the archive set `big_files`. Files bigger than 100 megabytes belong to the archive set `huge_files`. [Figure 8](#) shows the lines.

**Figure 8.** Using the `-minsize` and `-maxsize` Directive Examples

```
big_files . -minsize 500k -maxsize 100M
huge_files . -minsize 100M
```

### Owner and Group *search\_criteria*: `-user` and `-group`

The ownership and group affiliation can be used to determine archive set membership using the `-user name` and `-group name` characteristics.

[Figure 9](#) shows examples of these directives.

**Figure 9.** Using the `-user` and `-group` Directive Examples

```
adm_set . -user sysadmin
mktng_set . -group marketing
```

All files belonging to user `sysadmin` belong to archive set `adm_set`, and all files with the group name of `marketing` are in the archive set `mktng_set`.

### File Name *search\_criteria* Using Pattern Matching: `-name` *regex*

The names of files that are to be included in an archive set can be specified by using regular expressions. The `-name regex` specification as a *search\_criteria* specifies that any complete path matching the regular expression *regex* is a member of the archive set.

The *regex* argument follows the conventions as outlined in the `regexp(5)` man page. Note that regular expressions do not follow the same conventions as UNIX wildcards.

Internally, all files beneath the selected directory are listed (with their specified paths relative to the mount point of the file system) and passed along for pattern matching. This allows you to create patterns in the `-name regex` field to match both file names and path names.

**Examples**

1. The following directive restricts files in the archive set `images` to those files ending with `.gif`:

```
images . -name \.gif$
```

2. The following directive selects files that start with the characters `GEO`:

```
satellite . -name /GEO
```

3. You can use regular expressions with the `no_archive` archive set. The following specification prevents any file ending with `.o` from being archived:

```
no_archive . -name \.o$
```

4. Assume that your `archiver.cmd` file contains the lines shown in [Figure 10](#).

**Figure 10. Regular Expression Example**

```
# File selections.
fs = samfs1
    1 1s
    2 1s
no_archive share/marketing -name fred\.
```

With this `archiver.cmd` file, the archiver does not archive `fred.*` in the user directories or subdirectories. Archiving occurs for files as follows:

- [Figure 11](#) shows the files not archived if you specify the directives shown in [Figure 10](#).

**Figure 11. Files not Archived (Assuming Directives Shown in [Figure 10](#))**

```
/saml/share/marketing/fred.anything
/saml/share/marketing/first_user/fred.anything
/saml/share/marketing/first_user/first_user_sub/fred.anything
```

- [Figure 12](#) shows the files that are archived if you specify the directives shown in [Figure 10](#).

**Figure 12. Files Archived (Assuming Directives Shown in [Figure 10](#))**

```
/saml/fred.anything
/saml/share/fred.anything
/saml/testdir/fred.anything
/saml/testdir/share/fred.anything
/saml/testdir/share/marketing/fred.anything
/saml/testdir/share/marketing/second_user/fred.anything
```

5. Assume that your `archiver.cmd` file contains the lines shown in [Figure 13](#).

**Figure 13. Example archiver.cmd File**

```
# File selections.
fs = samfs1
    1 ls
    2 ls
no_archive share/marketing -name ^share/marketing/[^/]*fred\.
```

The `archiver.cmd` file in [Figure 13](#) does not archive `fred.*` in the user home directories. This archives `fred.*` in the user subdirectories and in the directory `share/marketing`. In this case, the user home directories happen to be `first_user`. This example takes anything as a user's home directory from `share/marketing/` until the next slash character (`/`). Archiving occurs for files as follows:

- The following files are not archived:

```
/saml/share/marketing/first_user/fred.anything
```

- [Figure 14](#) shows the files that are archived if you specify the directives shown in [Figure 13](#).

**Figure 14. Files Archived (Assuming Directives Shown in [Figure 13](#))**

```
/saml/share/fred.anything
/saml/share/marketing/fred.anything
/saml/share/marketing/first_user/first_user_sub/fred.anything
/saml/fred.anything
/saml/testdir/fred.anything
/saml/testdir/share/fred.anything
/saml/testdir/share/marketing/fred.anything
/saml/testdir/share/marketing/second_user/fred.anything
/saml/testdir/share/marketing/second_user/sec_user_sub/fred.any
```

## Release and Stage *file\_attributes*: **-release** and **-stage**

You can set the release and stage attributes associated with files within an archive set by using the `-release` and `-stage` options, respectively. Both of these settings override stage or release attributes that a user might have set previously.

The `-release` option has the following format:

```
-release attributes
```

The *attributes* for the `-release` directive follow the same conventions as the `release(1)` command and are as shown in [Table 41](#).

**Table 41. The `-release` Option**

<i>attributes</i>	Meaning
a	Release the file following the completion of the first archive copy.
d	Reset to default.
n	Never release the file.
p	Partially release the file's disk space.

The `-stage` option has the following format:

```
-stage attributes
```

The *attributes* for the `-stage` directive follow the same conventions as the `stage(1)` command and are as shown in [Table 42](#).

**Table 42. The `-stage` Directive's *attributes***

<i>attributes</i>	Meaning
a	Associative stage the files in this archive set.
d	Reset to default.
n	Never stage the files in this archive set.

The following example shows how you can use file name specifications and file attributes in order to partially release Macintosh resource directories:

```
MACS . -name .*/\.\rscs/ -release p
```

## Archive Set Membership Conflicts

Sometimes the choice of path and other file characteristics for inclusion of a file in an archive set results in ambiguous archive set membership. These situations are resolved in the following manner:

1. The membership definition occurring first in the archive set is chosen.
2. Membership definitions local to a file system are chosen before any globally defined definitions.
3. A membership definition that exactly duplicates a previous definition is noted as an error.

As a consequence of these rules, more restrictive membership definitions should be placed earlier in the directive file.

When controlling archiving for a specific file system (using the `fs=fsname` directive), the archiver evaluates the file system specific directives before evaluating the global directives. Thus, files can be assigned to a local archive set (including the `no_archive` archive set) instead of being assigned to a global archive. This has implications when setting global archive set assignments such as `no_archive`.

Figure 15 shows an `archiver.cmd` file.

**Figure 15. An `archiver.cmd` File With Possible Membership Conflicts**

```
no_archive . -name *.*\.*$
fs = samfs1
  allfiles .
fs = samfs2
  allfiles .
```

In reading Figure 15, it appears that the administrator did not intend to archive any of the `.o` files across both file systems. However, because the local archive set assignment `allfiles` is evaluated before the global archive set assignment `no_archive`, the `.o` files in the `samfs1` and `samfs2` file systems are archived.

Figure 16 shows the directives to use to ensure that no `.o` files are archived in both file systems.

**Figure 16. Corrected `archiver.cmd` File**

```
fs = samfs1
  no_archive . -name *.*\.*$
  allfiles .
fs = samfs2
  no_archive . -name *.*\.*$
  allfiles .
```

## Archive Copy Directives

If you do not specify archive copies, the archiver writes a single archive copy for files in the archive set. By default, this copy is made when the archive age of the file is four minutes. If you require more than one archive copy, all copies, including the first, must be specified using archive copy directives.

The archive copy directives begin with a *copy\_number* that is an integer digit. This digit (1, 2, 3, or 4) is the copy number. The digit is followed by one or more arguments that specify archive characteristics for that copy.

The archive copy directives must appear immediately after the archive set assignment directive to which they pertain. Each archive copy directive has the following format:

```
copy_number [ -release | -norelease ] [archive_age] [unarchive_age]
```

The following sections describe the archive copy directive arguments.

### Releasing Disk Space After Archiving: `-release`

You can specify that the disk space for files be automatically released after an archive copy is made by using the `-release` directive after the copy number. This option has the following format:

```
-release
```

In [Figure 17](#), files with the group `images` are archived when their archive age reaches 10 minutes. After archive copy 1 is made, the disk cache space is released.

**Figure 17. An `archiver.cmd` File Using the `-release` Directive**

```
ex_set . -group images
      1 -release 10m
```

### Delaying Disk Space Release: `-norelease`

You might not want to release disk space until multiple archive copies are completed. The `-norelease` option prevents the automatic release of disk cache until all copies marked with `-norelease` are made. This option has the following format:

```
-norelease
```

[Figure 18](#) specifies an archive set named `vault_tapes`. Two copies are created, but the disk cache associated with this archive set is not released until both copies are made. This scenario can be used at a site that requires online access to files before creating volumes for offsite storage.

**Figure 18. An `archiver.cmd` File Using the `-norelease` Directive**

```
vault_tapes
      1 -norelease 10m
      2 -norelease 30d
```

Note that the `-norelease` specification on a single copy has no effect on automatic releasing because the file cannot be released until it has at least one archive copy. Also, the `-norelease` and `-release` specifications are mutually exclusive.

## Setting the Archive Age

You can set the archive age for files by specifying the archive age as the next field on the directive. The archive age can be specified with a suffix character such as `h` for hours or `m` for minutes. [Table 32 on page 61](#), shows the complete list of suffix characters and their meanings.

In [Figure 19](#), the files in directory `data` are archived when their archive age reaches one hour.

**Figure 19. An archiver.cmd File that Specifies the Archive Age**

```
ex_set data
  1 1h
```

## Unarchiving Automatically

If you specify more than one archive copy of a file, it is possible to unarchive all but one of the copies automatically. This might occur when the files are archived to various media using various archive ages.

[Figure 20](#) shows directives that specify the unarchive age.

**Figure 20. An archiver.cmd File that Specifies the Unarchive Age**

```
ex_set home/users
  1 6m 10w
  2 10w
  3 10w
```

The first copy of the files in the path `home/users` is archived six minutes after modification. When the files are 10 weeks old, second and third archive copies are made. The first copy is then unarchived.

For more ways to control unarchiving, see [“Controlling Unarchiving” on page 89](#).

## Specifying More Than One Copy for Metadata

If more than one copy of metadata is required, you can place copy definitions in the directive file immediately after an `fs=` directive.

[Figure 21](#) shows an `archiver.cmd` file that specifies multiple metadata copies.

**Figure 21. An archiver.cmd File that Specifies Multiple Metadata Copies**

```
fs = samfs7
  1 4h
  2 12h
```

In this example, copy 1 of the metadata for the `samfs7` file system is made after four hours and a second copy is made after 12 hours.

File system metadata includes changes to path names in the file system. For this reason, if you have frequent changes to directories, new archive copies are created. This results in frequent loads of the volumes specified for metadata.

## Archive Set Copy Parameters

The archive set parameters section of the `archiver.cmd` file begins with the `params` directive and ends with the `endparams` directive. [Figure 22](#) shows the format for directives for an archive set.

**Figure 22. Archive Set Copy Parameter Format**

```
params
archive_set_name.copy_number[R] [ -param1 -param2 ...]
.
.
.
endparams
```

**Table 43. Arguments for the Archive Set Copy Parameters**

Argument	Meaning
<i>archive_set_name</i>	A site-defined name for the archive set. Usually indicative of the characteristics of the files belonging to the archive set. Can be allsets. Archive set names are restricted to the letters in the alphabet, numbers, and the underscore character (_). No other special characters or spaces are allowed. The first character in the archive set name must be a letter.
.	A period (.) character. Used to separate the <i>archive_set_name</i> from the <i>copy_number</i> .
<i>copy_number</i>	An integer number that defines the archive copy number. Can be 1, 2, 3, or 4.
R	Specifies that the parameters being defined are for rearchived copies of this archive set. For example, you can use the R and specify VSNs in the <i>-param1</i> argument to direct rearchived copies to specific volumes.
<i>-param1</i> <i>-param2</i>	One or more parameters. The following subsections describe the parameters than can be specified between the <code>params</code> and <code>endparams</code> directives.

The pseudo archive set `allsets` provides a way to set default archive set directives for all archive sets. All `allsets` directives must precede those for actual archive set copies. Parameters set for individual archive set copies



override parameters set by the `allsets` directive. For more information on the `allsets` archive set, see the `archiver.cmd(4)` man page.

All archive set processing parameters are described in this section, with the exception of the `-disk_archive` parameter. For information on the `disk_archive` parameter, see [“Disk Archiving” on page 99](#).

### Controlling the Size of Archive Files: `-archmax`

The `-archmax` directive sets the maximum file size for an archive set. The format is as follows:

```
-archmax target_size
```

This directive is very similar to the `archmax` global directive. For information on this directive, and the values to enter for `target_size`, see [“The `archmax` Directive: Controlling the Size of Archive Files” on page 64](#).

### Setting the Archiver Buffer Size: `-bufsize`

By default, a file being archived is stored in memory in a buffer prior to writing the file to archive media. You can use the `-bufsize` parameter to specify a nondefault buffer size. These actions can improve performance, and you can experiment with various `buffer_size` values.

This parameter has the following format:

```
-bufsize=buffer_size
```

For `buffer_size`, specify a number from 2 through 32. The default is 4. This value is multiplied by the `dev_blksize` value for the media type, and the resulting buffer size is used. The `dev_blksize` is specified in the `defaults.conf` file. For more information on this file, see the `defaults.conf(4)` man page.

For example, this parameter can be specified in the `archiver.cmd` file in a line such as the following:

```
myset.1 -bufsize=6
```

The equivalent of this directive can also be specified on a global basis by specifying the `bufsize=media buffer_size` directive. For more information on this topic, see [“The `bufsize` Directive: Setting the Archiver Buffer Size” on page 64](#).

### Specifying the Number of Drives for an Archive Request: `-drivemax`, `-drivemin`, and `-drives`

By default, the archiver uses only one media drive to archive one archive set's files. When an archive set has a many files or large files, it can be

advantageous to use more than one drive. In addition, if the drives in your automated library operate at different speeds, using these directives can contribute to archiving efficiency.

Figure 23 shows the parameters you can use to split archive requests across drives and to balance variations in tape drive transfer speeds.

**Figure 23. Formats for the -drivemax, -drivemin, and -drives Directives**

```

-drivemax max_size
-drivemin min_size
-drives number
```

**Table 44. Arguments for the -drivemax, -drivemin, and -drives Parameters**

Argument	Meaning
<i>maxsize</i>	The maximum amount of data to be archived using one drive.
<i>minsize</i>	The minimum amount of data to be archived using one drive. The default is the -archmax <i>target_size</i> value (if specified) or the default value for the media type.  If you specify the -drivemin <i>minsize</i> parameter, StorageTek ASM uses multiple drives only if there is enough work to warrant it. As a guideline, you can set <i>minsize</i> to be large enough to cause the transfer time to be significantly longer than the cartridge change time (load, position, unload).
<i>number</i>	The number of drives to use when archiving this archive set. The default is 1.

An archive request is evaluated against the parameters that are specified, as follows:

- If an archive request is less than *min\_size*, only one drive is used to write an archive request.
- If an archive request is larger than *min\_size*, the archive request is evaluated against *min\_size* and the appropriate number of drives is scheduled up to the full number of drives specified.
- If *min\_size* is zero, an attempt is made to split among the full number of drives specified.

When you use the -drives parameter, multiple drives are used only if data that is more than the *min\_size* is to be archived at once. The number of drives to be used in parallel is the lesser of *arch\_req\_total\_size/min\_size* and the number of drives specified by the -drives parameter.

To set these parameters, users need to consider file creation rates, the number of drives, the time it takes to load and unload drives, and drive transfer rates.

**Example 1.** You can use the `-drivemin` and `-drives` parameters if you want to divide an archive request among drives, but you want to avoid tying up all the drives with small archive requests. This might apply to operations that use very large files.

**Example 2.** Assume that you are splitting an archive set named `big_files` over five drives. Depending on its size, this archive set could be split as shown in [Table 45](#).

**Table 45. Archive Set Example Split**

Archive Set Size	Number of Drives
< 20 gigabytes	1
≥ 20 gigabytes to < 30 gigabytes	2
≥ 30 gigabytes to < 40 gigabytes	3
≥ 40 gigabytes to < 50 gigabytes	4
≥ 50 gigabytes	5

[Figure 24](#) shows the lines to use in the `archiver.cmd` file to split the archive request over multiple drives.

**Figure 24. Directives Used to Split an Archive Request Over Multiple Drives**

```
params
bigfiles.1 -drives 5 -drivemin 10G
endparams
```

**Example 3.** The following line is specified in the `archiver.cmd` file:

```
huge_files.2 -drives 2
```

When the total size of the files in archive set `huge_files.2` is equal to or greater than two times `drivemin` for the media, two drives are used to archive the files.

**Example 4.** An archive request is 300 gigabytes. The following line is specified in the `archiver.cmd` file to archive 10 gigabytes at a time on each of 5 drives:

```
-drives 5 -drivemax 10G
```

## Maximizing Space on a Volume: `-fillvsns`

By default, the archiver uses all volumes assigned to an archive set when it writes archive copies. When writing the archive copies, the archiver selects a volume with enough space for all the files. This action can result in volumes not being filled to capacity. If `-fillvsns` is specified, the archiver separates the archive request into smaller groups.

## Specifying Archive Buffer Locks: `-lock`

By default, a file being archived is stored in memory in a buffer prior to writing the file to archive media. If direct I/O is enabled, you can use the `-lock` parameter to lock this buffer. This action can improve performance, and you can experiment with this parameter.

This parameter has the following format:

```
-lock
```

The `-lock` parameter indicates whether or not the archiver should use locked buffers when making archive copies. If `-lock` is specified, the archiver sets file locks on the archive buffer in memory for the duration of the `sam-arcopy(1M)` operation. This avoids paging the buffer, and it can improve performance.

The `-lock` parameter should be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition.

The `-lock` parameter is effective only if direct I/O is enabled for the file being archived. By default, `-lock` is not specified and the file system sets the locks on all direct I/O buffers, including those for archiving. For more information on enabling direct I/O, see the `setfa(1)` man page, the `sam_setfa(3)` library routine man page, or the `-O forcedirectio` option on the `mount_samfs(1M)` man page.

For example, this parameter can be specified in the `archiver.cmd` file in a line such as the following:

```
yourset.3 -lock
```

You can also specify the equivalent of this parameter on a global basis by specifying the `lock` argument to the `bufsize=media buffer_size [lock]` directive. For more information on this topic, see [“The bufsize Directive: Setting the Archiver Buffer Size” on page 64](#).

## Making Archive Copies of Offline Files: `-offline_copy`

A file is a candidate for being released after one archive copy is made. If the file releases and goes offline before all the archive copies are made, the

archiver uses this parameter to determine the method to be used when making the other archive copies. In choosing the *method* to be used, consider the number of drives available to the StorageTek ASM system and the amount of disk cache available. This parameter has the following format:

```
-offline_copy method
```

Specify one of the keywords shown in for *method*.

**Table 46. Values for the `-offline_copy` Directive's *method* argument**

<i>method</i>	<b>Meaning</b>
none	Stages files as needed for each file before copying to the archive volume. Default.
direct	Copies files directly from the offline volume to the archive volume without using the cache. This method assumes that the source volume and the destination volume are different volumes and that two drives are available. If this method is specified, raise the value of the <code>stage_n_window</code> mount option to a value that is greater than its default of 256 kilobytes. For more information on mount options, see the <code>mount_samfs(1M)</code> man page.
stageahead	Stages one file while archiving another. When specified, the system stages the next archive file while writing a file to its destination.
stageall	Stages all files to disk cache before archiving. This method uses only one drive and assumes that room is available on disk cache for all files.

## Specifying Recycling

The recycling process allows you to reclaim space on archive volumes that is taken up by expired archive images. By default, no recycling occurs.

If you want to recycle, you can specify directives in both the `archiver.cmd` file and in the `recycler.cmd` file. For more information on the recycling directives supported in the `archiver.cmd` file, see ["Recycling" on page 153](#).

## Associative Archiving: `-join`

The archiver employs associative archiving if you specify the `-join path` parameter. Associative archiving is useful if you want an entire directory to be archived to one volume and you know that the archive file can physically reside on only one volume. Otherwise, if you want to keep directories together, use either the `-sort path` or `-rsort path` parameters to keep the files contiguous. The `-rsort` performs a reverse sort.

When the archiver writes an archive file to a volume, it efficiently packs the volume with user files. Subsequently, when accessing files from the same directory, you can experience delays as the stage process repositions through a volume to read the next file. To alleviate delays, you can archive files from the same directory paths contiguously within an archive file. The process of associative archiving overrides the space efficiency algorithm to archive files from the same directory together. The `-join path` parameter allows these files to be archived contiguously within an archive set copy.

Associative archiving is useful when the file content does not change but you want to access the group of files together at the same time all the time. For example, you might use associative archiving at a hospital for accessing medical images. Images associated with the same patient can be kept in a directory and the doctor might want to access those images together at one time. These static images can be accessed more efficiently if you archive them contiguously based on their directory location. For example:

```
patient_images.1 -join path
```

**Note:** The `-join path` parameter writes data files from the same directory to the same archive file. If there are many directories with a few small files, the archiver creates many small archive files. These small, discrete archive files slow the write performance of the system because the data files are relatively small compared to the `tar(1)` header for each archive file. This can impair performance when writing to high-speed tape drives.

Also, because the `-join path` parameter specifies that all the files from the same directory be archived on a single volume, it is possible that a group of files might not fit on any available volume. In this case, the files are not archived until more volumes are assigned to the archive set. It is also possible that the group of files to be archived is so large that it can never fit on a single volume. In such a case, the files are never archived.

For most applications, using either `-sort path` or `-join path` parameter is preferred if the more restrictive operation of `-join path` is not a requirement.

It is also possible to sort files within an archive set copy by age, size, or path. The `age` and `size` arguments are mutually exclusive. [Figure 25](#) shows how to sort an archive set using the `-sort` parameter with the argument `age` or `size`.

**Figure 25. Directives for Sorting an Archive Set**

```
cardiac.1 -sort path
cardiac.2 -sort age
catscans.3 -sort size
```

The first line forces the archiver to sort an archive request by path name. The second example line forces the archiver to sort an archive set copy called `cardiac.2` by the age of the file, oldest to youngest. The third line forces the archive set copy called `catiscans` to be sorted by the size of the file, smallest to largest. If you had wanted a reverse sort, you could have specified `-rsort` in place of `-sort`.

## Controlling Unarchiving

*Unarchiving* is the process by which archive entries for files or directories are deleted. By default, files are never unarchived. Files are unarchived based on the time since last access. All frequently accessed data can be stored on a fast media, such as disk, and all older, infrequently accessed data can be stored on tape.

**Example 1.** Figure 26. shows an `archiver.cmd` file.

**Figure 26. Directives to Control Unarchiving**

```

arset1 dir1
  1    10m    60d
  2    10m
  3    10m
vsns
arset1.1    mo    OPT00[0-9]
arset1.2    lt    DLTA0[0-9]
arset1.3    lt    DLTB0[0-9]

```

If `archiver.cmd` file shown in Figure 26. controls a file that is accessed frequently, it remains on disk all the time, even if it is older than 60 days. The copy 1 information is removed only if the file is not accessed for 60 days.

If the copy 1 information is removed (because the file was not accessed for 60 days) and someone stages the file from copy 2, it is read from tape. After the file is back online, the archiver makes a new copy 1 on disk and the 60-day access cycle starts all over again. The StorageTek ASM archiver regenerates a new copy 1 if the file is accessed again.

### **Example 2**

Assume that a patient is in the hospital for four weeks. During this time, all of this patient's files are on fast media (copy 1=`mo`). After four weeks, the patient is released from the hospital. If no data has been accessed for this patient for up to 60 days after the patient is released, the copy 1 entry in the inode is unarchived, and only copy 2 and copy 3 entries are available. The volume can now be recycled in order to make room for more current patients without having to increase the disk library. If the patient comes back to the hospital after six months for a checkup, the first access of the data is from tape (copy 2). Now the archiver automatically creates a new copy 1 on disk to ensure that the data is back on the fast media during the checkup, which could take several days or weeks.

## Controlling How Archive Files are Written: `-tapenonstop`

By default, the archiver writes a tape mark, an EOF label, and two more tape marks between archive files. When the next archive file is started, the driver backs up to the position after the first tape mark, causing a loss of performance. The `-tapenonstop` parameter directs the archiver to write only the initial tape mark. In addition, if the `-tapenonstop` parameter is specified, the archiver enters the archive information at the end of the copy operation.

For more information on the `-tapenonstop` parameter, see the `archiver.cmd(4)` man page.

## Reserving Volumes: `-reserve`

By default, the archiver writes archive set copies to any volume specified by a regular expression as described in the volume associations section of the `archiver.cmd` file. However, it is sometimes desirable for archive set volumes to contain files from only one archive set. The process of reserving volumes can be used to satisfy this data storage requirement.

**Note:** The `-reserve` parameter reserves a volume for exclusive use by one archive set. A site that uses reserved volumes is likely to incur more cartridge loads and unloads.

The `-reserve` parameter reserves volumes for an archive set. When the `-reserve` parameter is set and a volume has been assigned to an archive set copy, the volume identifier is not assigned to any other archive set copy, even if a regular expression matches it.

As volumes are selected for use by an archive set, a reserved name is assigned to the volume. The reserved name is a unique identifier that ties the archive set to the volume.

The format for the `-reserve` parameter is as follows:

```
-reserve keyword
```

The *keyword* specified depends on the form you are using. The possible forms are archive set form, owner form, and file system form, as follows:

- Archive set form. This form uses the *set keyword*, as follows: `-reserve set`
- Owner form. This form uses one of the following *keywords*: `dir`, `user`, or `group`. [Figure 27](#) shows the formats for these directives.

**Figure 27. Owner Forms for the `-reserve` Parameter**

```
-reserve dir
-reserve user
-reserve group
```



The three owner forms shown in [Figure 27](#) are mutually exclusive. That is, only one of the three owner forms can be used on an archive set and copy.

- File system form. This form uses the *fs keyword*, as follows: `-reserve fs`

In the `archiver.cmd` file, you can specify a `-reserve` parameter for one, two, or all three possible forms. The three forms can be combined and used together in an archive set parameter definition.

For example, [Figure 28](#) shows an `archiver.cmd` file fragment. The line that begins with `arset.1` creates a reserved name based upon an archive set, a group, and the file system.

**Figure 28. An archiver.cmd File With Reserved Volumes**

```
params
arset.1 -reserve set -reserve group -reserve fs
endparams
```

The information regarding reserved volumes is stored in the library catalog. The lines within the library catalog contain the media type, the VSN, the reserve information, and the reservation date and time. The reserve information includes the archive set component, path name component, and file system component, separated by slashes (`//`).

These slashes are *not* indicative of a path name; they are merely separators for displaying the three components of a reserved name. As [Figure 29](#) shows, the lines that describe reserved volumes begin with `#R` characters in the library catalog.

**Figure 29. Library Catalog Showing Reserved Volumes**

```
6 00071 00071 lt 0xe8fe 12 9971464 1352412 0x6a000000 131072 0x
# -il-o-b----- 05/24/00 13:50:02 12/31/69 18:00:00 07/13/01 14:03:00
#R lt 00071 arset0.3// 2001/03/19 18:27:31
10 ST0001 NO_BAR_CODE lt 0x2741 9 9968052 8537448 0x68000000 1310
# -il-o----- 05/07/00 15:30:29 12/31/69 18:00:00 04/13/01 13:46:54
#R lt ST0001 hgm1.1// 2001/03/20 17:53:06
16 SLOT22 NO_BAR_CODE lt 0x76ba 6 9972252 9972252 0x68000000 1310
# -il-o----- 06/06/00 16:03:05 12/31/69 18:00:00 07/12/01 11:02:05
#R lt SLOT22 arset0.2// 2001/03/02 12:11:25
```

Note that some lines in [Figure 29](#) have been truncated to fit on the page.

One or more of the reserve information fields can be empty, depending on the options defined in the `archiver.cmd` file. The date and time indicate when the reservation was made. A reservation line is appended to the file for each volume that is reserved to an archive set during archiving.

You can display the reserve information by using the `samu(1M)` utility's `v` display or by using the `archiver(1M)` or `dump_cat(1M)` commands in one of the formats shown in [Figure 30](#).

**Figure 30. Commands to Use to Display the Reserve Information**

```
archiver -lv
dump_cat -V catalog_name
```

The following formats illustrate each form showing the parameter, keywords, and examples of reserved names assigned to volumes.

- Archive set form. As [Table 47](#) shows, the `set` keyword activates the archive set component in the reserved name.

**Table 47. Archive Set Form Examples**

Directive and Keyword	Reserved Name Examples
-reserve set	users.1// Data.1//

For example, in [Figure 31](#), the `archiver.cmd` file fragment shows that the line that begins with the `allsets` archive set name sets reserve by archive set for all archive sets.

**Figure 31. Reserving Volumes by Archive Set**

```
params
allsets -reserve set
endparams
```

- Owner form. The `dir`, `user`, and `group` keywords activate the owner component in the reserved name. The `dir`, `user`, and `group` keywords are mutually exclusive. The `dir` keyword uses the directory path component immediately following the path specification of the archive set definition. The `user` and `group` keywords are self-explanatory. [Table 48](#) shows examples.

**Table 48. Owner Set Form Examples**

Directive and Keyword	Reserved Name Examples
-reserve dir	proj.1/p105/ proj.1/p104/
-reserve user	users.1/user5/ users.1/user4/
-reserve group	data.1/engineering/

**Note:** The `-reserve` parameter is intended to reserve a volume for exclusive use by one archive set. Many directories with a few small files cause many small archive files to be written to each reserved volume. These small discrete archive files slow the performance of the system because data files are relatively small compared to the `tar(1)` header for each archive file.

- File system form. The `fs` keyword activates the file system component in the Reserved Name. [Table 49](#) shows examples.

**Table 49. File System Form Examples**

Directive and Keyword	Reserved Name Examples
<code>-reserve fs</code>	<code>proj.1/p103/samfs1</code> <code>proj.1/p104/samfs1</code>

“[Example 4](#)” on [page 113](#) shows a complete archive example using reserved volumes.

The archiver records volume reservations in the library catalog files. A volume is automatically unreserved when it is relabeled because the archive data has been effectively erased.

You can also use the `reserve(1M)` and `unreserve(1M)` commands to reserve and unreserve volumes. For more information on these commands, see the `reserve(1M)` and `unreserve(1M)` man pages.

## Setting Archive Priorities: `-priority`

The StorageTek ASM file systems offer a configurable priority system for archiving files. Each file is assigned a priority computed from properties of the file and priority multipliers that can be set for each archive set in the `archiver.cmd` file. Properties include online/offline, age, number of copies made, and size.

By default, the files in an archive request are not sorted and all property multipliers are zero. This results in files being archived in first found, first archived order. For more information on priorities, see the `archiver(1M)` and `archiver.cmd(4)` man pages.

You can control the order in which files are archived by setting priorities and sort methods. The following are examples of priorities that you can set:

- Select the `priority` sort method to archive files within an archive request in priority order.
- Change the `archive_loaded` priority to reduce media loads.
- Change the `offline` priority to cause online files to be archived before offline files.

- Change the `copy#` priorities to make archive copies in copy order.

Table 50 lists the archive priorities.

**Table 50. Archive Priorities**

Archive Priority	Definition
<code>-priority age value</code>	Archive age property multiplier
<code>-priority archive_immediate value</code>	Archive immediate property multiplier
<code>-priority archive_overflow value</code>	Multiple archive volumes property multiplier
<code>-priority archive_loaded value</code>	Archive volume loaded property multiplier
<code>-priority copy1 value</code>	Copy 1 property multiplier
<code>-priority copy2 value</code>	Copy 2 property multiplier
<code>-priority copy3 value</code>	Copy 3 property multiplier
<code>-priority copy4 value</code>	Copy 4 property multiplier
<code>-priority copies value</code>	Copies made property multiplier
<code>-priority offline value</code>	File offline property multiplier
<code>-priority queuwait value</code>	Queue wait property multiplier
<code>-priority rearchive value</code>	Rearchive property multiplier
<code>-priority reqrelease value</code>	Reqrelease property multiplier
<code>-priority size value</code>	File size property multiplier
<code>-priority stage_loaded value</code>	Stage volume loaded property multiplier
<code>-priority stage_overflow value</code>	Multiple stage volumes property multiplier

For *value*, specify a floating-point number in the following range:

```
-3.400000000E+38  value  3.402823466E+38
```

**Scheduling Archiving: `-startage`, `-startcount`, and `-startsize`**

As the archiver scans a file system, it identifies files to be archived. Files that are recognized as candidates for archiving are placed in a list known as an *archive request*. At the end of the file system scan, the system schedules the archive request for archiving. The `-startage`, `-startcount`, and `-startsize` archive set parameters control the archiving workload and

assure timely archival of files. [Table 51](#) shows the formats for these parameters.

## Formats

**Table 51. The -startage, -startcount, and -startsize Directive**

Directive	Meaning
-startage <i>time</i>	Specifies the amount of <i>time</i> that can elapse between the first file in a scan being marked for inclusion in an archive request and the start of archiving. For <i>time</i> , specify a time in the format used in <a href="#">“Setting the Archive Age” on page 81</a> .
-startcount <i>count</i>	Specifies the number of files to be included in an archive request. When the number of files in the archive request reaches <i>count</i> , archiving begins. Specify an integer number for <i>count</i> . By default, <i>count</i> is not set.
-startsize <i>size</i>	Specifies the minimum total size, in bytes, of all files to be archived in an archive request. Archiving work is accumulated and archiving begins when the total size of the files reaches <i>size</i> . By default, <i>size</i> is not set.

The `examine=method` directive and the `interval=time` directives are global directives that interact with the `-startage`, `-startcount`, and `-startsize` directives. The `-startage`, `-startcount`, and `-startsize` directives optimize archive timeliness versus archive work done. These values override the `examine=method` specification, if any. For more information on the `examine` directive, see [“The examine Directive: Controlling Archive Scans” on page 66](#). For more information on the `interval` directive, see [“The interval Directive: Specifying an Archive Interval” on page 67](#).

Only one of the `-startage`, `-startcount`, and `-startsize` directives can be specified in an `archiver.cmd` file. If more than one of these directives is specified, the first condition encountered starts the archival operation. If neither `-startage`, `-startcount`, nor `-startsize` are specified, the archive request is scheduled based on the `examine=method` directive, as follows:

- If `examine=noscan`, the archive request is scheduled according to the `interval=time` directive’s specification after the first file is entered in the archive request. This is continuous archiving. By default, `examine=noscan`.
- If `examine=scan | scaninodes | scandirs`, the archive request is scheduled for archiving after the file system scan.

The `archiver.cmd(4)` man page has examples that show how to use these directives.

## VSN Association Directives

The VSN associations section of the `archiver.cmd` file assigns volumes to archive sets. This section starts with a `vsns` directive and ends with an `endvsns` directive.

Collections of volumes are assigned to archive sets by directives of the following form:

```
archive_set_name.copy_num media_type vsn_expr ... [ -pool vsn_pool_name ... ]
```

**Table 52. Arguments for the VSN Association Directive**

Argument	Meaning
<i>archive_set_name</i>	A site-defined name for the archive set. Must be the first field in the archive set assignment directive. An archive set name is usually indicative of the characteristics of the files belonging to the archive set. Archive set names are restricted to the letters in the alphabet, numbers, and the underscore character ( <code>_</code> ). No other special characters or spaces are allowed. The first character in the archive set name must be a letter.
<i>copy_num</i>	A digit that is followed by one or more arguments that specify archive characteristics for that copy. Archive copy directives begin with a digit. This digit (1, 2, 3, or 4) is the copy number.
<i>media_type</i>	The media type. For a list of valid media types, see the <code>mcf(4)</code> man page.
<i>vsn_expr</i>	A regular expression. See the <code>regex(5)</code> man page.
<code>-pool</code> <i>vsn_pool_name</i>	A named collection of VSNs.

An association requires at least three fields: the *archive\_set\_name* and *copy\_number*, the *media\_type*, and at least one volume. The *archive\_set\_name* and *copy\_number* are connected by a period (`.`).

The following examples specify the same VSNs in different ways.

**Example 1.** Figure 32 shows two lines of VSN specifications.

**Figure 32. VSN Specifications - Example 1**

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005
set.1 lt VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

**Example 2.** Figure 33 shows a VSN specification that uses a backslash character (\) to continue a line onto a subsequent line.

**Figure 33. VSN Specifications - Example 2**

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005 \
  VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

**Example 3.** Figure 34 specifies VSNs using a regular expression in a shorthand notation.

**Figure 34. VSN Specifications - Example 3**

```
vsns
set.1 lt VSN0[1-9] VSN10
endvsns
```

The volumes are noted by one or more *vsn\_expression* keywords, which are regular expressions as described in the `regexp(5)` man page. Note that these regular expressions do not follow the same conventions as wildcards. In addition to a regular expression, you can also specify VSN pools from which volumes are to be selected. Pools are expressed with the `-pool vsn_pool_name` directive with a VSN association.

When the archiver needs volumes for the archive set, it examines each volume of the selected media type in all automated libraries and manually mounted drives to determine if it would satisfy any VSN expression. It selects the first volume that fits an expression that contains enough space for the archive copy operation. For example:

- The following directive specifies that files belonging to archive set `ex_set` for copy 1 be copied to media type `mo` using any of the twenty volumes with the name `optic20` through `optic39`:

```
ex_set.1 mo optic[2-3][0-9]
```

- The following directive copies files belonging to archive set `ex_set` for copy 2 to media type `lt` with any volume beginning with `TAPE`:

```
ex_set.2 lt ^TAPE
```

If your StorageTek ASM environment is configured to recycle by archive set, do not assign a VSN to more than one archive set.

**Note:** Make sure you assign volumes to the archive set for the metadata when setting up the `archiver.cmd` file. Each file system has an archive set with the same name as the file system. For more information on preserving metadata, see the `samfsdump(1M)` man page or see the *StorageTek ASM, StorageTek QFS, and StorageTek SM QFS Disaster Recovery Guide QFS, StorageTek ASM, and StorageTek ASM QFS Disaster Recovery Guide*.

## VSN Pools Directives

The VSN pools section of the `archiver.cmd` file starts with a `vsnpools` directive and ends with either an `endvsnpools` directive or with the end of the `archiver.cmd` file. This section names a collection of volumes.

A *VSN pool* is a named collection of volumes. VSN pools are useful for defining volumes that can be available to an archive set. As such, VSN pools provide a useful buffer for assigning volumes and reserving volumes to archive sets.

You can use VSN pools to define separate groups of volumes for use by departments within an organization, users within a group, data types, and other convenient groupings. The pool is assigned a name, media type, and a set of volumes. A *scratch pool* is a set of volumes used when specific volumes in a VSN association are exhausted or when another VSN pool is exhausted. For more information on VSN associations, see [“VSN Association Directives” on page 96](#).

If a volume is reserved, it is no longer available to the pool in which it originated. Therefore, the number of volumes within a named pool changes as volumes are used. You can view the VSN pools by entering the `archiver(1M)` command in the following format:

```
# archiver -lv | more
```

A VSN pool definition requires at least three fields separated by white space: the pool name, the media type, and at least one VSN. The syntax is as follows:

```
vsn_pool_name media_type vsn_expression
```



**Table 53. Arguments for the VSN Pools Directive**

Argument	Meaning
<i>vsn_pool_name</i>	Specifies the VSN pool
<i>media_type</i>	The 2-character media type. For a list of valid media types, see the <code>mcf(4)</code> man page.
<i>vsn_expression</i>	Regular expression. There can be one or more <i>vsn_expression</i> arguments. See the <code>regcmp(3G)</code> man page.

The following example uses four VSN pools: `users_pool`, `data_pool`, `proj_pool`, and `scratch_pool`. If one of the three specific pools is out of volumes, the archiver selects the scratch pool VSNs. [Figure 35](#) shows an `archiver.cmd` file that uses four VSN pools.

**Figure 35. Example Showing VSN Pools**

```

vsnpools
users_pool    mo  ^MO[0-9][0-9]
data_pool     mo  ^DA.*
scratch_pool  mo  ^SC[5-9][0-9]
proj_pool     mo  ^PR.*
endvsnpools

vsns
users.1      mo  -pool users_pool    -pool scratch_pool
data.1       mo  -pool data_pool     -pool scratch_pool
proj.1       mo  -pool proj_pool     -pool scratch_pool
endvsns

```

## ■ Disk Archiving

Archiving is the process of copying a file from online disk to archive media. Often, archive copies are written to volumes on magneto optical or tape cartridges in an automated library, but with disk archiving, online disk in a file system is used as archive media.

Disk archiving can be implemented so that the files are archived from one StorageTek ASM file system to another file system on the same host computer system. Disk archiving can also be implemented so the source files are archived to another file system on a different Sun Solaris system. When disk archiving is implemented using two host systems, the systems involved act as a client and a server. The *client system* is the system that hosts the source files. The *server system* is the destination system that hosts the archive copies.

The file system to which the archive files are written can be any UNIX file system. It does not have to be a StorageTek ASM file system. If disk archive copies are written to a different host, the host must have at least one StorageTek ASM file system installed upon it.

The archiver treats files archived to disk volumes the same as it treats files archived to volumes in a library. You can still make one, two, three, or four archive copies. If you are making multiple archive copies, one of the archive copies could be written to disk volumes while the others are written to removable media volumes. In addition, if you typically archive to disk volumes in a StorageTek ASM file system, the archive file copies are themselves archived according to the `archiver.cmd` file rules in that file system.

The following list summarizes some of the similarities and differences between archiving to online disk and archiving to removable media:

- Unlike archive copies written to a magneto optical disk or to a tape, archive copies written to disk are not recorded in a catalog. In addition, archive files in disk volumes do not appear in the historian.
- If you are archiving to removable media volumes, you can begin archiving after the file system is mounted without changing any of the default values in the `archiver.cmd` file. If you are archiving to disk volumes, however, you must edit the `archiver.cmd` file and define disk archive sets prior to mounting the file system.
- Disk archiving does not rely on entries in the `mcf(4)` file. You need to specify the `-disk_archive` parameter in the `archiver.cmd` file, and you need to define disk volumes in `/etc/opt/SUNWsamfs/diskvols.conf`. This is an additional configuration file, and it is not needed if you are archiving to removable media volumes only.

A `diskvols.conf` file must be created on the system upon which the source files reside. Depending on where the archive copies are written, this file also contains the following information:

- If the archive copies are written to a file system on that same host system, the `diskvols.conf` file defines the VSNs and the paths to each VSN.
- If the archive copies are written to a different Sun Solaris system, the `diskvols.conf` file contains the host name of that server system. In this case, there must also be a `diskvols.conf` file on the server system that defines clients that are given permission to write to that system. If you want to create this client/server relationship, make sure that the host acting as the server has at least one StorageTek ASM file system installed up on it before starting the procedure called [“To Enable Disk Archiving” on page 102](#).

## Configuration Guidelines

While there are no restrictions on where disk archive volumes can reside, it is recommended that the disk volumes reside on a disk other than the one upon which the original files reside. Preferably, the archive copies from a client system would be written to disk volumes on a server system. It is

recommended that you make more than one archive copy and write to more than one type of archive media. For example, copy 1 could be archived to disk volumes, copy 2 to tape, and copy 3 to magneto-optical disk.

If you are archiving files to a file system on a server system, the archive files themselves can be archived to removable media cartridges in a library attached to the destination server.

## Directives for Disk Archiving

When archiving to online disk, the archiver recognizes most of the `archiver.cmd` directives. The directives it recognizes define the archive set and configure recycling. The directives that are silently ignored are those that are meaningless in a disk archiving environment because they specifically pertain to working with removable media cartridges. Specifically, the system recognizes the following directives for disk archive sets:

- All the recycling directives in “[Archive Set Copy Parameters](#)” on page 82 except for the following:
  - `-fillvsns`
  - `-ovflmin min_size`
  - `-reserve method`
  - `-tapenonstop`
- All the directives in “[Step 2: Editing the archiver.cmd File \(Optional\)](#)” on page 161 except for the following:
  - `-recycle_dataquantity size`
  - `-recycle_vsncount count`
- The `-disk_archive` parameter. This is an archive set processing parameter. You must specify `-disk_archive` parameters in the `archiver.cmd` file to define disk archive set. The archiver uses this parameter to maintain the file system hierarchy of the data as it is written to the archive disk’s mount point. Like all archive set processing parameters, it must be specified between the `params` and `endparams` directives. [Figure 36](#) shows the format of this directive.

**Figure 36. Format for the `-disk_archive` Parameter**

```
params
archive_set.copy_number -disk_archive VSN_Name
endparams
```

For `VSN_Name`, specify a VSN that is defined in the `diskvols.conf` file.

- The `clients` and `endclients` directives. If you implement disk archiving such that you archive source files from a client host to a server

host, you must configure a `diskvols.conf` file on the server host. The `diskvols.conf` file on the server system must contain the name of the client system. The format for these directives is as follows:

**Figure 37. Format for the `clients` and `endclients` Directive**

```
clients
client_system1
client_system2
...
endclients
```

For `client_system`, specify the hostname of the client system that contains the source files.

For more information on directives for disk archiving, see the `archiver.cmd(4)` man page.

## To Enable Disk Archiving

You can enable disk archiving at any time. The procedure in this section assumes that you have archiving in place and you are adding disk archiving to your environment. If you are enabling disk archiving as part of an initial installation, see the *StorageTek QFS and StorageTek ASM Software Installation and Configuration Guide* for information. Do not use this procedure because it contains steps that are not needed if you add disk archiving at installation time.

1. Make certain that the host to which you want to write your disk archive copies has at least one StorageTek ASM file system installed on it.
2. Become superuser on the host system that contains the files to be archived.
3. Follow the procedures in the *StorageTek QFS and StorageTek ASM Software Installation and Configuration Guide* for enabling disk archiving.

The StorageTek ASM initial installation procedure contains a step called *Enabling Disk Archiving*. That step is broken down into two procedures.

4. Become superuser on the host that contains the files to be archived.
5. On the host that contains the files to be archived, use the `samd(1M) config` command to propagate the configuration file changes and restart the system.

For example:

```
# samd config
```

6. Become superuser on the host system to which the archive copies are written. (Optional)

Perform this step only if you are archiving to disk on a different host.

7. On the host to which the archive copies will be written, use the `samd(1M) config` command to propagate the configuration file changes and restart the destination system. (Optional)

Perform this step only if you are archiving to disk on a different host.

For example:

```
# samd config
```

## Disk Archiving Examples

### Example 1

Figure 38 shows the `diskvols.conf` file that resides on client system `pluto`.

**Figure 38. The `diskvols.conf` File on `pluto`**

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on pluto
# VSN Name      [Host Name:]Path
#
disk01          /sam_arch1
disk02          /sam_arch2/proj_1
disk03          mars:/sam_arch3/proj_3
```

In the preceding `diskvols.conf` file, VSNs identified as `disk01` and `disk02` are written to the host system upon which the original source files reside. VSN `disk03` is written to a VSN on server system `mars`.

Figure 39 shows the `diskvols.conf` file on server system `mars`.

**Figure 39. The `diskvols.conf` File on `mars`**

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
pluto
endclients
```

Figure 40 shows a fragment of the `archiver.cmd` file on `pluto`.

**Figure 40. The `archiver.cmd` File on `pluto`**

```
params
arset1.2 -disk_archive disk01
arset2.2 -disk_archive disk02
arset3.2 -disk_archive disk03
endparams
```

## Example 2

In this example, file `/sam1/testdir0/filea` is in the archive set for `arset0.1`, and the archiver copies the content of `/sam1/testdir0/filea` to the destination path named `/sam_arch1`. [Figure 41](#) shows the `diskvols.conf` file.

**Figure 41. A `diskvols.conf` File**

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf
#
# VSN Name      [Host Name:]Path
#
disk01          /sam_arch1
disk02          /sam_arch12/proj_1
```

[Figure 42](#) shows the `archiver.cmd` file lines that pertain to disk archiving:

**Figure 42. Directives in the `archiver.cmd` File that Pertain to Disk Archiving**

```
.
.
.
params
arset0.1 -disk_archive disk01
endparams
.
.
.
```

The following shows output from the `sls(1)` command for file `filea`, which was archived to disk. In [Figure 43](#), note the following:

- `dk` is the media type for disk archive media
- `disk01` is the VSN
- `f192` is the path to the disk archive `tar(1)` file

**Figure 43. Output From `sls(1M)`**

```
# sls -D /sam1/testdir0/filea
/sam1/testdir0/filea:
mode: -rw-r----- links: 1 owner: root group: other
length: 797904 admin id: 0 inode: 3134.49
archdone;
copy 1: ---- Dec 16 14:03 c0.1354 dk disk01 f192
access: Dec 19 10:29 modification: Dec 16 13:56
changed: Dec 16 13:56 attributes: Dec 19 10:29
creation: Dec 16 13:56 residence: Dec 19 10:32
```

### Example 3

In this example, file `/sam2/my_proj/fileb` is on client host `snickers` in archive set `arset0.1`, and the archiver copies the content of this file to the destination path `/sam_arch1` on server host `mars`.

Figure 44 shows the `diskvols.conf` file on `snickers`.

**Figure 44. The `diskvols.conf` File on `snickers`**

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on snickers
#
# VSN Name      [Host Name:]Path
#
disk01         mars:/sam_arch1
```

Figure 45 shows the `diskvols.conf` file on `mars`.

**Figure 45. The `diskvols.conf` File on `mars`**

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
snickers
endclients
```

Figure 46 shows the directives in the `archiver.cmd` file that relate to this example.

**Figure 46. Directives in the `archiver.cmd` File that Pertain to Disk Archiving**

```
.
.
.
params
arset0.1 -disk_archive disk01
endparams
.
.
.
```

## ■ Archiver Examples

Table 54 shows the directory structure that all examples in this section use.

**Table 54. Directory Structure Example**

Top-most Directory	1st-level Subdirectory	2nd-level Subdirectory	3rd-level Subdirectory
/sam	/projs	/proj_1	/katie
/sam	/projs	/proj_1	/sara
/sam	/projs	/proj_1	/wendy

**Table 54. Directory Structure Example (Continued)**

Top-most Directory	1st-level Subdirectory	2nd-level Subdirectory	3rd-level Subdirectory
/sam	/projs	/proj_2	/joe
/sam	/projs	/proj_2	/katie
/sam	/users	/bob	
/sam	/users	/joe	
/sam	/users	/katie	
/sam	/users	/sara	
/sam	/users	/wendy	
/sam	/data		
/sam	/tmp		

## Example 1

This example illustrates the action of the archiver when no `archiver.cmd` file is used. In this example, a StorageTek ASM environment includes one file system, an optical automated library with two drives, and six cartridges.

Figure 47 shows the output produced by the `archiver(1M) -lv` command. It shows that the default media selected by the archiver is type `mo`. Only the `mo` media are available.

**Figure 47. archiver(1M) -lv Example Output Part One**

```
# archiver -lv
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh

Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
```

Figure 48 shows output that indicates that the archiver uses two drives. It lists the 12 volumes, storage capacity, and available space.

**Figure 48. archiver(1M) -lv Example Output Part Two**

```
Archive libraries:
Device:hp30 drives_available:2 archive_drives:2
Catalog:
mo.optic00      capacity: 1.2G space: 939.7M -il-o-----
mo.optic01      capacity: 1.2G space: 934.2M -il-o-----
mo.optic02      capacity: 1.2G space: 781.7M -il-o-----
mo.optic03      capacity: 1.2G space: 1.1G -il-o-----
mo.optic10      capacity: 1.2G space: 85.5M -il-o-----
mo.optic11      capacity: 1.2G space: 0 -il-o-----
mo.optic12      capacity: 1.2G space: 618.9k -il-o-----
```



**Figure 48.** archiver(1M) -lv **Example Output Part Two (Continued)**

mo.optic13	capacity:	1.2G	space:	981.3M	-il-o-----
mo.optic20	capacity:	1.2G	space:	1.1G	-il-o-----
mo.optic21	capacity:	1.2G	space:	1.1G	-il-o-----
mo.optic22	capacity:	1.2G	space:	244.9k	-il-o-----
mo.optic23	capacity:	1.2G	space:	1.1G	-il-o-----

Figure 49 shows that both the metadata and data files are included in the archive set samfs. The archiver makes one copy of the files when their archive age reaches the default four minutes (240 seconds).

**Figure 49.** archiver(1M) -lv **Example Output Part Three**

Archive file selections:	
Filesystem	samfs Logfile:
samfs Metadata	
copy:	1 arch_age:240
samfs1 path:..	
copy:	1 arch_age:240

Figure 50 shows the files in the archive sets archived to the volumes in the indicated order.

**Figure 50.** archiver(1M) -lv **Example Output Part Four**

Archive sets:	
allsets	
samfs.1	
media:	mo (by default)
Volumes:	
optic00	
optic01	
optic02	
optic03	
optic10	
optic12	
optic13	
optic20	
optic21	
optic22	
optic23	
Total space available:	8.1G

## Example 2

This example shows how to separate data files into two archive sets separate from the metadata. There is a manually mounted DLT tape drive in addition to the optical automated library from “[Example 2](#)” on page 104. The big files are archived to tape, and the small files are archived to optical cartridges.

Figure 51 shows the content of the archiver.cmd file.

**Figure 51.** archiver(1M) -lv Output Part One Showing the archiver.cmd File

```
# archiver -lv -c example2.cmd
Reading archiver command file "example2.cmd"
1: # Example 2 archiver command file
2: # Simple selections based on size
3:
4: logfile = /var/opt/SUNWsamfs/archiver/log
5: interval = 5m
6:
7: # File selections.
8: big . -minsize 500k
9: all .
10:    1 30s
11:
12: vsns
13: samfs.1 mo .*0[0-2]          # Metadata to optic00 - optic02
14: all.1 mo .*0[3-9] .*[1-2][0-9] # All others for files
15: big.1 lt .*
16: endvsns
```

Figure 52 shows the media and drives to be used, not the addition of the DLT and its defaults.

**Figure 52.** archiver(1M) -lv Output Part Two Showing Media and Drives

```
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00      capacity: 1.2G space: 939.7M -il-o-----
mo.optic01      capacity: 1.2G space: 934.2M -il-o-----
mo.optic02      capacity: 1.2G space: 781.7M -il-o-----
mo.optic03      capacity: 1.2G space: 1.1G -il-o-----
mo.optic04      capacity: 1.2G space: 983.2M -il-o-----
mo.optic10      capacity: 1.2G space: 85.5M -il-o-----
mo.optic11      capacity: 1.2G space: 0 -il-o-----
mo.optic12      capacity: 1.2G space: 618.9k -il-o-----
mo.optic13      capacity: 1.2G space: 981.3M -il-o-----
mo.optic20      capacity: 1.2G space: 1.1G -il-o-----
mo.optic21      capacity: 1.2G space: 1.1G -il-o-----
mo.optic22      capacity: 1.2G space: 244.9k -il-o-----
mo.optic23      capacity: 1.2G space: 1.1G -il-o-----
Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE02      capacity: 9.5G space: 6.2G -il-o-----
lt.TAPE03      capacity: 9.5G space: 3.6G -il-o-----
lt.TAPE04      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE05      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE06      capacity: 9.5G space: 7.4G -il-o-----
```

Figure 53 shows the organization of the file system. Files bigger than 512000 bytes (500 kilobytes) are archived after four minutes; all other files are archived after 30 seconds.

**Figure 53.** archiver(1M) -lv Output Part Three Showing File System Organization

```
Archive file selections:
Filesystem samfs  Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
    copy:1  arch_age:240
big  path:.. minsize:502.0k
    copy:1  arch_age:240
all  path:..
    copy:1  arch_age:30
```

Figure 54 shows the division of the archive sets among the removable media in the following output.

**Figure 54.** archiver(1M) -lv Output Part Four Showing Archive Sets and Removable Media

```
Archive sets:
allsets
all.1
  media: mo
Volumes:
  optic03
  optic04
  optic10
  optic12
  optic13
  optic20
  optic21
  optic22
  optic23
  Total space available: 6.3G
big.1
  media: lt
Volumes:
  TAPE01
  TAPE02
  TAPE03
  TAPE04
  TAPE05
  TAPE06
  Total space available: 42.8G
samfs.1
  media: mo
Volumes:
  optic00
  optic01
  optic02
  Total space available: 2.6G
```

## Example 3

In this example, user files and project data files are archived to various media. Files from the directory `data` are segregated by size to optical and tape media. Files assigned to the group ID `pict` are assigned to another set of volumes. Files in the directories `tmp` and `users/bob` are not archived. Archiving is performed on a 15-minute interval, and an archiving record is kept.

Figure 55 shows this example.

**Figure 55.** `archiver(1M) -lv -c` Command Output

```
# archiver -lv -c example3.cmd
Reading archiver command file "example3.cmd"
1: # Example 3 archiver command file
2: # Segregation of users and data
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: no_archive tmp
8:
9: fs = samfs
10: no_archive users/bob
11: prod_big data -minsize 50k
12:   1 1m 30d
13:   2 3m
14: prod data
15:   1 1m
16: proj_1 projs/proj_1
17:   1 1m
18:   2 1m
19: joe . -user joe
20:   1 1m
21:   2 1m
22: pict . -group pict
23:   1 1m
24:   2 1m
25:
26: params
27: prod_big.1 -drives 2
28: prod_big.2 -drives 2
29: endparams
30:
31: vsns
32: samfs.1 mo optic0[0-1]$
33: joe.1 mo optic01$
34: pict.1 mo optic02$
35: pict.2 mo optic03$
36: proj_1.1 mo optic1[0-1]$
37: proj_1.2 mo optic1[2-3]$
38: prod.1 mo optic2.$
39: joe.2 lt 0[1-2]$
40: prod_big.1 lt 0[3-4]$
```

**Figure 55. archiver(1M) -lv -c Command Output (Continued)**

```

41: prod_big.2 lt 0[5-6]$
42: endvsns

Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh

Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected

Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00      capacity: 1.2G space: 939.7M -il-o-----
mo.optic01      capacity: 1.2G space: 934.2M -il-o-----
mo.optic02      capacity: 1.2G space: 781.7M -il-o-----
mo.optic03      capacity: 1.2G space: 1.1G -il-o-----
mo.optic04      capacity: 1.2G space: 983.2M -il-o-----
mo.optic10      capacity: 1.2G space: 85.5M -il-o-----
mo.optic11      capacity: 1.2G space: 0 -il-o-----
mo.optic12      capacity: 1.2G space: 618.9k -il-o-----
mo.optic13      capacity: 1.2G space: 981.3M -il-o-----
mo.optic20      capacity: 1.2G space: 1.1G -il-o-----
mo.optic21      capacity: 1.2G space: 1.1G -il-o-----
mo.optic22      capacity: 1.2G space: 244.9k -il-o-----
mo.optic23      capacity: 1.2G space: 1.1G -il-o-----

Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE02      capacity: 9.5G space: 6.2G -il-o-----
lt.TAPE03      capacity: 9.5G space: 3.6G -il-o-----
lt.TAPE04      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE05      capacity: 9.5G space: 8.5G -il-o-----
lt.TAPE06      capacity: 9.5G space: 7.4G -il-o-----

Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
no_archive Noarchive path:users/bob
prod_big path:data minsize:50.2k
copy:1 arch_age:60 unarch_age:2592000
copy:2 arch_age:180
prod path:data
copy:1 arch_age:60
proj_1 path:projs/proj_1
copy:1 arch_age:60
copy:2 arch_age:60
joe path:.. uid:10006
copy:1 arch_age:60
copy:2 arch_age:60
pict path:.. gid:8005
copy:1 arch_age:60
copy:2 arch_age:60
no_archive Noarchive path:tmp

```

**Figure 55. archiver(1M) -lv -c Command Output (Continued)**

```
samfs path:.  
      copy:1 arch_age:240  
  
Archive sets:  
allsets  
  
joe.1  
  media: mo  
  Volumes:  
    optic01  
  Total space available: 934.2M  
  
joe.2  
  media: lt  
  Volumes:  
    TAPE01  
    TAPE02  
  Total space available: 14.7G  
  
pict.1  
  media: mo  
  Volumes:  
    optic02  
  Total space available: 781.7M  
  
pict.2  
  media: mo  
  Volumes:  
    optic03  
  Total space available: 1.1G  
  
prod.1  
  media: mo  
  Volumes:  
    optic20  
    optic21  
    optic22  
    optic23  
  Total space available: 3.3G  
  
prod_big.1  
  media: lt drives:2  
  Volumes:  
    TAPE03  
    TAPE04  
  Total space available: 12.1G  
  
prod_big.2  
  media: lt drives:2  
  Volumes:  
    TAPE05  
    TAPE06  
  Total space available: 16.0G  
  
proj_1.1
```

**Figure 55. archiver(1M) -lv -c Command Output (Continued)**

```

media: mo
Volumes:
  optic10
Total space available: 85.5M

proj_1.2
media: mo
Volumes:
  optic12
  optic13
Total space available: 981.9M

samfs.1
media: mo
Volumes:
  optic00
  optic01
Total space available: 1.8G

```

## Example 4

In this example, user files and project data files are archived to optical media. Note that [Figure 56](#) does not use the directory structure presented in [Table 54.](#)

Four VSN pools are defined; three pools are used for user, data, and project, and one is a scratch pool. When the `proj_pool` runs out of media, it relies on the `scratch_pool` to reserve volumes. This example shows how to reserve volumes for each archive set based on the set component, owner component, and file system component. Archiving is performed on a 10-minute interval, and an archiving log is kept.

[Figure 56](#) shows the `archiver.cmd` file and archiver output.

**Figure 56. archiver.cmd File and Archiver Output**

```

Reading archiver command file "example4.cmd"
1: # Example 4 archiver command file
2: # Using 4 VSN pools
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: fs = samfs
8: users users
9:   1 10m
10:
11: data data
12:   1 10m
13:
14: proj projects
15:   1 10m
16:
17: params
18: users.1 -reserve user

```

**Figure 56. archiver.cmd File and Archiver Output (Continued)**

```

Reading archiver command file "example4.cmd"
19: data.1 -reserve group
20: proj.1 -reserve dir -reserve fs
21: endparams
22:
23: vsnpools
24: users_pool mo optic0[1-3]$
25: data_pool mo optic1[0-1]$
26: proj_pool mo optic1[2-3]$
27: scratch_pool mo optic2.$
28: endvsnpools
29:
30: vsn
31: samfs.1 mo optic00
32: users.1 mo -pool users_pool -pool scratch_pool
33: data.1 mo -pool data_pool -pool scratch_pool
34: proj.1 mo -pool proj_pool -pool scratch_pool
35: endvsns

Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh

Archive media:
media:mo archmax: 4.8M Volume overflow not selected

Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00      capacity: 1.2G space: 939.7M -il-o-----
mo.optic01      capacity: 1.2G space: 934.2M -il-o-----
mo.optic02      capacity: 1.2G space: 781.7M -il-o-----
mo.optic03      capacity: 1.2G space: 1.1G -il-o-----
mo.optic04      capacity: 1.2G space: 983.2M -il-o-----
mo.optic10      capacity: 1.2G space: 85.5M -il-o-----
mo.optic11      capacity: 1.2G space: 0 -il-o-----
mo.optic12      capacity: 1.2G space: 618.9k -il-o-----
mo.optic13      capacity: 1.2G space: 981.3M -il-o-----
mo.optic20      capacity: 1.2G space: 1.1G -il-o-----
mo.optic21      capacity: 1.2G space: 1.1G -il-o-----
mo.optic22      capacity: 1.2G space: 244.9k -il-o-----
mo.optic23      capacity: 1.2G space: 1.1G -il-o-----

Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
  copy:1 arch_age:240
users path:users
  copy:1 arch_age:600
data path:data
  copy:1 arch_age:600
proj path:projects
  copy:1 arch_age:600
samfs path:.
  copy:1 arch_age:240

```



**Figure 56. archiver.cmd File and Archiver Output (Continued)**

```

Reading archiver command file "example4.cmd"
VSN pools:
data_pool media: mo Volumes:
    optic10
    Total space available: 85.5M

proj_pool media: mo Volumes:
    optic12
    optic13
    Total space available: 981.9M

scratch_pool media: mo Volumes:
    optic20
    optic21
    optic22
    optic23
    Total space available: 3.3G

users_pool media: mo Volumes:
    optic01
    optic02
    optic03
    Total space available: 2.7G

Archive sets:
allsets

data.1
    reserve:/group/
    media: mo
    Volumes:
        optic10
        optic20
        optic21
        optic22
        optic23
    Total space available: 3.4G

proj.1
    reserve:/dir/fs
    media: mo
    Volumes:
        optic12
        optic13
        optic20
        optic21
        optic22
        optic23
    Total space available: 4.2G

samfs.1
    media: mo
    Volumes:
        optic00
    Total space available: 939.7M

```

**Figure 56. archiver.cmd File and Archiver Output (Continued)**

```

Reading archiver command file "example4.cmd"

users.1
  reserve:/user/
media: mo
Volumes:
  optic01
  optic02
  optic03
  optic20
  optic21
  optic22
  optic23
Total space available: 6.0G

```

## ■ Archiver Guidelines

The archiver automates storage management operations using the `archiver.cmd` file. Before writing this file, it is useful to review some general guidelines that can improve the performance of your StorageTek ASM file system and the archiver. This ensures that your data is stored in the safest way possible.

Each site is unique in its application of computing, data storage hardware, and software. The following recommendations are based upon the experiences of StorageTek. When writing the `archiver.cmd` file for your site, be sure that you reflect the data storage requirements at your site by considering the following aspects.

1. Save your archive logs. The archive logs provide information that is essential to recovering data, even when the StorageTek ASM software is unavailable. It is recommended that you keep these logs in a safe place in the event of a catastrophic disaster during which the StorageTek ASM software is unavailable.
2. Use regular expressions for volumes. Let the system work for you by allowing it to put files on many different volumes. Volume ranges (specified using regular expressions) allow the system to run continuously. Using specific volume names for archive set copies can rapidly fill a volume, causing undue workflow problems as you remove a piece of media and replace it with another.
3. Base your archive interval on how often files are created and modified, and whether you want to save all modification copies. Remember, that the archive interval is the time between file system scans. A very short archive interval keeps the archiver scanning almost continuously.
4. Consider the number of file systems you are using. Multiple StorageTek ASM file systems generally increase the performance of the archiver as compared to a single StorageTek ASM file system. The archiver uses a

separate process for each file system. Multiple file systems can be scanned in considerably less time than a single file system.

5. Use directory structures to organize your files within the StorageTek ASM file system like UNIX file systems. For performance considerations, StorageTek recommends that you do not place more than 10,000 files in a directory.
6. Always make a minimum of two file copies on two separate volumes. Putting data on a single media type puts your data at risk if physical problems with the media occur. Do not rely on a single archive copy if at all possible.
7. Make sure you are dumping your metadata using `samfsdump(1M)` on a regular basis. The metadata (directory structure, file names, and so on) is stored in an archive set that has the same name as the file system. You can use this information to recover a file system in the event of a disaster. If you do not want to do this, you can prevent this data from being archived by assigning this archive set to a nonexistent VSN. For more information on preserving metadata, see the *StorageTek ASM, StorageTek QFS, and StorageTek SM QFS Disaster Recovery Guide QFS, StorageTek ASM, and StorageTek ASM QFS Disaster Recovery Guide* or the *StorageTek QFS and StorageTek ASM Software Installation and Configuration Guide*.

## ■ Troubleshooting the Archiver

Upon initial setup, the archiver might not perform the tasks as intended. Make sure that you are using the following tools to monitor the archiving activity of the system:

- `samu(1M)` utility's a display. This display shows archiver activity for each file system. It also displays archiver errors and warning messages, such as the following:

```
Errors in archiver commands - no archiving will be done
```

The `samu(1M)` utility's a display includes messages for each file system. It indicates when the archiver will scan the `.inodes` file again and the files currently being archived.

- Archive logs. You can define these logs in the `archiver.cmd` file, and you should monitor them regularly to ensure that files are archived to volumes. Archive logs can become excessively large and should be reduced regularly either manually or by using a `cron(1)` job. Archive these log files for safekeeping because the information enables data recovery.
- `sfind(1)`. Use this command to check periodically for unarchived files. If you have unarchived files, make sure you know why they are not being archived.

- `sls(1)`. Files are not considered for release unless a valid archive copy exists. The `sls -D` command displays inode information for a file, including copy information.

**Note:** Output from the `sls -D` command might show the word `archdone` on a file. This is not an indication that the file has an archive copy. It is only an indication that the file has been scanned by the archiver and that all the work associated with the archiver itself has been completed. An archive copy exists only when you can view the copy information displayed by the `sls(1)` command.

Occasionally, you might see messages to indicate that the archiver either has run out of space on cartridges or has no cartridges. These messages are as follows:

- When the archiver has no cartridges assigned to an archive set, it issues the following message:

```
No volumes available for Archive Set setname
```

- When the archiver has no space on the cartridges assigned to an archive set, it issues the following message:

```
No space available on Archive Set setname
```

## Why Files Are Not Archiving

The following checklist includes reasons why your StorageTek ASM environment might not be archiving files.

1. The `archiver.cmd` file has a syntax error. Run the `archiver -lv` command to identify the error, then correct the flagged lines.
2. The `archiver.cmd` file has a `wait` directive in it. Either remove the `wait` directive or override it by using the `samu(1M)` utility's `:arrun` command.
3. No volumes are available. You can view this from `archiver(1M) -lv` command output. Add more volumes as needed. You might have to export existing cartridges to free up slots in the automated library.
4. The volumes for an archive set are full. You can export cartridges and replace them with new cartridges (make sure that the new cartridges are labeled), or you can recycle the cartridges. For more information on recycling, see ["Recycling" on page 153](#).
5. The VSN section of the `archiver.cmd` file fails to list correct media. Check your regular expressions and VSN pools to ensure that they are correctly defined.

6. There is not enough space to archive any file on the available volumes. If you have larger files and it appears that the volumes are nearly full, the cartridges might be as full as the StorageTek ASM environment allows. If this is the case, add cartridges or recycle.

If you have specified the `-join path` parameter, and there is not enough space to archive all the files in the directory to any volume, no archiving occurs. You should add cartridges, recycle, or use one of the following parameters: `-sort path` or `-rsort path`. For more information on these parameters, see [“Associative Archiving: -join” on page 87](#).

7. The `archiver.cmd` file has the `no_archive` directive set for directories or file systems that contain large files.
8. The `archive(1) -n` (archive never) command has been used to set too many directories, and files are never archived.
9. Large files are busy. Thus, they never reach their archive age and are not archived.
10. Hardware or configuration problems exist with the automated library.
11. Network connection problems exist between client and server. Ensure that the client and the server have established communications.

## Additional Archiver Diagnostics

In addition to examining the items on the previous list, you should check the following when troubleshooting the archiver.

1. The syslog file (by default, `/var/adm/sam-log`). This file can contain archiver messages that can indicate the source of a problem.
2. Volume capacity. Ensure that all required volumes are available and have sufficient space on them for archiving.
3. If the archiver appears to cause excessive, unexplainable cartridge activity or appears to be doing nothing, turn on the trace facility and examine the trace file. For information on trace files, see the `defaults.conf(4)` man page.
4. You can use the `truss(1) -p pid` command on the archiver process (`sam-archiverd`) to determine the system call that is not responding. For more information on the `truss(1)` command, see the `truss(1)` man page.
5. The `showqueue(1M)` command displays the content of the archiver queue files. You can use this command to observe the state of archiver requests that are being scheduled or archived. Any archive request that cannot be scheduled generates a message that indicates the reason. This command also displays the progress of archiving.

## Why Files Are Not Releasing

The archiver and the releaser work together to balance the amount of data available on the disk cache. The main reason that files are not released automatically from disk cache is that they have not yet been archived.

For more information on why files are not being released, see [“Troubleshooting the Releaser” on page 137](#).

# Releasing

---

*Releasing* is the process by which the releaser makes disk cache space available by identifying archived files and releasing their disk cache copy. This makes room for other files to be created or staged from archive media. The releaser can release only archived files. Releasing the file results in a file without any data on the disk cache.

The StorageTek ASM file systems automatically invoke the releaser process when a site-specified disk threshold is reached. In contrast, you can use the `release(1)` command to release a file's disk space immediately or to set releasing parameters for a file. For more information about the releaser process, see the `sam-releaser(1M)` man page.

The releaser contains features that allow you to specify that files be released immediately after archiving, that files never be released, or that files be partially released. The partial release feature is particularly useful because some applications, such as `filemgr(1)`, read only the beginning of the file. With partial release, a portion of the file remains on the disk cache and the remainder of the file is released. Reading the first part of the file still on disk cache does not necessarily trigger the staging of the rest of the file back to disk cache from the archive media. These features, and many others, are described in this chapter.

This chapter contains the following topics:

- [“Releaser Overview” on page 122](#)
- [“Theory of Operation” on page 122](#)
- [“Definitions” on page 122](#)
- [“Partial Release and Partial Stage” on page 124](#)
- [“The `releaser.cmd` File” on page 128](#)
- [“The `archiver.cmd` File’s Role in Releasing” on page 135](#)
- [“Configuring the Releaser” on page 136](#)
- [“Running the Releaser Manually” on page 137](#)
- [“Troubleshooting the Releaser” on page 137](#)

## ■ Releaser Overview

When file system utilization exceeds its configured high watermark, the file system management software invokes the releaser. First, the releaser reads the `releaser.cmd` file and collects the directives that control the release process. Next, it scans the file system and collects information about each file. Finally, after scanning the entire file system, the releaser begins releasing files in priority order.

The releaser continues to release files as long as the file system remains above the configured low watermark. Typically, the releaser frees enough space to allow the file system to drop below the low water mark. If the releaser cannot find any files to release, it exits. The releaser runs later when more files can be released. While above the high watermark, the file system starts the releaser every one minute.

The high and low watermarks are set with the `high=percent` and `low=percent` file system mount options. For more information about these mount options, see the `mount_samfs(1M)` man page.

## ■ Theory of Operation

A file system can contain thousands of files. Keeping track of the release priority for all the files can be wasteful because releasing only several large files might return the file system to its low watermark. However, the releaser must examine the priority of each file or risk missing the best candidates for release. The releaser handles this condition by identifying only the first 10,000 candidates.

After identifying the first 10,000 candidates, the releaser discards subsequent candidates if they do not have a priority greater than the lowest-priority candidate among the first 10,000.

After the releaser has determined the priority of the first 10,000 candidates, it selects the files with the highest priority for release. After releasing each file, the releaser checks to see if the file system cache utilization is below the low watermark. If so, the releaser stops releasing files. If not, the releaser continues releasing the files in priority order.

If the releaser has released all 10,000 candidates and the file system is still above the low water mark, it starts over and identifies 10,000 new candidates.

The releaser exits if it cannot find any viable candidates. This can occur, for example, if files do not yet have archive copies. The StorageTek ASM file systems start the releaser again after one minute has elapsed.

## ■ Definitions

This section explains terms used throughout this chapter.



## Age

The *age* concept refers to the amount of elapsed time from a given event until now. A file's inode keeps track of the following times that are used by the releaser:

- Residence-change time
- Data-modified time
- Data-accessed time

You can view these times by using the `sls(1)` command with the `-D` option. Each time has a corresponding age. For example, if it is 10:15 a.m., a file with a modify time of 10:10 a.m. has a data-modified age of five minutes. For more information about the `sls(1)` command, see the `sls(1)` man page.

## Candidate

A *candidate* is a file that is eligible to be released. The reasons why a file would not be a candidate are as follows:

- The file is already offline.
- The file has not been archived.
- The `archiver.cmd` command file specifies the `-norelease` attribute for the file, and the required copies have not yet been made.
- The file is marked as damaged.
- The file is not a regular file. It is a directory, block, character-special file, or pipe.
- The archiver is staging the file to make an additional copy. The file becomes eligible for release after the archiver stages it.
- The age of the file is negative. This usually occurs for NFS clients with inaccurate clock settings.
- The file is marked to never be released. You can use the `release(1) -n` command to specify this.
- The file was staged at a time in the past that is less than the minimum residence time setting. For more information, see [“Specifying a Minimum Residence Time: `min\_residence\_age`” on page 133.](#)
- The file was flagged for partial release, by using the `release(1)` command's `-p` option, and is already partially released.
- The file is too small.

## Priority

A *priority* is a numeric value that indicates the rank of a candidate file based on user-supplied weights that are applied to numeric attributes of that candidate. The overall priority is the sum of two types of priority: age priority and size priority.

Candidate files with numerically larger priorities are released before candidates with numerically smaller priorities.

## Weight

The *weight* is a numeric value that biases the priority calculation to include file attributes in which you are interested and to exclude file attributes in which you are not interested. For example, the size attribute of a file is excluded from the priority calculation if the size weight is set to zero. Weights are floating-point values from 0.0 to 1.0.

## Partial release

A file can be *partially released* by specifying that a beginning portion of the file remain in disk cache while the rest of the file is released. For example, partial release is valuable when using utilities like `filemgr(1)` that read the beginning of a file.

## ■ Partial Release and Partial Stage

Releasing and staging are complementary processes. Files can be completely released from online disk cache after they are archived, or a site can specify that the beginning of a file (the *stub*) remain in disk cache while the remainder of the file is released. This ability to partially release a file provides immediate access to data in the file stub without staging the file.

A system administrator can specify both the default partial release size and the maximum size of the stub to remain online when a file system is mounted. The system administrator can set these on the `mount(1M)` command, as follows:

- Specify the `-o partial=n` option to set the default size (*n*) of a file stub to remain online. The `-o partial=n` setting must be less than or equal to the `-o maxpartial=n` setting. The smallest possible setting is `-o partial=8` kilobytes. The default setting is `-o partial=16` kilobytes.
- Specify the `-o maxpartial=n` option to set the maximum size (*n*) of a file stub to remain online. To limit the size of the file stub that can be left online, use the `-o maxpartial=n` option and specify a size equal to the largest stub that can be left online. To disable the partial release feature, specify `-o maxpartial=0`.

An user can specify the default stub size for a file by specifying the `-p` option on the `release(1)` command or the `p` option on the `sam_release(3)` library routine. To specify different-sized file stubs for different types of files or different applications, a user can specify the `-s` option on the `release(1)` command or the `s` option on the `sam_release(3)` library routine. The `-s` and `s` values must be less than the `-o maxpartial` value used on the `mount(1M)` command when the file system was mounted.

Another mount option, `-o partial_stage=n`, allows a system administrator to establish how much of a partial release stub must be read before the rest of the file is staged. That is, reading past the `-o partial_stage=n` size specification initiates the stage of the file.

By default, the `-o partial_stage=n` option is set to the size of the partial release stub. This value can be configured, though, and it affects file staging as follows:

- If the `-o partial_stage=n` option is set to the size of the partial release stub, the default behavior prevents the file from being staged until the application reaches the end of the partial release stub. Waiting until the end of the stub is reached causes a delay in accessing the rest of the file.
- If the `-o partial_stage=n` option is set to a value smaller than the partial release stub, the following occurs. After the application crosses the threshold set by the `-o partial_stage=n` option, the rest of the file is staged. This reduces the chance of a delay in accessing the rest of the file data.

**Example.** Assume that the following options are in effect:

- `-o partial_stage=16` (which is 16 kilobytes)
- `-o partial=2097152` (which is 2 gigabytes)
- `-o maxpartial=2097152` (which is 2 gigabytes)

The `filemgr(1)` program is being used, and it reads the first 8 kilobytes of a file. The file is not staged. A video-on-demand program reads the same file, and the file is staged after it reads past the first 16 kilobytes of the file. The application continues reading the 2 gigabytes of disk data while the archive tape is mounted and positioned. When the video-on-demand program reads past 2 gigabytes of file data, the application reads immediately behind the staging activity. The application does not wait because the tape mounting and positioning is done while the application reads the partial file data.

Several command line options affect whether a file can be marked for partial release. Some options are enabled by the system administrator, and others can be enabled by individual users. The following sections describe the release characteristics that can be set by the various types of users.

## System Administrator Option Summary

The system administrator can change the maximum value and default value for partial release when the file system is mounted. The `mount(1M)` options in [Table 55](#) affect partial release. For more information about the `mount(1)` command, see the `mount_samfs(1M)` man page.

**Table 55. Mount Options for Partial Release**

<b>mount(1M) Option</b>	<b>Effect</b>
<code>-o maxpartial=<i>n</i></code>	<p>Determines the maximum amount of space, in kilobytes, that can remain in online disk cache if a file is marked for partial release. The maximum value is 2,097,152 kilobytes, which is 2 gigabytes. The minimum value is 0, which prevents any file from being partially released.</p> <p>If <code>-o maxpartial=0</code> is specified, the partial release feature is disabled, released files are released completely, and no portion of a file remains in disk cache. Users cannot override the value specified on this option after the file system is mounted.</p> <p>By default, the <i>n</i> argument is set to 16. This setting enables users to mark files for partial release with the maximum amount remaining on disk being 16 kilobytes.</p>

**Table 55. Mount Options for Partial Release (Continued)**

<code>-o partial=<i>n</i></code>	Sets a default amount of space, in kilobytes, to remain in disk cache if a user marks a file for partial release by using the <code>release(1)</code> command's <code>-p</code> option. The <i>n</i> argument must be at least 8, but it can be as great as the value specified for the <code>-o maxpartial=<i>n</i></code> option. Because some applications do not need access to the entire file to complete their work, this option can be used to ensure that applications have the needed beginnings of files available to them. At the same time, using this option prevents files from being staged unnecessarily. The default value is <code>-o partial=16</code> .
<code>-o partial_stage=<i>n</i></code>	Specifies that when a partially released file is accessed, <i>n</i> bytes of the file must be read before the entire file is staged from the archive media. This value is typically set to be lower than the amount of the <code>-o partial</code> setting. For <i>n</i> , specify an integer value from 0 to the <code>-o maxpartial</code> specification. By default, this is set to 16, or whatever value was specified for the <code>-o partial</code> option.
<code>-o stage_n_window=<i>n</i></code>	Specifies the amount of data to be staged at any one time to <i>n</i> . For <i>n</i> , specify an integer from 64 to 2,048,000. The default is 256 kilobytes. This option applies only to files that have the <code>stage -n</code> attribute set.

---

## User Option Summary

The system administrator sets maximum and default values for the size of a file stub that can remain in disk cache after the file is released. The system administrator also determines whether or not the partial release feature is enabled for a particular file system.

By using the `release(1)` command and the `sam_release(3)` library routines, however, a user can set other release attributes and can specify the files to be marked for partial release. The command and library options that determine partial release attributes are shown in [Table 56](#). For more information about the `release(1)` command, see the `release(1)` man page.

For more information about the `sam_release(3)` library routine, see the `sam_release(3)` man page.

**Table 56. User Release Options**

Options	Effect
release(1) command and <code>-p</code> option or sam_release(3) library routine and <code>p</code> option	The <code>-p</code> and <code>p</code> options mark the named file for partial release. If these options are used, the amount of the file remaining in online disk cache after the file is released depends on the value of the <code>-o partial=<i>n</i></code> option that was set when the file system in which the file resides was mounted. These options cannot be used to specify the number of bytes to remain online.
release(1) command and <code>-s partial_size</code> option or sam_release(3) library routine and <code>s</code> option	The <code>-s</code> and <code>s</code> options mark the named file for partial release, and they specify the amount of the file to remain in online disk cache. The arguments to the <code>-s</code> or <code>s</code> options specify the amount, in kilobytes, to remain online. A user cannot specify that the amount of a file remaining online be greater than the amount specified for the <code>-o maxpartial=<i>n</i></code> value when the file system was mounted. If the user's value is greater than the value for the file system, the value for the file system is used, and the user's specification is ignored.

## ■ The `releaser.cmd` File

The `/etc/opt/SUNWsamfs/releaser.cmd` file consists of directive lines that specify site-specific releasing actions. The `releaser.cmd` file can contain directives for setting the release priority, specifying a log file, and other actions.

The following sections describe the `releaser.cmd` directives:

- “Specifying Age-Related and Size-Related Release Priority Directives: `weight_age`, `weight_age_access`, `weight_age_modification`, and `weight_age_residence`” on page 129
- “Specifying Directives for Individual File Systems: `fs`” on page 132
- “Specifying Debugging Directives: `no_release` and `display_all_candidates`” on page 132
- “Specifying a Minimum Residence Time: `min_residence_age`” on page 133
- “Specifying a Log File: `logfile`” on page 133
- “Inhibiting Releasing for Rearchived Files: `rearch_no_release`” on page 134
- “Adjusting the size of the Releaser Candidate List: `list_size`” on page 135

For more information about these directives, see the `releaser.cmd(4)` man page.

## Specifying Age-Related and Size-Related Release Priority

**Directives:** `weight_age`, `weight_age_access`, `weight_age_modification`, and `weight_age_residence`

Files are released from a file system using a priority order determined by directives defined in the `releaser.cmd` file. Both file age and file size are considered. By default, sites release the largest, oldest files first, leaving the smallest, newest files on disk. The following sections show how the releaser considers a file’s age and size when determining the release priority of files in a file system.

For additional information about releaser directives, see the `releaser.cmd(4)` man page.

### File Age

The releaser considers the following possible ages when determining the age-related component of a file’s release priority:

- The age since it was last accessed
- The age since it was last modified
- The age since it changed residency in disk cache

In some cases, you might want the access age of a file to take precedence over the modification age. In other cases, a simple age derived from the most

recently accessed time, modified time, and residence-changed time is preferred.

By default, the age of a file is the more recent of the file's three ages:

- File access age
- File modification age
- File residency age

You can use directives to specify that a weighted age priority be used when calculating the release priority for a file.

Figure 57 shows the age priority directives' formats.

**Figure 57. Age Priority Directive Formats**

```
weight_age = float
weight_age_access = float
weight_age_modification = float
weight_age_residence = float
```

- The `weight_age` directive specifies that a file's default age (the smaller of the file's access, modification, or residence age) be given a weighting factor. For *float*, specify a floating-point number in the following range: 0.0 *float* 1.0. By default, *float* = 1.0.
- This directive cannot be specified in conjunction with the `weight_age_residence`, `weight_age_modify`, or `weight_age_access` directives.
- The `weight_age_residence`, `weight_age_modify`, and `weight_age_access` directives specify that that a file's age be determined based on a combination of one, two, or three of these possible ages. For *float*, specify a floating-point number in the following range: 0.0 *float* 1.0. By default, *float* = 1.0.

These directives cannot be specified in conjunction with the `weight_age` directive.

If the `weight_age_residence`, `weight_age_modify`, and `weight_age_access` directives are used, the age-related priority for a file is calculated based on a combination of all three ages. First, file age data is gathered for each file's possible age. Secondly, the file age data is multiplied by the weighting factors specified in the `releaser.cmd` file. Finally, the file's age-related priority is calculated by summing the product



of the age data multiplied by each weighting factor, as shown in the equation in [Figure 58](#).

**Figure 58. Priority Calculation**

```

file access age * weight_age_access
+ file modification age * weight_age_modification
+ file residency age * weight_age_residence
-----
= age_related_priority

```

**Example.** [Figure 59](#) shows lines in a `releaser.cmd` file that specify that only the file's residence age be considered (and that the modification age and the access age be ignored) when calculating the release priority of a file.

**Figure 59. `releaser.cmd` File Fragment**

```

weight_age_residence = 1.0
weight_age_modify = 0.0
weight_age_access = 0.0

```

After a file's age-related priority is calculated, it is multiplied by the file's size-related priority. The size-related priority is calculated as shown in the following section.

## File Size

The releaser considers a file's size when determining the size-related component of a file's release priority. The size of the file (in 4-kilobyte blocks) is multiplied by the weight specified for the `weight_size` directive to obtain the size-related component of a file's release priority.

The format of the `weight_size` directive is as follows:

```
weight_size = float
```

For *float*, specify a floating-point number in the following range: 0.0 *float* 1.0. By default, *float* = 1.0.

**Example.** [Figure 60](#) shows a `releaser.cmd` file that specifies that when calculating a file's release priority, a file's size is to be ignored for all files in the `samfs1` and `samfs2` file system.

**Figure 60. `releaser.cmd` File**

```

# releaser.cmd file
logfile = /var/adm/default.releaser.log
weight_size = 0.0
#
fs = samfs1
weight_age = 1.0
logfile = /var/adm/samfs1.releaser.log

```

**Figure 60. releaser.cmd File (Continued)**

```
#
fs = samfs2
weight_age_modify = 0.3
weight_age_access = 0.03
weight_age_residence = 1.0
logfile = /var/adm/samfs2.releaser.log
```

## Specifying Directives for Individual File Systems: `fs`

You can use the `fs = family_set_name` directive in the `releaser.cmd` file to indicate that the directives that follow the `fs =` directive apply only to the named file system. This directive has the following format:

```
fs = family_set_name
```

For *family\_set\_name*, specify the name of a Family Set in the `mcf` file.

Directives preceding the first `fs =` directive are global and apply to all file systems. Directives following the `fs =` directive override global directives. The directives described in this chapter can be used as either global directive or as directives specific to one file system.

The `releaser.cmd(4)` man page includes examples of the `fs =` directive.

## Specifying Debugging Directives: `no_release` and `display_all_candidates`

The `no_release` and `display_all_candidates` directives can be useful when tuning or debugging the releaser. These directives are as follows:

- The `no_release` directive prevents files from being removed from online disk cache. You can use this directive to check the directives in the `releaser.cmd` without actually releasing files. This directive has the following format:

```
no_release
```

- The `display_all_candidates` directive writes the names of all release candidates to the log file. This directive has the following format:

```
display_all_candidates
```

These directives are helpful when debugging because the releaser writes the names of release candidates to the log file, but it does not physically release them from the file system.

## Specifying a Minimum Residence Time:

### min\_residence\_age

The `min_residence_age` directive enables you to specify the minimum amount of time that a file must reside in a file system before it becomes a candidate for release. This directive has the following format:

```
min_residence_age = time
```

For *time*, specify a time in seconds. The default time is 600, which is 10 minutes. There is no practical minimum or maximum *time* setting.

## Specifying a Log File: logfile

If a `logfile` directive is specified in the `releaser.cmd` file, the releaser either appends its activity log to the indicated file name, or the releaser creates the file name if it does not exist. This directive has the following format:

```
logfile = filename
```

For *filename*, specify the name of a log file.

[Figure 61](#) shows a sample log file (note that some lines have been wrapped to fit on the page).

**Figure 61. Releaser Log File Example**

```
Releaser begins at Wed Apr 28 17:29:06 1999
inode pathname      /sam1/.inodes
low-water mark      24%
weight_size         1
weight_age          1
fs equipment ordinal 1
family-set name     samfs1
started by sam-amld? yes
release files?      yes
display_all_candidates? no
---before scan---
blocks_now_free:    3481504
lwm_blocks:         3729362
---scanning---
10501 (R: Wed Apr 21 18:47:50 CDT 1999) 10001 min, 500 blks /sam1/testdir0/filevp
10500 (R: Wed Apr 21 18:48:10 CDT 1999) 10000 min, 500 blks /sam1/testdir0/filewq
...
---after scan---
blocks_now_free:    3730736
lwm_blocks:         3729362
archnodrop: 0
already_offline: 0
bad_inode_number: 0
```

**Figure 61. Releaser Log File Example (Continued)**

```

damaged: 0
extension_inode: 0
negative_age: 0
nodrop: 1
not_regular: 9
number_in_list: 675
released_files: 202
too_new_residence_time: 0
too_small: 2
total_candidates: 675
total_inodes: 1376
wrong_inode_number: 0
zero_arch_status: 689
zero_inode_number: 0
zero_mode: 0
CPU time: 2 seconds.
Elapsed time: 10 seconds.
Releaser ends at Wed Apr 28 17:29:16 1999

```

The `releaser(1M)` man page describes the information contained in the log file. Because the size of the log increases with each releaser run, be sure to allow for decreasing the size of the log, or omit the `logfile` keyword.

[Figure 62](#) shows the mathematical relationships that exist among the statistics shown under the `---after scan---` line:

**Figure 62. Mathematical Relationships From the `---after scan---` Line in [Figure 61](#)**

```

total_inodes = wrong_inode_number +
zero_inode_number +
zero_mode +
not_regular +
extension_inode +
zero_arch_status +
already_offline +
damaged +
nodrop +
archnodrop +
too_new_residence_time +
too_small +
negative_age +
total_candidates
released_files = total_candidates

```

## Inhibiting Releasing for Rearchived Files:

`rearch_no_release`

By default, files marked for rearchiving are released. If the `rearch_no_release` directive is specified in the `releaser.cmd(4)` file, the

releaser does not release the files marked for rearchiving. This directive has the following format:

```
rearch_no_release
```

## Adjusting the size of the Releaser Candidate List:

`list_size`

You can use the `list_size` directive to specify the number of releaser candidates. If you notice that the releaser makes multiple file system scans before it releases the number of files needed to get to the low water mark, you might want to consider raising this value to a level greater than the default of 10,000. This might be true in a file system that contains many small files. You can get information about releaser activities from the releaser log file. This directive has the following format:

```
list_size = number
```

For number, specify an integer such that 10 *number* 2,147,483,648.

## ■ The archiver.cmd File's Role in Releasing

Most directives in the `archiver.cmd` file affect archiving, but the archive set assignment directive allows you to specify release attributes that apply to all files in an archive set.

The archive set assignment directive has the following format:

```
archive_set_name path [search_criteria ...] directives ...
```

Table 57 shows the *directives* that pertain to releasing.

**Table 57. Archive Set Assignment *directives***

Directive	Effect
-release a	Specifies that the files in the archive set should be released after the first archive copy is made. Do not use this option if you are making more than one archive copy of each file. In such a situation, copy 1 would be staged in order to make copy 2.
-release n	Specifies that the files in the archive set should never be released.
-release p	Specifies that the files in the archive set should be partially released after archiving.

For more information about these and the other `archiver.cmd` directives, see [“Archiving” on page 47](#).

## ■ Configuring the Releaser

It is necessary to decide the characteristics of files in cache for your site. It is wasteful to load a tape if you are staging only a few kilobytes, so you may want to bias your system to retain small files in cache. [Figure 63](#) shows the directives to use in the `releaser.cmd` file to release the largest files first.

**Figure 63. Directives to Release the Largest Files First**

```
weight_size = 1.0
weight_age = 0.0
```

Alternately, you may want to retain recently modified files in cache since a recently modified file might be modified again soon. This avoids the overhead created when the file is staged to enable modification. In this case, use the second set of age weights. [Figure 64](#) shows the directives to use in the `releaser.cmd` file to weight files in strict order starting with the oldest modified to the most recently modified.

**Figure 64. Directives to Release Oldest-Modified Files First**

```
weight_size = 0.0
weight_age_access = 0.0
weight_age_modify = 1.0
weight_age_residence = 0.0
```

However, as the following examples demonstrate, most situations are not this straightforward.

**Example 1.** Assume that you want to release the largest files first. There are hundreds of small files that are the same size, and there are several large files. The cumulative size of the small files might exceed the size of the single, largest file. Eventually, the releaser releases all the large files. If `weight_age = 0.0` is specified, the releaser releases the small files in essentially random order because they are all the same size and have the same release priority.

In this scenario, you could set `weight_age = 0.01` as a tiebreaker. The releaser would release the older of two equally sized files first.

**Example 2.** This example presents a better method to specify how to release the largest files first.

Set `weight_size = 1.0` and `weight_age = 0.01`.

These directives violate the largest-first policy by counting smaller, less recently accessed files as better candidates than larger, more recently accessed files. You can make this effect as small as you want by making `weight_age` smaller than `weight_size`. For example, based on the

previous settings, a 4-kilobyte file that staged 100 minutes ago and an 8-kilobyte file that just staged both have the same release priority.

The releaser randomly chooses a file to release. If it chooses a 4-kilobyte file, it violates the largest-first intent. Setting `weight_age` considerably smaller (for example, to 0.001) reduces this effect. If a 4-kilobyte file staged 1,000 minutes ago, it has the same priority as the 8-kilobyte file that just staged.

You can use the `no_release` and `display_all_candidates` directives and run the releaser manually to obtain a list of candidates in priority order for use in adjusting the priority weights.

## ■ Running the Releaser Manually

From time to time, you might want to run the releaser manually. For this, you need to know the mount point of the file system and the low watermark the releaser should attempt to reach.

For example, to release files in the `/sam1` file system until it reaches 47 percent full, log in as root and type the following:

```
# /opt/SUNWsamfs/sbin/sam-releaser /sam1 47 1.0
```

The final argument, `weight-size`, is overridden by a `weight_size` command in the `releaser.cmd` file. As the releaser runs, it writes information to your screen and to the releaser log file (if specified in the `releaser.cmd` file.) For more information, see the `sam-releaser(1M)` man page.

## ■ Troubleshooting the Releaser

There can be several reasons for the releaser to not release a file. Some possible reasons are as follows:

- Files can be released only after they are archived. There might not be an archive copy. For more information about this, see [“Why Files Are Not Archiving” on page 118](#).
- The archiver requested that a file not be released. This can occur under the following conditions:
  - The archiver has just staged an offline file to make an additional copy.
  - The `-norelease` directive in the `archiver.cmd` file was set and all the copies flagged `-norelease` have not been archived. Note that the releaser summary output displays the total number of files with the `archnodrop` flag set.

- The file is set for partial release, and the file size is less than or equal to the partial size rounded up to the disk allocation unit (DAU) size (block size).
- The file changed residence in the last *min\_residence\_age* minutes.
- The `release -n` command has been used to prevent directories and files from being released.
- The `archiver.cmd` file has the `-release n` option set for too many directories and files.
- The releaser high watermark is set too high, and automatic releasing occurs too late. Verify this in the `samu(1M)` utility's `m` display or with ASM QFS Manager, and lower this value.
- The releaser low watermark is set too high, and automatic releasing stops too soon. Check this in the `samu(1M)` utility's `m` display, or with ASM QFS Manager, and lower this value.
- Large files are busy. They will never reach their archive age, never be archived, and never be released.



# Staging

*Staging* is the process of copying file data from nearline or offline storage back to online storage. The staging capabilities enable you to stage files immediately, to never stage files, to specify partial staging, and to specify other staging actions. The never-stage capability can be used, for example, by applications that randomly access small records from large files; when this is enabled, the data is accessed directly from the archive media without staging the file online.

This chapter describes the StorageTek ASM file staging capability. It contains the following topics:

- [“The `stager.cmd` File” on page 139](#)
- [“The `archiver.cmd` File’s Role in Staging” on page 145](#)
- [“Prioritizing Preview Requests Using the `preview.cmd` File” on page 146](#)
- [“Calculating Total Preview Request Priority” on page 149](#)
- [“How to Set Up a Preview Request Priority Scheme” on page 149](#)

## ■ The `stager.cmd` File

You can use the `stager.cmd` file to specify the stager’s behavior. The full path name to this file is `/etc/opt/SUNWsamfs/stager.cmd`. By default, the stager performs the following actions:

- The stager attempts to use all the drives in the library to stage files.
- The stage buffer size is determined by the media type, and the stage buffer is not locked.
- No log file is written.
- Up to 1000 stage requests can be active at any one time.

The `stager.cmd` file allows you to specify directives to override the default behaviors. The rest of this section describes the stager directives. For additional information on stager directives, see the `stager.cmd(4)` man page.

The [“Example `stager.cmd` File” on page 145](#) shows the completed `stager.cmd` file after all possible directives have been set.

Figure 65 shows the example `mcf` file used by the examples in this chapter.

**Figure 65. `mcf` File Used in this Chapter's Examples**

```
#
# Sun StorEdge SAM-FS file system configuration example
#
# Equipment      Eq Eq Family Dev Additional
# Identifier     Or Tp Set   St  Parameters
# -----
samfs1          60 ms samfs1
/dev/dsk/c1t1d0s6 61 md samfs1 on
/dev/dsk/c2t1d0s6 62 md samfs1 on
/dev/dsk/c3t1d0s6 63 md samfs1 on
/dev/dsk/c4t1d0s6 64 md samfs1 on
/dev/dsk/c5t1d0s6 65 md samfs1 on
#
samfs2          2 ms samfs2
/dev/dsk/c1t1d0s0 15 md samfs2 on
/dev/dsk/c1t0d0s1 16 md samfs2 on
#
/dev/samst/c0t2d0 20 od -      on
/dev/samst/c1t2u0 30 rb dog   on /var/opt/SUNWsamfs/catalog/dogcat
/dev/samst/c1t5u0 31 od dog   on
/dev/samst/c1t6u0 32 od dog   on
/dev/rmt/0cbn    40 od -      on
/dev/samst/c1t3u1 50 rb bird on /var/opt/SUNWsamfs/catalog/birdcat
/dev/rmt/2cbn   51 tp bird  on
```

## To Create or Modify a `stager.cmd` File and Propagate Your Changes

1. Use `vi(1)` or another editor to edit the `stager.cmd` file.

The full path to this file is as follows:

```
/etc/opt/SUNWsamfs/stager.cmd
```

For information on the directives you can include in this file, see the following subsections:

- [“Specifying the Number of Drives” on page 141](#)
- [“Setting the Stage Buffer Size” on page 141](#)
- [“Specifying a Log File” on page 142](#)
- [“Specifying the Number of Stage Requests” on page 144](#)

2. Save and close the `stager.cmd` file.

- Use the `samd(1M)` command with its `config` option to propagate the file changes and restart the system.

```
# samd config
```

## Specifying the Number of Drives

By default, the stager uses all available drives when staging files. If the stager keeps all the drives busy, this can interfere with the archiver's activities. The `drives` directive specifies the number of drives available to the stager. This directive has the following format:

```
drives = library count
```

**Table 58. Arguments for the `drives` Directive**

Argument	Meaning
<i>library</i>	The Family Set name of a library as it appears in the StorageTek ASM mcf file.
<i>count</i>	The maximum number of drives to be used. By default, this is the number of drives configured in the mcf file for this library.

For example, the following directive line specifies that only one drive from the `dog` family set's library be used for staging files:

```
drives = dog 1
```

For more information on the `mcf` file, see the `mcf(4)` man page.

## Setting the Stage Buffer Size

By default, a file being staged is read into memory in a buffer prior to restoring the file from the archive media back to online disk cache. You can use the `bufsize` directive to specify a nondefault buffer size and, optionally, to lock the buffer. These actions can improve performance, and you can experiment with various `buffer_size` values. This directive has the following format:

```
bufsize = media buffer_size [ lock ]
```

**Table 59. Arguments for the *bufsize* Directive**

Argument	Meaning
<i>media</i>	Specify the archive media type from the list on the <i>mcf(4)</i> man page.
<i>buffer_size</i>	Specify a number from 2 through 32. The default is 4. This value is multiplied by the <i>dev_blksize</i> value for the media type, and the resulting buffer size is used. The <i>dev_blksize</i> can be specified in the defaults.conf file. The higher the number specified for <i>buffer_size</i> , the more memory is used. For more information on this file, see the defaults.conf(4) man page.
lock	<p>The lock argument indicates whether or not the archiver should use locked buffers when staging archive copies. If lock is specified, the stager sets file locks on the stage buffer in memory for the duration of the copy operation. This avoids the overhead of locking and unlocking the buffer for each I/O request and can result in a reduction in system CPU time.</p> <p>The lock argument should be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition.</p> <p>The lock argument is effective only if direct I/O is enabled for the file being staged. By default, lock is not specified and the file system sets the locks on all direct I/O buffers, including those for staging. For more information on enabling direct I/O, see the <i>setfa(1)</i> man page, the <i>sam_setfa(3)</i> library routine man page, or the <i>-O forcedirectio</i> option on the <i>mount_samfs(1M)</i> man page.</p>

For example, this directive can be specified in the *stager.cmd* file in a line such as the following:

```
bufsize=od 8 lock
```

## Specifying a Log File

You can request that the StorageTek ASM file system collect file-staging event information and write it to a log file. The *logfile* directive specifies a log file to which the stager can write logging information. This directive has the following format:

```
logfile=filename [ event ]
```

For *filename*, specify a full path name.

For *event*, specify one or more staging events. If you specify more than one *event*, use spaces to separate each *event*. Events enabled by default are as follows: finish cancel error. Possible *event* specifications are as follows:

**Table 60. Keywords for the *event* Argument**

<i>event</i>	Action
all	Logs all staging events.
start	Logs when staging begins for a file.
finish	Logs when staging ends for a file. Enabled by default.
cancel	Logs when the operator cancels a stage. Enabled by default.
error	Logs staging errors. Enabled by default.

When a log file is specified, the stager writes one or more lines to the log file for each file staged. This line includes information such as the name of the file, the date and time of the stage, and the VSN. For example, the following directive line specifies file `/var/adm/stage.log`:

```
logfile=/var/adm/stage.log
```

Figure 66 shows an example of a stager log file.

**Figure 66. Stager Log File Example**

```
S 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1
root other root 0
F 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1
root other root 0
S 2003/12/16 14:06:27 dk disk02 4.a68 1218.1387 519464 /sam1/testdir1/fileaq 1
root other root 0
S 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1
root other root 0
F 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1
root other root 0
S 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /sam1/testdir0/filedb
1 root other root 0
F 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /sam1/testdir0/filedb
1 root other root 0
S 2003/12/16 14:06:59 dk disk02 f.f82 3501.115 1458848 /sam1/testdir1/filecb 1
root other root 0
S 2003/12/16 14:07:15 dk disk01 1f.473 1368.1419 636473 /sam1/testdir0/fileed 1
root other root 0
S 2003/12/16 14:07:15 dk disk02 16.f15 3362.45 1065457 /sam1/testdir1/filecz 1
root other root 0
```

**Figure 66. Stager Log File Example (Continued)**

```
S 2003/12/16 14:07:31 dk disk01 23.201d 3005.1381 556807 /sam1/testdir0/fileeq
1 root other root 0
S 2003/12/16 14:07:47 dk disk01 26.c4d 2831.1113 1428718 /sam1/testdir0/fileez
1 root other root 0
S 2003/12/16 14:07:47 dk disk02 1b.835 3736.59 1787855 /sam1/testdir1/filedp 1
root other root 0
```

As [Figure 66](#) shows, the stager log file consists of lines of information divided into nine fields. [Table 61.](#) describes the content of the stager log file fields.

**Table 61. Stager Log File Fields**

Field	Content Description
1	Stage activity. S for start. C for canceled. E for error. F for finished.
2	Date of stage action in <i>yyyy/mm/dd</i> format.
3	Time of stage action in <i>hh:mm:ss</i> format.
4	Archive media type. For information on media types, see the <i>mcf(4)</i> man page.
5	VSN.
6	Physical position of start of archive file on media ( <i>tar(1)</i> file) and file offset on the archive file in hexadecimal.
7	Inode number and generation number. The generation number is an additional number used in addition of the inode number for uniqueness since inode numbers get re-used.
8	Length of file.
9	Name of file.
10	Archive copy number.
11	The file's user ID.
12	The file's group ID.
13	The requestor's group ID.
14	The Equipment Ordinal of the drive from which the file was staged.

## Specifying the Number of Stage Requests

You can specify the number of stage requests that can be active at any one time by using the `maxactive` directive. This directive has the following format:

```
maxactive=number
```

By default, *number* is 4000. The minimum number allowed is 1.

For example, the following directive line specifies that no more than 500 stage requests can be in the queue simultaneously:

```
maxactive=500
```

## Example stager.cmd File

Figure 67 shows an example `stager.cmd` file.

**Figure 67. Example `stager.cmd` File**

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

## ■ The archiver.cmd File's Role in Staging

Most directives in the `archiver.cmd` file affect archiving, but the archive set assignment directive allows you to specify stage attributes that apply to all files in an archive set. The archive set assignment directive has the following format:

```
archive_set_name path [search_criteria ...] directives ... ]
```

The chapter called “[Archiving](#)” on page 47 describes the archive set assignment directive and its arguments completely. [Table 62](#) shows the staging directives that can appear as *directives* in an archive set assignment directive.

**Table 62. Staging *directives* that can Appear in the `archiver.cmd` File**

Directive	Effect
-stage a	Specifies that the files in the archive set should be associatively staged.
-stage n	Specifies that the files in the archive set should never be staged.

For more information on these and the other `archiver.cmd` directives, see “[Archiving](#)” on page 47.

## ■ Prioritizing Preview Requests Using the `preview.cmd` File

The archiver and stager processes both can request that media be loaded and unloaded. If the number of requests exceeds the number of drives available for media loads, the excess number of requests is sent to the preview queue.

Archive and stage requests in the preview queue are those that cannot be immediately satisfied. By default, preview requests are satisfied in first-in-first-out (FIFO) order.

The number of entries that can be in the preview queue is determined by the `previews=` directive in the `defaults.conf` file. For information on changing the value of this directive, see the `defaults.conf(4)` man page.

You can assign different priorities to preview requests. You can override the FIFO default by entering directives in the preview command file, which is written to the following location:

```
/etc/opt/SUNWsamfs/preview.cmd
```

This file schedules preview requests based on whether the request is for file staging or archiving. You can also increase the priority for specific VSNs. Further, settings in the `preview.cmd` file can also reprioritize preview requests for all or for specific file systems based on the high watermark (HWM) or low watermark (LWM) settings.

The `sam-amld` daemon reads the preview directives at startup. You must specify the directives one per line. If you change this file while the `sam-amld` daemon is running, you have to restart the `sam-amld` daemon to have them take effect. Comment lines begin with a pound sign (`#`) and extend through the end of the line. For more information on this file, see the `preview.cmd(4)` man page.

The following two types of directives can appear in the `preview.cmd` file:

- Global directives, which apply to all file systems. These must appear before the first `fs =` line.
- Directive that are specific to a file system, which follow the global directives. Like the `archiver.cmd` file, the `preview.cmd` file can contain directives specific to individual file systems. The directives specific to individual file systems must appear in the file after all global directives.

The file system directives must begin with an `fs = file_system_name` directive. This directive names the file system to which all subsequent directives pertain. More than one block of file directives can appear in a file. File system directives apply until the next `fs =` line is encountered or until the end of file is encountered.



**Note:** When multiple directives affect a file system, the directives that are specific to a particular file system override the global directives.

## VSN and Age Directives (Global)

The VSN and age priority directives are global directives. If they are present in your `preview.cmd` file, they must appear before any directives that are specific to a file system.

That is, they must appear prior to any `fs =` directives. The VSN priority directive has the following format:

```
vsn_priority = value
```

This directive is a static priority factor. It indicates the value by which the total priority increases for a VSN flagged as a high-priority VSN. The default value for `vsn_priority` is 1000.0. VSNs must have their priority flag set when they are scheduled as preview requests to gain this value. Use the `chmed(1M)` command to set the priority flag with the `p` option (for example, `chmed +p lt.AAA123`). Setting this flag takes effect for all submitted requests for the VSN that are not already preview requests. The age priority directive has the following format:

```
age_priority = factor
```

This directive is a static priority factor. Its overall effect is dynamic. The `age_priority` factor is multiplied by the number of seconds a request is a preview request. The result is added to the overall priority of the request. The longer a request waits to be satisfied, the larger the age factor becomes. Setting this factor helps to ensure that older requests are not indefinitely superseded by newer requests with other higher-priority factors.

If this factor is more than 1.0, it increases the importance of the time factor in calculating the total priority. If it is less than 1.0, it decreases the importance of the time factor. Setting the factor to 0.0 eliminates the time factor from the overall priority calculation.

A VSN whose priority flag is not set increases in priority based on the time it remains in the queue. Its priority can become higher than a VSN that comes into the queue later with the priority flag already set.

## Watermark Directives (Global or File System Specific)

The watermark preview request directives can be used as either global or file system specific directives. The watermark priority directives determine the

watermark priority (`wm_priority`) of the preview requests. [Figure 68](#) shows that the `wm_priority` factor is the sum of several settings.

**Figure 68.** `wm_priority` Calculation

$$\begin{array}{l}
 \text{lwm\_priority} + \\
 \text{lhwm\_priority} + \\
 \text{hlwm\_priority} + \\
 \text{hwm\_priority} \\
 \hline
 = \text{wm\_priority}
 \end{array}$$

When the `wm_priority` factor is a positive number, the result on the overall calculated priorities increases archiving requests over staging requests. However, the `wm_priority` factor can also be a negative number. In this case, the overall priority for archiving requests is reduced, which tends to favor staging requests over archival requests. A setting of 0.0 (or no specified command at all) indicates that no special action occurs to archival requests when the file system is in this condition. For more information on this, see the example in [“Example 1: Enforcing Stage Requests” on page 150](#).

[Table 63](#) shows the four watermark priority directives and their arguments

**Table 63. Watermark Priority Directives**

Priority Directive	Argument
<code>lwm_priority = value</code>	For <i>value</i> , specify the amount by which the <code>wm_priority</code> factor changes for archiving requests when the file system is below the LWM level. The default is 0.0.
<code>lhwm_priority = value</code>	For <i>value</i> , specify the amount by which the <code>wm_priority</code> factor changes for archiving requests when the file system crosses from below to above the LWM but remains below the HWM level. This generally indicates that the file system is filling up. The default is 0.0.
<code>hlwm_priority = value</code>	For <i>value</i> , specify the amount by which the <code>wm_priority</code> factor changes for archiving requests when the file system has crossed from above to below the HWM but remains above the LWM level. This generally indicates that the releaser was not able to free enough disk space to leave the file system below LWM. The default is 0.0.
<code>hwm_priority = value</code>	For <i>value</i> , specify the amount by which the <code>wm_priority</code> factor changes for archiving requests when the file system is above the HWM level. The default is 0.0.

Together, the four watermark settings create a dynamic priority factor that includes a percentage value indicating how full the file system is and the levels at which the HWM and LWM are set. The value assigned to a preview request is determined by whether a factor is global, specific to a file system, or not set.

When a file system crosses from one condition to another, the priority of each VSN associated with that file system is recalculated based on the appropriate watermark priority setting, with or without the `chmed(1M)` command's `p` option.

The watermark priorities are used only to calculate media requests for archiving. They are not used to calculate media requests for staging.

The following example directives show how to slightly increase the priority for archiving requests when the file system is at HLWM. [Figure 69](#) shows the settings to use to enable the releaser to free enough disk space so that the file system gets below LWM.

**Figure 69. Settings for Going Below the LWM**

```
lhwm_priority = -200.0
hlwm_priority = 100.0
```

## ■ Calculating Total Preview Request Priority

The numeric priority of preview requests is determined by combining several static and dynamic factors. Higher numbers correspond to higher priority. A static priority factor is set when the request is generated. Its effect does not change the overall priority after the request is generated and is waiting to be satisfied. A dynamic priority factor can increase or decrease the overall priority of a request while the request is waiting to be satisfied.

The total priority for a preview request is the sum of all priority factors. It is calculated as follows:

$$\text{total priority} = \text{vsn\_priority} + \text{wm\_priority} + (\text{age\_priority} * \text{time\_in\_sec\_as\_preview\_request})$$

## ■ How to Set Up a Preview Request Priority Scheme

It is necessary to change the default preview request FIFO scheme only when there are compelling system reasons to do so. The following possible conditions might necessitate changing the default preview request FIFO scheme:

- Condition 1: Ensure that staging requests are processed before archive requests.
- Condition 2: Ensure that archive requests gain top priority when a file system is about to fill up.

- Condition 3: Push requests that use a specific group of media to the top of the preview request list.

For environments in which user access to data is of paramount importance, the VSN drives are limited, or file archival is performed as a background function, you can use the `preview.cmd` file to influence how the storage system resources service the staging requests. You can customize the settings in the `preview.cmd` file to support any of the preceding scenarios and influence the configured StorageTek ASM environment.

Because data is not affected by the settings in this file, you are encouraged to experiment and adjust the directive settings to achieve the proper balance between archiving and staging requests when weighed against the priorities of each preview request.

Figure 70 shows an example `preview.cmd` file that addresses the three conditions listed previously.

**Figure 70. Example `preview.cmd` File**

```
# condition 1
lwm_priority = -200.0
lhwm_priority = -200.0
hlwm_priority = -200.0
# condition 2
hwm_priority = 500.0
# condition 3
age_priority = 1.0
```

## Example 1: Enforcing Stage Requests

The following example settings demonstrate one way to ensure that stage requests have priority over archive requests. This example assumes the following:

- Several requests are sitting in the queue for 100 seconds.
- The default `vsn_priority` is 1000.

Table 64 shows how the total request priorities are calculated.

**Table 64. Request Priority Example**

Priority	Calculation
Archive VSN with priority, LWM:	$1000 + (-200) + (1 \times 100) = 900$
Stage VSN with priority, LWM:	$1000 + 0 + (1 \times 100) = 1100$
Stage VSN without priority, LWM:	$0 + 0 + (1 \times 100) = 100$

This example shows that a negative value for `wm_priority` tends to favor staging requests over archival requests when the other factors are equal.

## Example 2: Enforcing Archive Requests

When the environment is balanced between the importance of staging a file back to the user versus getting new files archived to media, the biggest concern is exceeding the HWM. In this situation, if there are not enough files who have met their archive requirements to lower the percent full of the file system, completing the pending archive requests is the next best way to keep the file system from filling up.

In this situation, the `preview.cmd` file can be as simple as the following:

```
hwm_priority = 500.0
```

## Example 3: Prioritizing Requests by Media

In project-oriented environments, specific users might be working on groups of files that use specific VSNs and are segregated from other users. In this environment, certain projects might have higher priorities at certain times; hence, greater priority might be required from the available system storage resources. You can configure the `preview.cmd` file with the following directive to give users and their media the appropriate priority for media drives:

```
hwm_priority = 5000.0
```

Then, for every VSN in the priority user's group, enter the following information:

```
# chmed +p lt.AAA123 ## or whatever VSN is used
```

Thereafter, every request that requires VSN `AAA123` (or whatever VSN is used) is placed above other pending mount requests in the preview queue.

Later, to deprioritize the user's media, enter the following reverse command for every VSN:

```
# chmed -p lt.AAA123 ## or whatever media type is used
```

## Example 4: Complex Prioritization

Assume that there are two StorageTek ASM file systems with the following requirements:

- No request should sit too long in the queue (`age_priority`).

- When a file system is below the LWM, staging requests should take precedence.
- When a file system is above the LWM but below the HWM, it is not necessary to prioritize archive or stage requests one over the other. [Figure 71](#) shows the affected directives.

**Figure 71. Directives**

```
lwm_priority = -200.0
lhwm_priority = 0.0
hlwm_priority = 0.0
```

In this case, the other directives remain unchanged.

When a file system goes over the HWM, archive requests should take priority.

If both file systems are over the HWM, it is more important to prevent the second file system (for example, `samfs2`) from filling up. This might occur if, for example, `samfs1` is a user working file system and `samfs2` is the critical-system file system.

In all cases, regardless of the situation, a request for a select group of VSNs takes precedence in the preview request queue if the `chmed(1M)` command's `p` flag is set.

[Figure 72](#) shows a `preview.cmd` file that prioritizes requests according to the requirements in the preceding list.

**Figure 72. The `preview.cmd` File**

```
age_priority = 100.0
vsn_priority = 20000.0
lhwm_priority = -200.0
hlwm_priority = -200.0
fs = samfs1
hwm_priority = 1000.0
fs = samfs2
hwm_priority = 5000.0
```

# Recycling

---

*Recycling* is the process of reclaiming space on archive volumes. The recycler works with the archiver to reclaim the space occupied by unused archive copies. As users modify files, the archive copies associated with the old versions can be purged from the system. The recycler identifies the volumes with the largest proportions of expired archive copies and directs the moving of unexpired copies to different volumes. If only expired copies exist on a given volume, a site-defined action is taken. For example, such a volume can be relabeled for immediate reuse or exported to offsite storage, thus keeping a separate historical record of file changes. Users are unaware of the recycling process as it relates to their data files.

This chapter includes the following topics:

- [“Recycler Overview” on page 153](#)
- [“Recycling Directives” on page 155](#)
- [“Configuring the Recycler” on page 157](#)
- [“Troubleshooting the Recycler” on page 164](#)

## ■ Recycler Overview

The recycler keeps the amount of space consumed by expired archive copies to a minimum as defined by site-specified parameters. At any time, the space on a given archive volume consists of the following:

- *Current data* is space being used for archive images that are active currently.
- *Expired data* is space used by archive images that are no longer active currently.
- *Free space* is space that is not being used by currently active or expired archive images.

The *capacity* of a volume is the total amount of space for data on a volume. For example, a 10-gigabyte tape volume with 3 gigabytes written to it has a capacity of 10 gigabytes and 7 gigabytes of free space.

New or newly labeled archive media starts out with all its capacity as free space. As data is archived to the media, the amount of free space decreases and the amount of current data increases.

As archived files in the file system are changed or removed, their archive images expire and they move from the current data classification to the expired data classification. The physical space used by these images remains the same; there is simply no longer a file in the file system pointing to that space.

These expired images (and thus, expired data) would eventually consume all free space. Only when space is recycled can these images be removed and the space they occupy become free. The goal of the recycler is to transform space used by expired data into free space without losing any current data.

For example, removable media cartridges, such as tapes, can only be appended to. They cannot be rewritten in place. The only way to reuse a cartridge is to move all of the current data off of the cartridge, relabel the cartridge, and start using it again from the beginning.

You initiate recycling by entering the `sam-recycler(1M)` command. This can be done manually or through a `cron(1)` job. [Table 65](#) shows recycling methods.

**Table 65. Recycling Methods and Media Types**

Recycling Method	Media and Notes
By automated library	Removable media cartridges. When you archive by library, you put recycling directives in the <code>recycler.cmd</code> file.
By archive set	Removable media cartridges and disk. When you archive by archive set, you do not use a <code>recycler.cmd</code> file. You put all your recycling directives in the <code>archiver.cmd</code> file.

Note that you can recycle either by library or by archive set. As [Table 65](#) shows, if you are archiving to disk, you can recycle only by archive set.

The recycler and the archiver work together, as follows:

1. The recycler marks all the current (valid) archive images that are present on a volume with the `rearchive` attribute.
2. If you are archiving to removable media, the recycler marks the selected archive volume with the `recycle` attribute. This prevents the archiver from writing any more archive images to the volume.
3. The archiver moves all the marked images to another volume. This operation is called *rearchiving*. After the archiver moves the current archive images from the old volume to the new volume, the old volume contains only free space and expired space. If you are archiving to removable media cartridges, you can relabel and reuse the cartridge. If you are archiving to disk, the recycler removes the file that contains the expired archive images.



The recycler is designed to run periodically. It performs as much work as it can each time it is invoked. The recycler has to finish marking copies for rearchiving before the archiver can rearchive the files.

Sometimes expired archive images, with the `rearchive` attribute set, remain on media. This can happen under the following conditions:

- The archiver does not run after the recycler marks expired archive images.
- Media is not available for the archiver to use when moving the unexpired archive images.
- Miscellaneous other archiver anomalies.

Between executions, the recycler keeps state information in the library catalogs and the inodes. During the recycling process, you can use the `sls(1)` command and its `-D` option to display information about a file. The output from the `sls(1)` command shows whether or not a file is scheduled for rearchiving.

## ■ Recycling Directives

The `recycler.cmd` file accepts the directives described in the following sections:

- [“Specifying a Log File: the `logfile` Directive” on page 155](#)
- [“Preventing Recycling: the `no\_recycle` Directive” on page 156](#)
- [“Specifying Recycling for an Entire Automated Library: the Library Directive” on page 156](#)

### Specifying a Log File: the `logfile` Directive

The `logfile` directive specifies a recycler log file. This directive has the following format:

```
logfile = filename
```

For *filename*, specify the path to the log file.

The following is an example of a `logfile=` directive line:

```
logfile=/var/adm/recycler.log
```

## Preventing Recycling: the `no_recycle` Directive

The `no_recycle` directive enables you to prevent recycling of volumes. To specify the VSNs, you use regular expressions and one or more specific media types. This directive has the following format:

```
no_recycle media_type VSN_regex [ VSN_regex ... ]
```

**Table 66. Arguments for the `no_recycle` Directive**

Argument	Meaning
<code>media_type</code>	Specify a media type from the <code>mcf(4)</code> man page.
<code>VSN_regex</code>	Specify one or more space-separated regular expressions to describe the volumes. For information on the format of a regex, see the <code>regex(5)</code> man page or see <a href="#">“File Name search_criteria Using Pattern Matching: -name regex” on page 75.</a>

By specifying a `media_type`, you can prevent the recycling of volumes stored on a particular type of media. One or more `VSN_regex` specifications enables you to use a regular expression to identify specific cartridges to be excluded from recycling.

For example, the following directive line excludes from recycling any tape volumes whose VSN identifiers begin with `DLT`:

```
no_recycle lt DLT.*
```

## Specifying Recycling for an Entire Automated Library: the `Library` Directive

The `library` directive enables you to specify various recycling parameters for the VSNs associated with a specific library. This directive has the following format:

```
library parameter [ parameter ... ]
```

For `library`, specify the library’s name as specified in the Family Set field of the `mcf(4)` file.

For *parameter*, specify one or more space-separated *parameter* keywords from [Table 67](#).

**Table 67. Library Directive *parameter* Values**

<i>parameter</i>	Action
-dataquantity <i>size</i>	Limits the amount of data that the recycler can schedule for rearchiving in its efforts to clear volumes of useful data. Default is 1 gigabyte.
-hwm <i>percent</i>	Library high watermark. Default is 95.
-ignore	Prevents volumes in this library from being recycled. This directive is useful when testing the recycler.cmd file.
-mail [ <i>email_address</i> ]	Sends email messages to the designated <i>email_address</i> . By default, no email is sent. If -mail is specified with no argument, email is sent to root.
-mingain <i>value</i>	Minimum VSN gain. Default is 50.
-vsncount <i>count</i>	Limits the number of volumes to be recycled to count. Default is 1.

For example, consider the following directive line:

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

It specifies the following for library `gr47`:

- The library should be considered for recycling when the volumes in the library are 85 percent full.
- The minimum percent gain is 40 percent.
- No more than 1 gigabyte is to be rearchived. This is the default, so it is not specified in the `recycler.cmd` file.
- Only one volume is to be recycled. This is also a default setting.
- Recycling messages are emailed to `root`.

## ■ Configuring the Recycler

Prior to configuring the recycler, note the following:

- Directives in the `archiver.cmd` file control recycling by archive set. Directives in the `recycler.cmd` file control recycling by library. In addition, the `recycler.cmd` file controls general recycler behavior. For information on recycler directives, see [“Recycling Directives” on page 155](#).

- Do not recycle volumes that contain removable media files. You create removable media files by using the `request(1)` command. The recycler does not preserve removable media files created by the `request(1)` command. A volume with removable media files can never be drained.
- Do not run the recycler while performing maintenance on a StorageTek ASM file system. The recycler uses the `.inodes` file and the `mcf` file to help identify files that are current or expired and the devices associated with a file system. Absence of proper information in these files can cause current archived data to appear as expired and be recycled.
- All StorageTek ASM file systems must be mounted when the recycler is run. If you are recycling from online disk, the file system that contains the disk volumes must be mounted and the host system must be accessible.

The recycler is not enabled by default. You must initiate recycling by entering the `sam-recycler(1M)` command. When the recycler is initiated, the default recycler settings specified in [“Specifying Recycling for an Entire Automated Library: the Library Directive” on page 156](#) take effect. For more information on the recycler, see the `sam-recycler(1M)` man page.

The following sections describe the process for configuring the recycler. This process includes the following steps:

- [“Step 1: Creating a `recycler.cmd` File \(Optional\)” on page 158](#)
- [“Step 2: Editing the `archiver.cmd` File \(Optional\)” on page 161](#)
- [“Step 3: Running the Recycler” on page 162](#)
- [“Step 4: Creating a `crontab` File for the Recycler \(Optional\)” on page 163](#)
- [“Step 5: Removing `-recycle\_ignore` and `ignore` Parameters” on page 164](#)
- [“Step 6: Creating a `recycler.sh` File \(Optional\)” on page 164](#)

If you are archiving to cartridges in a library, this process includes creating a `recycler.cmd` file and, optionally, editing the `archiver.cmd` file. If you are archiving to disk, you can archive only by archive set, so to enable recycling of these disk volumes, you edit the `archiver.cmd` file. The following procedure describes configuring the recycler for any archive media.

## Step 1: Creating a `recycler.cmd` File (Optional)

Perform this step if you are recycling archive copies on cartridges in a library.

If you are recycling archive copies on disk volumes, you cannot complete this step because recycling is controlled by directives in the `archiver.cmd` file. For information on the configuring recycling in the `archiver.cmd` file, see [“Step 2: Editing the `archiver.cmd` File \(Optional\)” on page 161](#).

The `recycler.cmd` file contains general recycling directives and can also contain directives for each library in the StorageTek ASM environment. For information on the recycling directive, see [“Recycling Directives” on page 155](#).

Even if you are recycling by archive set, you still should configure each library in the `recycler.cmd` file. This ensures that VSNs that do not fall into an archive set can be recycled if needed.

A typical `recycler.cmd` file contains the following directive lines:

- A `logfile=` directive line to specify a recycler log file. The system writes recycling messages and recycling reports to this file.
- One or more directive lines for each library that contains volumes to be recycled. This line must contain the Family Set name (from the `mcf` file) for the library being recycled. This identifies the library to the recycler.

Because you are still creating the `recycler.cmd` line, and it has not yet been tested, use the `ignore` keyword. You remove the `ignore` keyword in a later step in this process.

To create a `recycler.cmd` file, perform the following steps:

1. Become superuser.
2. Use `vi(1)` or another editor to open file `/etc/opt/SUNWsamfs/recycler.cmd`.
3. Add one or more directives described in this chapter to control recycler activity.
4. Save and close the file.

## Example `recycler.cmd` File

[Figure 73](#) shows an example of a `recycler.cmd` file.

**Figure 73. A `recycler.cmd` File Example**

```
logfile = /usr/tmp/recycler.log
stk30 -hwm 51 -mingain 60 -ignore -mail root
```

The following sections describe the parameters specified in [Figure 73](#).

### **The `-hwm 51` Parameter**

By specifying a high watermark, you can set the percentage of media usage below which recycling cannot occur. This percentage is the ratio of the used space in the library to its total capacity. As an example, a library that holds 10 20-gigabyte tapes, three of them 100 percent full and the remaining seven each 30 percent full, has the following media utilization percentage:

$$((3 * 1.00 + 7 * 0.30) * 20G) / (10 * 20G) * 100\% = 51\%$$

Note that this calculation does not distinguish between current data and expired data. It only addresses the amount of media used.

In this example, if the high watermark is 51 percent or less, the recycler does not automatically select any of the automated library's VSNs for recycling.

**Note:** You can force a VSN to be recycled by using the following command to set the recycling flag:

```
# chmed +c lt.AAA123
```

When the `+c` flag is set, the archiver does not write any more archive images to the volume. The `+c` flag can be viewed through the `samu(1M)` utility. For more information, see the `chmed(1M)` and `samu(1M)` man pages.

### **The `-mingain 60` Parameter**

The *minimum VSN gain percentage* sets a lower limit on the amount of space to be gained by recycling a cartridge. For example, if a cartridge in an automated library is 95 percent current data and 5 percent expired data, the gain obtained by recycling the cartridge is only 5 percent. It might not be worth moving the other 95 percent to retrieve this space. Setting the minimum-gain to 6 percent or more inhibits the recycler from automatically selecting this example VSN.

Another example is a cartridge with 90 percent expired data, 5 percent current data, and 5 percent free space. This would have a gain of 90 percent if recycled.

### **The `-ignore` Parameter**

The `-ignore` parameter keeps the recycler from recycling a particular library and should be used when you are configuring the recycler.

### **The `-mail root` Parameter**

The `-mail` parameter specifies that the recycler send mail when recycling occurs on a given library. The mail message has the following subject line:

```
Robot robot-name recycle
```

[Table 66](#) shows sample message bodies.

## **Figure 74. Sample Recycling Messages**

```
I will recycle VSN vsu.
Cannot find any candidate VSN in this media changer.
Previously selected VSN vsu is not yet finished recycling.
Previously selected VSN vsu is now finished recycling. It will now
be post-recycled.
```

## Step 2: Editing the `archiver.cmd` File (Optional)

Perform this step if you are recycling by archive set. If you are archiving to disk, recycling by archive set is the only means of recycling that is possible, so if you are archiving to disk, you must complete this step in order to recycle.

If you are recycling by library, you can proceed to the next step.

- To edit the `archiver.cmd` file, perform the procedure called “[To Create or Modify an `archiver.cmd` File and Propagate Your Changes](#)” on [page 59](#).

The directives you add to the `archiver.cmd` file to enable recycling by archive set, must appear between `params` and `endparams` directives. [Table 68](#) shows the archive set recycling directives that you can use.

**Table 68. Archive Set Recycling Directives**

Directive	Function
<code>-recycle_dataquantity <i>size</i></code>	Limits the amount of data that the recycler can schedule for rearchiving in its efforts to clear volumes of useful data.
<code>-recycle_hwm <i>percent</i></code>	Sets the high watermark percentage.
<code>-recycle_ignore</code>	Prevents the archive set from being recycled.
<code>-recycle_mailaddr <i>mail_address</i></code>	Sends recycler messages to <i>mail_address</i> .
<code>-recycle_mingain <i>percent</i></code>	Limits recycling to those VSNs that would increase their free space by <i>percent</i> or more.
<code>-recycle_vsncount <i>count</i></code>	Limits the number of volumes to be rearchived to <i>count</i> .

For more information about the preceding directives, see “[Archiving](#)” on [page 47](#) or see the `archiver.cmd(4)` man page.

[Figure 75](#) shows an `archiver.cmd` example for recycling disk archives.

**Figure 75. Disk Archiving Specifications in the `archiver.cmd` File**

```
fs = samfs1
  1 2m

arset0 testdir0
  1 2m
  2 4m

arset1 testdir1
  1 2m
```

**Figure 75. Disk Archiving Specifications in the archiver.cmd File (Continued)**

```

2 4m

params
arset0.1 -disk_archive disk01 -recycle_hwm 80 \
        -recycle_mingain 20 -recycle_ignore
arset1.1 -disk_archive disk02 -recycle_hwm 80 \
        -recycle_mingain 20 -recycle_ignore
endparams

```

### Step 3: Running the Recycler

1. Issue the `sam-recycler(1M)` command.  
The recycler reads the `recycler.cmd` file.
2. Examine the standard output, log, ASM log, and `/var/adm/messages` for any error messages from the recycler.  
Correct your files if errors appear.

Figure 76 shows a sample recycler log file for recycling removable media cartridges.

**Figure 76. Recycler Log File Example for Removable Media Cartridges**

```

===== Recycler begins at Wed Dec 12 14:05:21 2001 =====
Initial 2 catalogs:

0  Family: m160                Path: /var/opt/SUNWsamfs/catalog/m160
   Vendor: ADIC                Product: Scalar 100
   SLOT      ty      capacity    space vsn
     0       at       25.0G       25.0G CLN005
     1       at       48.5G       6.1G 000003
     2       at       48.5G       32.1G 000004
     3       at       48.5G       35.1G 000005
     4       at       48.5G       44.6G 000044
     5       at       48.5G       45.1G 000002
     6       at       48.5G       45.9G 000033
     7       at       48.5G       48.5G 000001

Total Capacity: 364.8G bytes, Total Space Available: 282.3G bytes
Volume utilization 22%, high 95% VSN_min 50%
Recycling is ignored on this robot.

1  Family: hy                  Path: /var/opt/SUNWsamfs/catalog/historian
   Vendor: Sun SAM-FS          Product: Historian
   SLOT      ty      capacity    space vsn
   (no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.

```



**Figure 76. Recycler Log File Example for Removable Media Cartridges (Continued)**

```

8 VSNs:

      ---Archives---      -----Percent-----      m160
----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN              0          0          0    87    13      m160:at:000003
no-data VSN              0          0          0    33    67      m160:at:000004
no-data VSN              0          0          0    27    73      m160:at:000005
no-data VSN              0          0          0     8    92      m160:at:000044
no-data VSN              0          0          0     7    93      m160:at:000002
no-data VSN              0          0          0     5    95      m160:at:000033
empty VSN                0          0          0     0    100     m160:at:CLN005
empty VSN                0          0          0     0    100     m160:at:000001

Recycler finished.

===== Recycler ends at Wed Dec 12 14:05:32 2001 =====

```

Figure 77 shows a sample recycler log file for recycling disk archive files.

**Figure 77. Recycler Log File Example for Disk Archive Files**

```

---Archives---      -----Percent-----
----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
new candidate          0          0          0    41    59      <none>:dk:disk01

677 files recycled from VSN disk01 (mars:/sam4/copy1)
0 directories recycled from VSN disk01 (mars:/sam4/copy1)

```

## Step 4: Creating a `crontab` File for the Recycler (Optional)

If the system is performing as expected, you are ready to make a `crontab` entry for the superuser to run the recycler periodically. You might want to run the recycler no more than once every two hours, depending on your site's conditions.

- Create a `crontab` entry.

For information about this, see the `cron(1M)` man page.

The following example entry in `root`'s `crontab` file ensures that the `cron` daemon runs the recycler every five minutes after the hour for every odd-numbered hour:

```
5 1,3,5,7,9,11,13,15,17,19,21,23 * * * /opt/SUNWsamfs/sbin/sam-recycler
```

## Step 5: Removing `-recycle_ignore` and `ignore` Parameters

1. Use `vi(1)` or another editor to remove the `-recycle_ignore` parameters from the `archiver.cmd` file.
2. Use `vi(1)` or another editor to remove the `ignore` parameters from the `recycler.cmd` files.

You are now recycling.

## Step 6: Creating a `recycler.sh` File (Optional)

Perform this step if you are recycling archive copies on removable media cartridges. If you are archiving only to disk, do not perform this step.

The recycler executes the `recycler.sh` script when all the current images from a VSN have been rearchived to another VSN. For an example, see the `recycler.sh(1M)` man page. Another example, found in `/opt/SUNWsamfs/examples/recycler.sh`, shows how to relabel a recycled VSN and send mail to the superuser.

The recycler called the `/opt/SUNWsamfs/sbin/recycler.sh` script with the following arguments:

```
Media type: $1 VSN: $2 Slot: $3 Eq: $4
```

The `/opt/SUNWsamfs/sbin/recycler.sh` script is called when the recycler determines that a VSN has been drained of all known active archive copies. You should determine your site requirements for dispensing with recycled cartridges. Some sites choose to relabel and reuse the cartridges; others choose to remove the cartridges from the automated library to use later for accessing historical files. For more information, see the `recycler(1M)` and `recycler.sh(1M)` man pages.

## ■ Troubleshooting the Recycler

The most frequent problem encountered with the recycler occurs when the recycler generates a message similar to the following when it is invoked:

```
Waiting for VSN mo:OPT000 to drain, it still has 123 active archive copies.
```

One of the following conditions can cause the recycler to generate this message:

- Condition 1: The archiver fails to rearchive the 123 archive copies on the volume.

- Condition 2: The 123 archive copies do not refer to files in the file system. Rather, they refer to 123 metadata archive copies.

Condition 1 can exist for one of the following reasons:

- Files that need to be rearchived are marked `no_archive`.
- Files that need to be rearchived are in the `no_archive` archive set.
- Files cannot be archived because there are no available VSNs.
- The `archiver.cmd` file contains a `wait` directive.

To determine which condition is in effect, run the recycler with the `-v` option. As [Figure 78](#) shows, this option displays the path names of the files associated with the 123 archive copies in the recycler log file.

**Figure 78. Recycler Messages**

```
Archive copy 2 of /sam/fast/testA resides on VSN LSDAT1
Archive copy 1 of /sam3/tmp/dir2/filex resides on VSN LSDAT1
Archive copy 1 of Cannot find pathname for file system /sam3 inum/
gen 30/1 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfila00 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilF82 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilV03 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfila06 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfila33 resides on VSN LSDAT1
Waiting for VSN dt:LSDAT1 to drain, it still has 8 active archive
copies.
```

In this example output, messages containing seven path names are displayed along with one message that includes `Cannot find pathname... text`. To correct the problem with `LSDAT1` not draining, you need to determine why the seven files cannot be rearchived. After the seven files are rearchived, only one archive copy is not associated with a file. Note that this condition should occur only as the result of a system crash that partially corrupted the `.inodes` file.

To solve the problem of finding the path name, run `samfsck(1M)` to reclaim orphan inodes. If you choose not to run `samfsck(1M)`, or if you are unable to unmount the file system to run `samfsck(1M)`, you can manually relabel the cartridge after verifying that the `recycler -v` output is clean of valid archive copies. However, because the recycler continues to encounter the invalid inode remaining in the `.inodes` file, the same problem might recur the next time the VSN is a recycle candidate.

Another recycler problem occurs when the recycler fails to select any VSNs for recycling. To determine why each VSN was rejected, you can run the recycler with the `-d` option. This displays information on how the recycler selects VSNs for recycling.



# Upgrading the Hardware in Your Environment

---

This chapter describes how to upgrade the hardware within your existing StorageTek ASM environment. The following topics are presented:

- [“Adding Slots in an Automated Library” on page 168](#)
- [“Upgrading or Replacing a Library” on page 169](#)
- [“Upgrading DLT Tape Drives” on page 172](#)

Certain other types of operations and upgrades also need to be performed within a StorageTek ASM environment. The following publications describe these operations:

- The *StorageTek QFS and StorageTek ASM Software Installation Guide* describes upgrading StorageTek QFS and StorageTek ASM software.
- The *StorageTek QFS and StorageTek ASM File System Administration Guide* describes the following types of operations and upgrades:
  - Initializing a File System
  - Propagating Configuration File Changes to the System
  - Mounting a File System
  - Unmounting a File System
  - Checking File System Integrity and Repairing File Systems
  - Preserving Information for an Upgrade
  - Preparing for a Hardware Device Upgrade
  - Adding Disk Cache to a File System
  - Replacing Disks in a File System
  - Upgrading a Host System
  - Upgrading the Solaris OS

## ■ Adding Slots in an Automated Library

Your software license controls the number of cartridge slots managed by the StorageTek ASM system. To increase the number of slots, follow the steps in this section.

### To Add Slots In a Library

1. Determine whether you need to obtain a new set of license keys through your authorized service provider (ASP) or, if an ASP is not assigned to your account, through StorageTek.

If you do not need new license keys, continue with Step 2.

If you need new license keys, perform the following steps:

- a. Replace the existing license keys with the new license keys.

Perform this step if you obtained new license keys. License keys start in column 1 of the following file:

```
/etc/opt/SUNWsamfs/LICENSE.4.1
```

No other keywords, host IDs, or other information can appear.

- b. Issue the `samd(1M) config` command to enable the StorageTek ASM software to recognize the new license keys. (Optional)

Perform this step if you obtained new license keys. For example:

```
# samd config
```

2. Unload the library catalog.

Use the `sacmd(1M) unload` command in the following format:

```
sacmd unload eq
```

For *eq*, specify the Equipment Ordinal of the automated library as defined in the `mcf` file. This command moves the library catalog entries into the historian catalog and preserves the catalog information for each cartridge.

After this command executes, you can use `samu(1M) v display` to observe the automated library's `v display` empty and the historian's `v display` fill up with the VSNs that used to be in the automated library.

3. Bring down the StorageTek ASM system.

For information on how to perform this step, see [“Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment” on page 13.](#)

4. Power down the host system and the library according to the manufacturer’s suggested procedure.
5. Have the library hardware engineer add slots to the automated library.
6. Power on the host system using your normal start-up procedure.
7. Start the StorageTek ASM system.

For information about how to perform this step, see [“Using Automated Libraries and Manually Loaded Drives in a StorageTek ASM Environment” on page 13.](#) The new license information appears in the `samu(1M)` utility’s `l` display.

## ■ Upgrading or Replacing a Library

Prior to disconnecting and installing a different automated library, prepare for the upgrade as described in the Storagetek QFS and StorageTek ASM File System Administration Guide in the section called "Preparing for Hardware Upgrades."

### To Replace or Upgrade a Library

1. Unload the library catalog.

Use the `sacmd(1M) unload` command in the following format:

```
sacmd unload eq
```

For `eq`, specify the Equipment Ordinal of the automated library as defined in the `mcf` file. This command moves the library catalog entries into the historian catalog and preserves the catalog information for each cartridge.

After this command executes, you can use `samu(1M) v` display to observe the automated library’s `v` display empty and the historian’s `v` display fill up with the VSNs that used to be in the automated library.

2. Update the `/etc/opt/SUNWsamfs/inquiry.conf` file. (Optional)

The new library should be identified in this file by the vendor, the automated library model, and a StorageTek ASM internal name.

For example, the released `inquiry.conf` file includes the following line:

```
"HP",    "C1710T",  "hpoplib"  # HP optical library
```

This line indicates that if the system detects a SCSI device made by vendor HP of model C1710T, the system drives it as an `hpoplib`. The first two fields (vendor/product) are returned from the hardware devices. The last field, `hpoplib`, is a name that the system uses internally to determine how to communicate with the device. If the `inquiry.conf` file needs to be changed, the change does not become effective until the `sam-amld` daemon is restarted.

3. Save the current `/etc/vfstab` file as `/etc/vfstab.cur`.
4. Edit the `/etc/vfstab` file.  
Change any StorageTek ASM mounts from `yes` to `no`.
5. Save the `/etc/opt/SUNWsamfs/archiver.cmd` file as `archiver.cmd.cur`.
6. Edit the `/etc/opt/SUNWsamfs/archiver.cmd` file  
Add a `wait` directive as the first line.
7. Power off the host system and peripherals using the manufacturer's suggested procedure.
8. Disconnect the automated library.
9. Attach the connecting cable to the new automated library.
10. Power on the peripherals and the host system using the suggested power-on sequence.
11. Ensure that the host system identifies the new automated library.

Enter the following command:

```
> probe-scsi-all
```

The new automated library and its drives must display prior to proceeding. If these devices are not identified, the automated library and its drives probably have a connection problem.

12. Boot the system.

Enter the following command to boot with the new configuration:

```
> boot -rv
```

13. If the target numbers of the drives or automated library changed, or if the ordering or number of the drives in the automated library changed, modify the `/etc/opt/SUNWsamfs/mcf` file to reflect the new configuration.  
(Optional)



This is similar to an initial installation as described in the *StorageTek QFS and StorageTek ASM Software Installation Guide*.

14. Create new `/dev/samst` entries. (Optional)

Perform this step if you are adding new equipment. Enter the following command:

```
# samdev
```

15. Initialize the StorageTek ASM system.

You can perform this step by either mounting a file system or by typing the following command:

```
# samd start
```

When the system initializes, it recognizes that the number of slots in the automated library has changed. The system runs a full audit on the automated library to update the library catalog. A full audit must be completed before archiving is resumed.

If there are problems in the audit, the most likely reason is that the ordering of the drives in the automated library does not match the ordering in the `/etc/opt/SUNWsamfs/mcf` file. Remember, that drives have two attributes: the SCSI target ID and the position in the automated library. Both of these attributes must be correct both before and after the upgrade.

If the audit completes without problems, proceed to the next step.

16. Replace the

`/etc/vfstab` and `/etc/opt/SUNWsamfs/archiver.cmd` files with the pre-upgrade versions.

Use the saved `/etc/vfstab.cur` and `/etc/opt/SUNWsamfs/archiver.cur` files, respectively.

17. Reboot the system to ensure that no errors exist in the configuration.

The automated library calls the drives by position number. When the system wants to load a cartridge into a drive, it must, for example, send a command to the automated library to load a cartridge from slot 123 into drive 3.

Drive 3 might be SCSI target 6 based on the third `mcf` entry. The system knows it is drive 3 because it is the third drive entry in the `mcf` file. The automated library knows that it is drive 3 because of the physical location it occupies in the automated library.

After the automated library receives a request to load the cartridge into the drive, the system tests the drive for unit ready status. Here, the system

uses the SCSI target ID as defined in the `/dev/samst/scsi-target` entry in the `mcf` file. Therefore, it is important that the entry match the drive that was just loaded with the cartridge.

There is no good way to determine this information. Usually, the manufacturer ships the automated library set up with ascending SCSI IDs on the drives, but there is no guarantee of this. One way to determine this is to use the `samu(1M)` utility's `:load` command to load a cartridge, and then watch the `samu(1M)` utility's `s` display to see which drive shows the `r`, rather than the `p`, in the status flags of the `t` display.

## ■ Upgrading DLT Tape Drives

To take advantage of higher density and faster tape technology, you can upgrade DLT tape drives in an automated library or a standalone tape drive. For example, you can move from DLT 4000 drives to DLT 7000 drives.

In a StorageTek ASM environment, this is a matter of adding the new drive, rebooting the new configuration, and updating the `mcf` file as necessary prior to starting the StorageTek ASM environment. In addition, if you are upgrading the number of slots, you need to contact your authorized service provider (ASP) or StorageTek because you might need an upgraded license.

Note the following restrictions and general information before upgrading drives:

- StorageTek ASM environments do not support mixed DLT tape drives within the same direct-attached automated library. For example, a StorageTek ASM system cannot differentiate between a DLT 4000 tape drive and a DLT 7000 tape drive in the same automated library. Therefore, you should plan on replacing all the DLT drives with the new drives at the same time.
- The lower-density tapes can coexist with higher-density tapes and tape drives. You can continue to read and write to the lower-capacity tapes using a higher-density drive.
- To take full advantage of the higher-density DLT tapes, you might want to recycle existing files and migrate them to a higher-density tape. You can accomplish this by marking all the lower-density tapes as read-only and then marking these tapes to be recycled. For information on recycling tape, see [“Recycling” on page 153](#).
- As each tape is labeled, the density of the tape is acknowledged and recorded in the library catalog.

## To Upgrade Tape Drives

1. Decide whether or not your current dump files are sufficient. (Optional)

If you decide that they are not, perform a `samfsdump(1M)` on your file systems prior to proceeding to the next step.

2. Update the `/kernel/drv/st.conf` file to identify the new drives.

The tape drives are identified in this file by the vendor, the tape model, and a StorageTek ASM internal name. For example, the released `st.conf` file contains the following line:

```
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape"
```

An example file is provided in `/opt/SUNWsamfs/examples/st.conf_changes`. You can read in the entire file to `/kernel/drv/st.conf` or you can merge the necessary changes. For more information on updating the `st.conf` file, see the StorageTek QFS and StorageTek ASM Software Installation Guide.

3. Power off the host system and peripherals using the manufacturer's suggested procedure.
4. Replace the tape drives with the new drives.
5. Power on the peripherals and host system using the suggested power-on sequence.
6. Ensure that the host system identifies the new drives.

Enter the following command:

```
> probe-scsi-all
```

The automated library and the new drives must be appear prior to proceeding. If these devices do not appear, a connection problem probably exists and needs to be corrected. After this command returns the expected information, you can proceed to the next step.

7. Boot the system.

Enter the following command to boot with the new configuration:

```
> boot -rv
```

8. Modify the `/etc/opt/SUNWsamfs/mcf` file to reflect the new configuration. (Optional)

Perform this step if the target numbers of the drives or automated library changed, or if the ordering or number of the drives in the automated library

changed. This is similar to an initial installation as described in the *StorageTek QFS and StorageTek ASM Software Installation Guide*.

9. Create new `/dev/samst` entries for the new equipment. (Optional)

To create these entries, enter the following command:

```
# samdev
```

10. Start your StorageTek ASM system.
11. Mount the file systems.

You can now continue to use the existing StorageTek ASM tapes.

# Advanced Topics

This chapter discusses advanced topics that are beyond the scope of basic system administration and usage.

The following topics are presented.

- “Device Logging” on page 175
- “Removable Media Files” on page 178
- “Segmented Files” on page 180
- “System Error Facility Reporting” on page 181

## ■ Device Logging

The device-logging facility provides device-specific error information that you can use to analyze certain types of device problems. It can help to determine a failing sequence of events for an automated library, tape drive, or optical drive. Note that the device-logging facility does not collect soft media errors (such as recoverable read errors).

Device-logging messages are written to individual log files. There is a log file for each automated library, each tape and optical drive device, and one for the historian. The log files are located in `/var/opt/SUNWsamfs/devlog`. The name of each individual log file is the same name as the Equipment Ordinal.

**Example.** Assume a StorageTek ASM file system and a single Hewlett Packard optical library with two optical drives.

Figure 79 shows the `mcf` file.

**Figure 79. Example `mcf` File**

```
/dev/samst/clt5u0 40 hp hp40 - etc/opt/SUNWsamfs/hp40_cat
/dev/samst/clt4u0 41 mo hp40 -
/dev/samst/clt6u0 42 mo hp40 -
```

Figure 80 shows the `/var/opt/SUNWsamfs/devlog` file.

**Figure 80. The `devlog` File**

```
# pwd
/var/opt/SUNWsamfs/devlog
```

**Figure 80. The devlog File (Continued)**

```
# ls
40      41      42      43
#
```

Device 43 is the historian.

## When to Use the Device Log

The device log can easily generate many log messages, especially when all logging options for all devices are turned on and there is a great deal of device activity. Initially, the device log settings are set to the following default values:

```
err retry syserr date
```

If you suspect there is a problem with one of the devices configured within a StorageTek ASM environment, it is appropriate to enable additional logging events for that device. Also, it is appropriate to enable device logging if you are advised to do so by your service provider. In these situations, set the event to `detail`. In extreme cases, you might be advised by your service provider to set the event to `all` for a device. This adds additional log information. However, in general, it is probably not useful or practical to run the system with excessive logging.

The device log information is collected automatically when the `samexplorer(1M)` command is issued. This allows the file system service to review any possible device error information as part of problem analysis activity.

## Enabling the Device Log

There are two methods you can use to enable the device log.

For both method 1 and method 2:

- `eq` is the Equipment Ordinal of the device from the `mcf` file or the keyword `all` for all equipment.
- The device-logging events are listed in the `samset(1M)` man page. They are also listed in [“Enabling the Device Log” on page 176](#). Note that the device log messages are available only in English text. An *event* is one or more event types from the following list:
  - `all`
  - `date`
  - `default`
  - `detail`
  - `err`
  - `event`

- label
- mig
- module
- msg
- none
- retry
- stage
- stage\_ck
- syserr
- tapealert
- time

You can enable the device log in one of two ways. These procedures are as follows:

- [“To Enable the Device Log By Using the `samset\(1M\)` Command” on page 177](#)
- [“To Enable the Device Log by Editing the `defaults.conf` File” on page 177](#)

## To Enable the Device Log By Using the `samset(1M)` Command

- Use the `samset(1M)` command.

For example:

```
# samset devlog eq event
```

For more information about the `samset(1M)` command, see the `samset(1M)` man page.

## To Enable the Device Log by Editing the `defaults.conf` File

1. Become superuser.
2. Use `vi(1)` or another editor to open file `/etc/opt/SUNWsamfs/defaults.conf`.
3. Add the `devlog` directive in the `defaults.conf` file.

Add the following directive:

```
devlog eq event
```

For `eq`, specify the Equipment Ordinal of the device for which you want to log messages.

For `event`, specify one or more of the events described in [“Enabling the Device Log” on page 176](#). If you specify more than one event, separate them with space characters.

When a StorageTek ASM file system starts up, it automatically sets the event type for each available device to `default`. You can also use the `samset(1M)` command to determine the present settings for each device log.

4. Save and close the `defaults.conf` file.
5. Use the `samd(1M) config` command to propagate the `defaults.conf` file changes.

```
# samd config
```

## ■ Removable Media Files

You can use the `request(1)` command to manually create, write, and read files that do not use the disk cache for buffering the data. Files created in this manner are called *removable media files*.

Removable media files look like typical StorageTek ASM files in that they have permissions, a user name, a group name, and size characteristics. However, the data does not reside in the disk cache. Thus, you can create files that are larger than the disk cache and write them to removable media cartridges. The system creates an inode entry in the `.inodes` file for the file that you specify on the `request(1)` command. The user does not need to know where the file begins on the removable media. (It is the same for a file with data in the disk cache.) The StorageTek ASM file systems read that information from the inode entry. Multiple removable media files can reside on the same volume.

If the removable media file spans multiple volumes, it is called a volume overflow file. The *volume overflow* feature enables the system to span a single file over multiple volumes on multiple cartridges. A volume overflow file is a type of removable media file. The volume overflow feature is useful if you have very large files that exceed the capacity of their chosen media.

## To Create a Removable Media or Volume Overflow File

1. Use the `tplabel(1M)` or `odlabel(1M)` command to label a tape or magneto-optical cartridge.

For information on these commands, see their respective man pages.

2. Use the `request(1)` command.

At a minimum, use the following options:

```
request -m media_type -v vsn [vsn/vsn ...] [-l vsn_file] input_file
```



**Table 69. Arguments for the `request(1)` Command**

Argument	Meaning
<i>media_type</i>	The media type of the removable media cartridge. For information about valid <i>media_type</i> specifications, see the <code>mcf(4)</code> man page.
<i>vsn</i>	The Volume Serial Name of the removable media cartridge. If you specify more than one <i>vsn</i> , you are creating a volume overflow file. You can specify up to 256 <i>vsns</i> for volume overflow files. Use forward slash characters (/) to separate the <i>vsn</i> arguments. The <i>vsns</i> specified should not be among the volumes that are used in a StorageTek ASM environment for automated archiving. Archiving appends the next file to be archived to the end of the current data and moves the EOF label beyond the data each time.
<i>vsn_file</i>	An input file that contains a list of <i>vsns</i> . If you have many <i>vsns</i> , it can be easier to specify the list of <i>vsns</i> in an input file rather than on the command line.
<i>input_file</i>	The file to be written to the removable media cartridge. This file must reside in a StorageTek ASM file system.

**Example 1.** The following command creates a removable media file:

```
# request -m lt -v aaa rem1
```

**Example 2.** The following command creates a volume overflow file on three volumes:

```
# request -m lt -v TAPE01/TAPE02/TAPE03 large.file
```

You must read and write removable media files sequentially. The StorageTek ASM file system automatically mounts the requested volume if the volume resides in an automated library defined in the `mcf` file.

The presence of a removable media file on a volume prevents that volume from being recycled. The recycler expects that only archived files reside on the particular volume that is assigned for archiving. In addition, the removable media files are never archived.

Removable media files are not supported over NFS.

Using the `request(1)` command bypasses the typical functions of the archiver.

For examples that describe how to create removable media files, see the `request(1)` man page.

## ■ Segmented Files

The StorageTek ASM environments support segmented files. Segmenting files improves tape storage retrieval speed, access, and manageability for very large files. A segmented file can be larger than the physical disk cache. With a segmented file, it is possible for only part of a file to reside on the disk cache at any one time.

The `segment(1)` command allows you to specify the segment size. You cannot set a segment size that is smaller than the current file size.

Segmented files support tape striping. After a file is segmented, it can be striped simultaneously over multiple tape devices, which significantly reduces the time needed to store the file segments. Data access is accelerated by allowing users to retrieve only the desired file segments rather than the entire file.

Segmentation can enhance archiving efficiency because only changed portions of a file are rearchived. Segments of a file can be archived in parallel, and segmented files can be staged in parallel. This increases performance when archiving and retrieving.

Segmentation can be enabled on a file, directory, or entire file system. Segmented files support all other StorageTek ASM capabilities.

The following sections describe how segmented files differ from nonsegmented files. For more information about segmented files, see the `segment(1)` or the `sam_segment(3)` man pages.

## Archiving

For a segmented file, the archivable unit is the segment itself, not the file. All archiving properties and priorities apply to the individual segments, and not to the file.

The unit archived is the segment. You can stripe a segment by specifying both the `-drives` and `-drivemin` parameters for the archive set in the `archiver.cmd` file.

For example, assume that there is a 100-megabyte segmented file in the file system, and its segment size is 10 megabytes. If the `archiver.cmd` file defines an archive set with a `-drives 2` directive, this file is archived to 2 drives in parallel. Segments 1, 3, 5, 7, and 9 are archived using the first drive, and segments 2, 4, 6, 8, and 10 are archived using the second drive.

Only segments that have been modified are archived—not the entire file. Up to four archive copies can be made for each segment. StorageTek ASM supports volume overflow for segments.

**Note:** The index of a segmented file contains no user data. It is considered metadata. It is assigned to the file system archive set.

## Disaster Recovery

For information about recovering a segmented file in the event of a disaster, see the *StorageTek ASM, ASM-QFS, and ASM/QFS-Standalone Disaster Recovery Guide*.

## ■ System Error Facility Reporting

The system error facility (SEF) reporting system captures log sense data from tape devices in an automated library, writes it to a log file, and translates that data into human-readable form. It consists of the following:

- A log file containing data from tape device log sense pages.
- A command, `sefreport(1M)`, for writing the log file to `stdout` in a human-readable format. This log file can be used as input to a user-supplied analysis script.

The `sefreport(1M)` command reads the content of a StorageTek ASM SEF log file. The log file contains data gathered from the log sense pages of the peripheral tape devices used in a StorageTek ASM environment. The log sense pages differ from vendor to vendor. For the meanings of the parameter codes, control bits, and parameter values, see the vendor documentation for each specific device.

SEF is not supported for standalone tape drives. SEF reporting is most useful for older SCSI-2 devices that do not support the `tapealert(1M)` functionality. For more information, see the `tapealert(1M)` man page.

## To Enable SEF Reporting

1. Become superuser.
2. Use the `mkdir(1)` command to create the SEF directory.

For example:

```
# mkdir /var/opt/SUNWsamfs/sef
```

3. Use the `touch(1)` command to enable SEF reporting.

You can enable SEF reporting any time after installation by creating the `sefdata` log file. Initially, the SEF log file must be empty.

```
# touch /var/opt/SUNWsamfs/sef/sefdata
```

The preceding example command shows the SEF log file being created in `/var/opt/SUNWsamfs/sef/sefdata`. This is the default location.

4. Use the `samd(1M) stop` and `samd(1M) start` to initialize SEF reporting.

For example:

```
# samd stop
# samd start
```

SEF data is appended to the log file as it is generated.

You can configure SEF reporting to log and read log sense data from an alternate location. For more information about reading log sense data from an alternate location, see the `sefreport(1M)` man page.

## SEF Report Output

Before you use the `sefreport(1M)` command, ensure that `/opt/SUNWsamfs/sbin` is in your command path. The SEF report output consists of header lines and log sense data.

Following the header lines, the log sense data for each page in the record is printed. For each log sense page, a line identifying the page code is printed, followed by a line of column headings. The data is then printed, three columns per line, with the following headings: `param code`, `control`, and `param value`. All data is generated in hexadecimal notation.

## To Generate SEF Output

- Use the `sefreport(1M)` command to generate SEF output.

The following are the most commonly used options on the `sefreport(1M)` command:

- The `-d` option. The `-d` option generates additional device information. It writes an additional header line that contains the Equipment Ordinal and path name to the device for each record. This makes it easier to search for and to locate SEF records that pertain to a specific device.
- The `-v` option or the `-t` option.

The `-v` option generates information in verbose mode. It appends information regarding the Equipment Ordinal, page code, and VSN to each line of a record. This makes it possible to select only those lines that pertain to a specific device or a specific volume.

The `-t` option generates log sense output with text descriptions. For each line of log sense data output, the report includes an additional string containing the Equipment Ordinal, page code, VSN, and parameter code description.

Do not specify the `-t` and `-v` options on the same command line. They are mutually exclusive.

For example, the following SEF command reads the SEF log file from the default location, writes the device number and path name for each device, and generates output:

```
# sefreport -d /var/opt/SUNWsamfs/sef/sefdata > sef.output
```

Figure 81 shows the content of `sef.output` file.

**Figure 81.** `sef.output` Contents

```
Record no. 1
Mon Mar 26 11:17:48 2001  STK      9840          1.25 VSN 002981
  Eq no. 32   Dev name /dev/rmt/1cbn

  PAGE CODE 2
  param code  control  param value
    00h       74h     0x0
    01h       74h     0x0
    02h       74h     0x0
    03h       74h     0x0
    04h       74h     0x0
    05h       74h     0x40050
    06h       74h     0x0

  PAGE CODE 3
  param code  control  param value
    00h       74h     0x0
    01h       74h     0x0
    02h       74h     0x0
    03h       74h     0x0
    04h       74h     0x0
    05h       74h     0x140
    06h       74h     0x0

  PAGE CODE 6
  param code  control  param value
    00h       74h     0x0

Record no. 2
Mon Mar 26 11:30:06 2001  STK      9840          1.25 VSN 002999
  Eq no. 31   Dev name /dev/rmt/0cbn

  PAGE CODE 2
  param code  control  param value
```

**Figure 81. sef.output Contents (Continued)**

00h	74h	0x0
01h	74h	0x0
02h	74h	0x0
03h	74h	0x0
04h	74h	0x0
05h	74h	0x1400a0
06h	74h	0x0
PAGE CODE 3		
param code	control	param value
00h	74h	0x0
01h	74h	0x0
02h	74h	0x0
03h	74h	0x0
04h	74h	0x0
05h	74h	0x190
06h	74h	0x0
PAGE CODE 6		
param code	control	param value
00h	74h	0x0
Record no. 3		
Mon Mar 26 11:30:23 2001	STK 9840	1.25 VSN 002981
Eq no. 32	Dev name /dev/rmt/1cbn	
PAGE CODE 2		
param code	control	param value
00h	74h	0x0
01h	74h	0x0
02h	74h	0x0
03h	74h	0x0
04h	74h	0x0
05h	74h	0x18400f0
06h	74h	0x0
PAGE CODE 3		
param code	control	param value
00h	74h	0x0
01h	74h	0x0
02h	74h	0x0
03h	74h	0x0
04h	74h	0x0
05h	74h	0x1e0
06h	74h	0x0
PAGE CODE 6		
param code	control	param value
00h	74h	0x0

**Figure 81.** `sef.output` Contents (Continued)

```
.  
. .  
. . .
```

**Note:** The preceding output has been truncated for inclusion in this manual.

For more information about the SEF log file, including its content and format, see the `sefdata(4)` man page. For more information about optional SEF report formats, see the `sefreport(1M)` man page.

## Managing the SEF Log File

You manage the SEF log file just as you manage any other StorageTek ASM log file. You can run a `cron(1)` job periodically to save the current log file to another location, to delete old SEF files, to create new (empty) SEF files, or to perform other tasks.

You can also use the `log_rotate.sh(1M)` utility to rotate this log file.

For more information about tools for managing the SEF log file, see the `cron(1)` or `log_rotate.sh(1M)` man pages.





# Glossary

---

## A

**addressable storage** The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through a StorageTek QFS or StorageTek ASM file system.

**archive media** The media to which an archive file is written. Archive media can be removable tape or magneto-optical cartridges in a library. In addition, archive media can be a mount point on another system.

**archive storage** Copies of file data that have been created on archive media.

**archiver** The archive program that automatically controls the copying of files to removable cartridges.

**audit (full)** The process of loading cartridges to verify their VSNs. For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog.

**automated library** A robotically controlled device designed to automatically load and unload removable media cartridges without operator intervention. An automated library contains one or more drives and a transport mechanism that moves cartridges to and from the storage slots and the drives.

## B

**backup storage** A snapshot of a collection of files for the purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.

**block allocation map** A bitmap representing each available block of storage on a disk and indicating whether the block is in use or free.

**block size** See DAU.

## C

**cartridge** A physical entity that contains media for recording data. A tape or optical disk. Sometimes referred to as *a piece of media*, *a volume*, or *the medium*.

**catalog** A record of the VSNs in an automated library. There is one catalog for each automated library, and at a site, there is one historian for all automated libraries.

**client-server** The model of interaction in a distributed system in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called the client. The program satisfying the response is called the server.

**connection** The path between two protocol modules that provides reliable stream delivery service. A TCP connection extends from a TCP module on one machine to a TCP module on the other.

## D

**data device** For a StorageTek QFS or StorageTek ASM file system, a device or group of devices upon which file data is stored.

**DAU** (Disk allocation unit) The basic unit of online storage. Also called block size.

In addition, the StorageTek QFS file systems support a fully adjustable DAU, sized from 16 kilobytes through 65,528 kilobytes. The DAU you specify must be a multiple of 8 kilobytes.

The StorageTek ASM file systems support both a small and a large DAU. The small DAU is 4 kilobytes ( $2^{14}$  or 4096 bytes). The large DAU is 16, 32, or 64 kilobytes. The available DAU size pairs are 4 and 16; 4 and 32; and 4 and 64.

**device logging** A configurable feature that provides device-specific error information used to analyze device problems.

**device scanner** Software within the StorageTek ASM file system that periodically monitors the presence of all manually mounted removable devices and that detects the presence of mounted cartridges that can be requested by a user or other process.

**direct access** A file attribute (stage never) designating that a nearline file can be accessed directly from the archive media and need not be retrieved to disk cache.

**direct-attached library** An automated library connected directly to a server using a SCSI interface. A SCSI attached library is controlled directly by the StorageTek ASM software by using the SCSI standard for automated libraries.

**direct I/O** An attribute used for large block-aligned sequential I/O. The `setfa(1)` command's `-D` option is the direct I/O option. It sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.

**directory** A file data structure that points to other files and directories within the file system.

**disk allocation unit** See DAU.

**disk buffer** When using StorageTek ASM-Remote software, the disk buffer is a buffer on the server system that is used when archiving data from the client to the server.

**disk cache** The disk-resident portion of the StorageTek ASM file system software. It is used to create and manage data files between online disk cache and archive media. Individual disk partitions or an entire disk can be used as disk cache.

**disk space thresholds** An administrator-defined amount of disk space that is available to a user. This defines the range of desirable disk cache utilization. The high threshold indicates the maximum level of disk cache utilization. The low threshold indicates the minimum level of disk cache utilization. The releaser controls disk cache utilization based on these predefined disk space thresholds.

**disk striping** The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity. See also entries for striping.

**drive** A mechanism for transferring data to and from a removable media volume.

## E

**Ethernet** A local-area, packet-switched network technology. Originally designed for coaxial cable, it is now found running over shielded, twisted-pair cable. Ethernet is a 10- or 100-Mbytes/sec LAN.

**extent array** The array within a file's inode that defines where each data block assigned to the file is located on the disk.

## F

**family device set** See Family Set.

**Family Set** A storage device that is represented by a group of independent physical devices, such as a collection of disks or the drives within an automated library. Also see storage Family Set.

**FDDI** (Fiber distributed data interface) A 100-Mbytes/sec fiber-optic LAN.

**ibre channel** The ANSI standard that specifies high-speed serial communication between devices. Fibre channel is used as one of the bus architectures in SCSI-3.

**iber-distributed data interface** See FDDI.

**file system** A hierarchical collection of files and directories.

**file system specific directives** Archiver and releaser directives that follow global directives, are specific to a particular file system, and begin with `fs =`. File system specific directives apply until the next `fs =` directive line or until the end of file is encountered. If multiple directives affect a file system, the file system-specific directives override the global directives.

**FTP** (File transfer protocol) An internet protocol for transferring files between two hosts over a TCP/IP network.

## G

**global directives** Archiver and releaser directives that apply to all file systems and that appear before the first `fs =` line.

**grace period** For disk quotas, this is the amount of time that can elapse during which a user is allowed to create files and allocate storage after the user reaches their soft limit.

## H

**hard limit** For disk quotas, a maximum limit on file system resources, blocks and inodes, that users cannot exceed.

## I

**indirect block** A disk block that contains a list of storage blocks. The StorageTek QFS and StorageTek ASM file systems have up to three levels of indirect blocks. A first-level indirect block contains a list of blocks used for data storage. A second-level indirect block contains a list of first-level indirect blocks. A third-level indirect block contains a list of second-level indirect blocks.

**inode** Index node. A data structure used by the file system to describe a file. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permission, size, and the file location on the disk system.

**inode file** A special file (`.inodes`) on the file system that contains the inode structures for all files resident in the file system. All StorageTek QFS and StorageTek ASM inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the StorageTek QFS file systems.

## K

**kernel** The central controlling program that provides basic system facilities. The UNIX kernel creates and manages processes, provides functions to access the file system, provides general security, and supplies communication facilities.

## L

**LAN** Local area network.

**lease** In a StorageTek QFS shared file system, a lease grants a client host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each client host. The leases are renewed as necessary to permit continued file operations.

**library** See automated library.

**library catalog** See catalog.

**local file system** A file system that is installed on one node of a Sun Cluster and is not made highly available to another node. Also a file system that is installed on a standalone server.

**LUN** Logical unit number.

## M

**mcf** Master configuration file. The file that is read at initialization time that defines the relationships between the devices (the topology) within a StorageTek QFS or StorageTek ASM environment.

**media** Tape or optical disk cartridges.

**media recycling** The process of recycling or reusing archive media with low use. Archive media with low use is archive media with few active files.

**metadata** Data about data. Metadata is the index information needed to locate the exact data position of a file on a disk. It consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files. Metadata must be protected because if data is lost, the metadata that locates the data must be restored before the lost data can be retrieved.

**metadata device** A separate device (for example, a solid-state disk or mirrored device) upon which StorageTek QFS file system metadata is stored. Separating the file data from the metadata can increase performance. In the `mcf` file, a metadata device is declared as an `mm` device within an `ma` file system.

**mirror writing** The process of maintaining two copies of a file on disjointed sets of disks to prevent loss from a single disk failure.

**mount point** The directory on which a file system is mounted.

**multireader file system** The StorageTek QFS multireader file system is a single-writer, multireader capability that enables you to specify a file system that can be mounted on multiple hosts. Multiple hosts can read the file system, but only one host can write to the file system. Multiple readers are specified with the `-o reader` option on the `mount(1M)` command. The single-writer host is specified with the `-o writer` option on the `mount(1M)` command. For more information on the `mount(1M)` command, see the `mount_samfs(1M)` man page.

## N

**name space** The metadata portion of a collection of files that identifies the file, its attributes, and its storage locations.

**nearline storage** Removable media storage that requires robotic mounting before it can be accessed. Nearline storage is usually less expensive than online storage, but it incurs a somewhat longer access time.

**network-attached automated library** A library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, that is controlled using a software package supplied by the vendor. The StorageTek ASM file system interfaces with the vendor software using a StorageTek ASM media changer daemon designed specifically for the automated library.

**NFS** Network file system. A Sun distributed file system that provides transparent access to remote file systems on heterogeneous networks.

**NIS** The SunOS 4.0 (minimum) Network Information Service. A distributed network database containing key information about the systems and the users on the network.

The NIS database is stored on the master server and all the slave servers.

## O

**offline storage** Storage that requires operator intervention for loading.

**offsite storage** Storage that is remote from the server and is used for disaster recovery.

**online storage** Storage that is immediately available (for example, disk cache storage).

## P

**partition** A portion of a device or a side of a magneto-optical cartridge.

**preallocation** The process of reserving a contiguous amount of space on the disk cache for writing a file. This ensures that the space is contiguous. Preallocation can be performed only on zero-sized files. That is, the `setfa -l` command can be specified only for a file that is size zero. For more information, see the `setfa(1)` man page.

**prioritizing preview requests** Assigning priority to archive and stage requests that cannot be immediately satisfied.

**pseudo device** A software subsystem or driver with no associated hardware.

## Q

**quota** The amount of system resources that a user is allowed to consume.

## R

**RAID** Redundant array of independent disks. A disk technology that uses several independent disks to reliably store files. It can protect against data loss from a single disk failure, can provide a fault-tolerant disk environment, and can provide higher throughput than individual disks.

**recycler** A StorageTek ASM utility that reclaims space on cartridges that is occupied by expired archive copies.

**release priority** A method of calculating the release priority of a file within a file system by multiplying various weights by the corresponding file properties and then summing the results.

**releaser** A StorageTek ASM component that identifies archived files and releases their disk cache copies, thus making more disk cache space available. The releaser automatically regulates the amount of online disk storage to high and low thresholds.

**remote procedure calls** See RPC.

**removable media file** A special type of user file that can be accessed directly from where it resides on a removable media cartridge, such as magnetic tape or optical disk cartridge. Also used for writing archive and stage file data.

**robot** The portion of an automated library that moves cartridges between storage slots and drives. Also called a transport.

**round robin** A data access method in which entire files are written to logical disks in a sequential fashion. When a single file is written to disk, the entire file is written to the first logical disk. The second file is written to the next logical disk, and so on. The size of each file determines the size of the I/O.

By default, StorageTek QFS and StorageTek ASM file systems implement striped data access unless striped groups are present. Files are round-robin if round robin access is specified. If the file system contains mismatched striped groups, striping is not supported and round robin is forced.

Also see glossary entries for disk striping and striping.

**RPC** Remote procedure calls. The underlying data exchange mechanism used

by NFS to implement custom network data servers.

## S

**samfsdump** A program that creates a control structure dump and copies all the control structure information for a given group of files. It is analogous to the UNIX `tar(1)` utility, but it does not generally copy file data.

**samfsrestore** A program that restores inode and directory information from a control structure dump.

**SCSI** Small Computer System Interface. An electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.

**small computer system interface** See SCSI.

**soft limit** For disk quotas, a threshold limit on file system resources (blocks and inodes) that you can temporarily exceed. Exceeding the soft limit starts a timer. When you exceed the soft limit for the specified time (default is one week), no further system resources can be allocated until you reduce file system use to a level below the soft limit.

**staging** The process of copying a nearline or offline file from archive storage back to online storage.

**storage Family Set** A set of disks that are collectively represented by a single disk family device.

**storage slots** Locations inside an automated library in which cartridges are stored when not being used in a drive. If the library is direct-attached, the contents of the storage slots are kept in the automated library's catalog.

**stripe size** The number of disk allocation units (DAUs) to allocate before moving to

the next device of a stripe. If `stripe=0`, the file system uses round-robin access, not striped access.

**striped group** A collection of devices within a StorageTek QFS file system and defined in the `mcf` file as one (usually two) or more `gXXX` devices. Striped groups are treated as one logical device and are always striped with a size equal to the disk allocation unit (DAU). You can specify up to 128 striped groups within a file system, but you can specify no more than 252 total devices.

**striping** A data access method in which files are simultaneously written to logical disks in an interlaced fashion. All StorageTek QFS and StorageTek ASM file systems enable you to declare either striped or round robin access for each individual file system. The StorageTek QFS file systems enable you to declare striped groups within each file system. Also see the glossary entry for round robin.

**StorageTek QFS** A high-speed UNIX file system that separates the file system metadata from the file data by storing them on separate devices. The StorageTek QFS software controls the access to all files stored and all devices configured in the master configuration file (`mcf`).

**StorageTek ASM** The StorageTek ASM Storage and Archive Manager File System. The StorageTek ASM software controls the access to all files stored and all devices configured in the master configuration file (`mcf`).

**StorageTek ASM-QFS** The StorageTek ASM-QFS software combines the StorageTek ASM software with the StorageTek QFS file system. StorageTek ASM-QFS offers a high-speed, standard UNIX file system interface to users and administrators in conjunction with the storage and archive management utilities. It uses many of the commands available in the

StorageTek ASM command set as well as standard UNIX file system commands.

**StorageTek ASM-Remote client** A StorageTek ASM-Remote client is a StorageTek ASM system that establishes a StorageTek ASM-Remote client daemon that contains a number of pseudodevices. It might or might not have its own library devices. The client depends on a StorageTek ASM-Remote server for archive media for one or more archive copies.

**StorageTek ASM-Remote server** The StorageTek ASM-Remote server is both a full-capacity StorageTek ASM storage management server and a StorageTek ASM-Remote server daemon that defines libraries to be shared among StorageTek ASM-Remote clients.

**superblock** A data structure in the file system that defines the basic parameters of the file system. It is written to all partitions in the storage Family Set and identifies the partition's membership in the set.

## T

**tar** Tape archive. A standard file and data recording format used by the StorageTek ASM software for archive images.

**TCP/IP** Transmission Control Protocol/Internet Protocol. The internet protocols responsible for host-to-host addressing and routing, packet delivery (IP), and reliable delivery of data between application points (TCP).

**thresholds** A mechanism for defining the desirable available storage window for online storage. Thresholds set the storage goals for the releaser. Also see disk space thresholds.

**timer** Quota software that keeps track of the time elapsed between a user reaching a soft limit and a hard limit being imposed on the user.

## V

**volume** A named area on a cartridge for sharing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side.

**volume overflow** A capability that enables the system to span a single file over multiple volumes. Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.

**VSN** Volume serial name. If you are archiving to removable media cartridges, the VSN is a logical identifier for magnetic tape and optical disk that is written in the volume label. If you are archiving to disk cache, this is the unique name for the disk archive set.

## W

**WORM** Write once read many. A storage classification for media that can be written only once but read many times.





# Index

---

## Symbols

`/etc/opt/SUNWsamfs/archiver.cmd`, see `archiver.cmd` file  
`/etc/opt/SUNWsamfs/diskvols.conf`, see `diskvols.conf` file  
`/etc/opt/SUNWsamfs/preview.cmd`, see `preview.cmd` file  
`/etc/opt/SUNWsamfs/recycler.cmd`, see `recycler.cmd` file  
`/etc/opt/SUNWsamfs/scripts/archiver.sh`, see `archiver.sh` script  
`/opt/SUNWsamfs/examples/recycler.sh`, see `recycler.sh` file  
`/var/opt/SUNWsamfs/devlog` file, see `devlog` file  
`/var/opt/SUNWsamfs/sef/sefdata`, see `sefdata` file

## A

`-access` archiver directive, 74  
ACSAPI interface, 4  
Adding slots, 168  
ADIC/Grau automated library  
    operations, 35  
`age_priority` `preview.cmd` directive, 147  
`allsets` archive set, 48, 82  
API routines, 12  
Application programmer interface, see API  
`archive(1)` command, 5  
`archive_audit(1M)` command, 10  
Archiver  
    archive age definition, 48  
    archive interval definition, 48  
    archive media definition, 1, 47  
    archive set membership conflicts, 78  
    archive set membership directive, 72  
    archive set processing directives, 82  
    archive sets, 48  
    `archiver.cmd` examples, 61  
    associative archiving, 87  
    commands, 10  
    continuous archiving, 66  
    controlling file size, 64  
    copy creation, 2  
    copy definition directives, 79  
    daemons, 56  
    defaults, 48  
    defined, 1, 47  
    deleting entries, 11  
    determining the archive age, 89  
    directives, also see Directives, 60, 62  
    disk archiving, see Disk archiving  
    examples, 105  
    guidelines, 116  
    logfiles, 57, 117  
    overview, 2  
    preventing archiving, 73  
    reserved VSNs, 91  
    role in staging, 145  
    setting automatic unarchiving, 81  
    setting priorities, 93  
    setting the archive age, 81  
    specifying a file system in  
        `archiver.cmd`, 71  
    specifying copies of file system data, 81  
    theory of operations, 47  
    troubleshooting, 117  
    unarchiving, 89  
    using regular expressions, 75  
    VSN association directives, 96  
    VSN pools, 98  
`archiver(1M)` command, 10, 59  
`archiver.cmd` file, 47, 56, 59, 100, 116, 161  
`archiver.sh(1M)` command, 10  
`archiver.sh(1M)` script, 69  
`archmax` archiver directive, 64  
ASM QFS Manager, 12  
Associative archiving, 87  
`attended` directive, 29  
Audit  
    automated library, 21

volume, 21  
 auditslot(1M) command, 9, 21  
 Automated library  
   also see entries for individual vendors, 4  
   auditing, 21  
   commands, 9  
   daemons, 4  
   defined, 13  
   direct-attached, 3, 28  
   historian  
     see Library historian, 29  
   IBM, 4  
   network-attached, 3, 28  
   operations, 28  
   SCSI attached, see Automated library, di-  
     rect attached  
   Sony, 4  
   StorageTek, 4  
   turning off, 17  
   turning on, 17

## B

-bufsize archiver directive, 83  
 bufsize archiver directive, 64  
 bufsize stager directive, 141  
 build\_cat(1M) command, 9

## C

Capacity, defined, 153  
 Cartridges  
   cleaning, 22  
   exporting, 31  
   importing, 30  
   loading, 9, 18, 33  
   removing, 26  
   unloading, 9, 18, 33  
 Cataloging opticals, 10  
 Chapter 1, 1  
 checksum attribute, 6  
 chmed(1M) command, 9, 25, 160  
 cleandrive(1M) command, 9, 25  
 Cleaning tape drives, 24  
 Commands  
   archive(1), 5  
   archive\_audit(1M), 10  
   archiver(1M), 10, 59

archiver.sh(1M), 10  
 auditslot(1M), 9, 21  
 automated library, 9  
 build\_cat(1M), 9  
 chmed(1M), 9, 25, 160  
 cleandrive(1M), 9, 25  
 dev\_down.sh(1M), 11  
 dmpshm(1M), 10  
 du(1), 5  
 dump\_cat(1M), 9  
 exarchive(1M), 10  
 file system, 7  
 find(1), also see sfind(1) command, 6  
 general system administrator, 6  
 GUI, see ASM QFS Manager  
 import(1M), 9, 23, 28, 30, 36, 37, 41, 43,  
   45  
 itemize(1M), 10  
 load\_notify.sh(1M), 11  
 log\_rotate.sh(1M), 11  
 ls(1), also see sls(1) command, 6  
 makedev(1M), see samdev(1M)  
 mount(1M), 7  
 move(1M), 9, 41, 42  
 odlabel(1M), 9, 20  
 qfsdump(1M), 7  
 qfsrestore(1M), 7  
 rearch(1M), 10  
 recover.sh(1M), 11  
 release(1), 5, 121  
 request(1), 5, 158, 178  
 reserve(1M), 10  
 restore.sh(1M), 12  
 sambcheck(1M), 7  
 samchaid(1M), 7  
 samcmd(1M), 6, 16  
 samd(1M), 6, 16  
 samdev(1M), 9, 10  
 samexplorer(1M), 7  
 samexport(1M), 9, 28, 31, 36, 38, 42, 44,  
   46  
 samfsck(1M), 7  
 samfsconfig(1M), 8  
 samfsdump(1M), 8  
 samfsinfo(1M), 8  
 samfsrestore(1M), 8  
 samgrowfs(1M), 8  
 samload(1M), 9

sammkfs(1M), 8  
 samncheck(1M), 8  
 samquota(1M), 8  
 samquotastat(1M), 8  
 sam-recycler(1M), 10, 162  
 sam-releaser(1M), 10, 121  
 samset(1M), 7, 10, 177  
 samsharefs(1M), 8  
 samtrace(1M), 8  
 samu(1M), 7, 12  
 samunhold(1M), 8  
 sdu(1), 5  
 sefreport(1M), 181  
 segment(1), 5, 180  
 set\_admin(1M), 11  
 set\_state(1M), 11  
 setfa(1), 5  
 sfind(1), 6  
 showqueue(1M), 10  
 sls(1), 6  
 squota(1), 6  
 ssum(1), 6  
 stage(1), 6  
 stageback.sh(1M), 11, 12  
 star(1M), 11  
 tar(1), see star(1M) command  
 tarback.sh(1M), 12  
 tplabel(1M), 9, 19  
 trace\_rotate(1M), 8  
 unarchive(1M), 11  
 undamage(1M), 11  
 unload(1M), 9  
 unrearch(1M), 10  
 unreserve(1M), 10  
 user, 5  
 Continuous archiving, 66  
 crontab entry, 163  
 Current data, defined, 153  
 Customer Resource Center (CRC), xviii

## D

### Daemons

automated library daemons, 4  
 sam-archiverd, 56  
 sam-arcopy, 56  
 sam-arfind, 56  
 sam-fsd, 56

sam-genericd, 4  
 sam-ibm3494d, 4  
 sam-initd, 4  
 sam-robotsd, 4  
 sam-rpcd, 12  
 sam-sonyd, 4  
 sam-stkd, 4

### Defaults

archiver, 48  
 setting system defaults, 10

defaults.conf file, 177

dev\_down.sh(1M) command, 11

### Device

create StorageTek ASM, 10  
 links, 10  
 logging, see Logfiles  
 set state of, 11  
 states, 17

### devlog

file, 175

Direct-attached automated library, see Automated library

### Directives

#### archiver

-access, 74  
 archive set copy number, 79  
 archive set membership, 72  
 archmax, 64  
 -bufsize, 83  
 bufsize, 64  
 -disk\_archive, 99, 101  
 -drivemin, 84  
 -drives, 83  
 drives, 65  
 endparams, 82  
 endvsnpools, 98  
 endvsns, 96  
 examine, 66  
 fs, 71  
 -group, 75  
 interval, 67  
 -join path, 87  
 -lock, 86  
 logfile, 68  
 -maxsize, 74  
 -minsize, 74  
 -name, 75  
 -norelease, 80

- notify, 68
  - overview, 60, 62
  - ovflmin, 69
  - params, 82
  - pool, 96
  - priority, 93
  - recycle\_dataquantity directive, 161
  - recycle\_hwm directive, 161
  - recycle\_ignore directive, 161, 164
  - recycle\_mailaddr directive, 161
  - recycle\_mingain directive, 161
  - recycle\_vsncount directive, 161
  - recycling directives, 87
  - release, 77, 80
  - reserve, 90
  - setting automatic unarchiving, 81
  - setting the archive age, 81
  - sort, 87
  - specifying copies of file system
    - data, 81
  - stage, 77
  - startage, 94
  - startcount, 94
  - startsize, 94
  - tapenonstop, 90
  - user, 75
  - vsnpools, 98
  - vsns, 96
  - wait, 71
  - previewer
    - age\_priority, 147
    - hlwm\_priority, 148
    - hwm\_priority, 148
    - lhwm\_priority, 148
    - lwm\_priority, 148
    - vsn\_priority, 147
  - recycler
    - ignore directive, 164
    - logfile, 155
    - no\_recycle, 156
  - releaser
    - display\_all\_candidates, 132
    - fs, 132
    - list\_size, 135
    - logfile, 133
    - min\_residence\_age, 133
    - no\_release, 132
    - rearch\_no\_release, 134
    - release priority directives, 129
    - weight\_age, 130
    - weight\_age\_access, 130
    - weight\_age\_modify, 130
    - weight\_age\_residence, 130
    - weight\_size, 131
  - stager
    - bufsize, 141
    - drives, 141
    - logfile, 142
    - maxactive, 144
  - disclaimer, ii
  - Disk archiving, 99, 154
  - Disk cache
    - high watermark, 122
    - low watermark, 122
    - release priority, 2
    - thresholds, 2
  - disk\_archive archiver directive, 101
  - diskvols.conf file, 100
  - display\_all\_candidates releaser
    - directive, 132
  - DLT tape drives, 172
  - dmpshm(1M) command, 10
  - drivemin archiver directive, 84
  - drives archiver directive, 65
  - drives archiver set parameter directive, 83
  - drives stager directive, 141
  - Drives, cleaning, 22, 24
  - du(1) command, 5
  - dump\_cat(1M) command, 9
  - DZC-8000S interface, 4
- ## E
- endparams archiver directive, 82
  - endvsnpools archiver directive, 98
  - endvsns archiver directive, 96
  - e-Partners Web site, xviii
  - Error processing of stage requests, 2
  - examine archiver directive, 66
  - exarchive(1M) command, 10
  - Expired data, defined, 153
  - exported\_media directive, 29
  - Exporting media overview, 28

## F

File system  
  commands, 7  
  data, 81  
  overview, 1

Files, setting attributes, 77

find(1) command, also see sfind(1)  
command, 6

Free space, defined, 153

fs archiver directive, 71

fs releaser directive, 132

Fujitsu LMF automated library  
operations, 37

## G

General system administrator commands, 6

-group archiver directive, 75

GUI tools, see ASM QFS Manager

## H

hardcopy publications, xviii

Hardware upgrades, 167

High water mark

  previewer directives, 147

  with recycler, 159

hlwm\_priority previewer directive, 148

hwm\_priority previewer directive, 148

## I

IBM 3494 Automated Library  
  overview, 39

IBM 3584 Automated Library  
  cleaning, 38  
  importing, 38  
  overview, 38  
  partitioning, 39

IBM automated library, 4

ignore recycler directive, 164

import(1M) command, 9, 23, 28, 30, 36, 37,  
41, 43, 45

Importing media overview, 28

interval archiver directive, 67

itemize(1M) command, 10

## J

-join path archiver directive, 87

Jukebox, see Automated library

## L

lhwm\_priority previewer directive, 148

liability, ii

Library catalog  
  overview, 28  
  viewing, 34

Library historian, 29

Library, see Automated library

libsam, 12

libsamrpc, 12

Licensing

  adding slots to an automated library, 168

list\_size releaser directive, 135

lmcpc interface, 4

load\_notify.sh(1M) command, 11

Loading media

  manually loaded drive, 33

  with samu(1M), 18

-lock archiver directive, 86

log\_rotate.sh(1M) command, 11

logfile

  archiver directive, 68

  recycler directive, 155

  releaser directive, 133

  stager directive, 142

Logfiles

  archiver, 57, 68, 117

  device logging, 175

  recycler, 162

  releaser, 133

  stager, 142

Low watermark directives, 147

ls(1) command, see sls(1) command

lwm\_priority previewer directive, 148

## M

makedev(1M) command, see samdev(1M)  
command

Master configuration file

  see mcf file

maxactive stager directive, 144

-maxsize archiver directive, 74  
 mcf file  
   library historian, 29  
   purpose, 4  
 Media  
   changer, see Automated library  
   errors, 25  
   library, see Automated library  
   loading, 18  
   moving, 28  
   types, 14  
 min\_residence\_age releaser directive, 133  
 -minsize archiver directive, 74  
 mount(1M) command, 7  
 move(1M) command, 9, 41, 42

## N

-name archiver directive, 75  
 no\_archive archive set, 48, 73  
 no\_recycle recycler directive, 156  
 no\_release releaser directive, 132  
 -norelease archiver directive, 80  
 notify archiver directive, 68

## O

odlabel(1M) command, 9, 20  
 ovflmin archiver directive, 69

## P

params archiver directive, 82  
 Partial release and partial stage feature, 124  
 -pool archiver directive, 96  
 Preview requests  
   age\_priority directive, 147  
   calculating priority, 149  
   hlwm\_priority directive, 148  
   hwm\_priority directive, 148  
   initialization, 149  
   lhwm\_priority directive, 148  
   lwm\_priority directive, 148  
   planning, 149  
   prioritizing, 146  
   vsn\_priority directive, 147  
 preview.cmd file, also see Preview

requests, 146, 150  
 -priority archiver directive, 93  
 Provides, 12

## Q

qfsdump(1M) command, 7  
 qfsrestore(1M) command, 7

## R

research(1M) command, 10  
 research\_no\_release releaser directive, 134  
 Rearchiving, defined, 154  
 recover.sh(1M) command, 11  
 -recycle\_dataquantity archiver directive, 161  
 -recycle\_hwm archiver directive, 161  
 -recycle\_ignore archiver directive, 161, 164  
 -recycle\_mailaddr archiver directive, 161  
 -recycle\_mingain archiver directive, 161  
 -recycle\_vsncount archiver directive, 161

### Recycler

  configuration, 157  
   crontab entry, 163  
   defined, 2  
   directives, 155  
   disk archive copies, 154  
   forcing with the chmed(1M)  
     command, 160  
   ignore option, 160  
   logfile directive, 155  
   mail option, 160  
   no\_recycle directive, 156  
   overview, 3, 153  
   recycler.cmd example, 159  
   recycler.sh file, 164  
   theory of operations, 155  
   troubleshooting, 164  
 recycler.cmd file, 159  
 recycler.sh file, 164  
 Regular expressions, 75  
 -release archiver directive, 77, 80  
 release(1) command, 5, 121  
 Releaser  
   age, 123  
   candidate, 123  
   command file, 128  
   configuration, 136

- defined, 1
- directives, 128
- fs directive, 132
- logfile, 133
- manual operation, 137
- overview, 2, 121, 122
- partial release, 124
- priority, 124
- troubleshooting, 120, 137
- weight, 124
- releaser.cmd file, 128, 136
- Removable media files, 178
- Removing damaged files, 11
- Request files, see removable media files
- request(1) command, 5, 158, 178
- reserve archiver directive, 90
- reserve(1M) command, 10
- restore.sh(1M) command, 12
- Retrieving data
  - also see Disaster recovery, 71
- Robot, see Automated library

## S

- sam\_segment(3), 180
- sam-archiverd daemon, 56
- sam-arcopy daemon, 56
- sam-arfind daemon, 56
- sambcheck(1M) command, 7
- samchaid(1M) command, 7
- samcmd(1M) command, 6, 16
- samd(1M) command, 6, 16
- samdev(1M) command, 9, 10
- samexplorer(1M) command, 7
- samexport(1M) command, 9, 28, 31, 36, 38, 42, 44, 46
- samfsck(1M) command, 7
- samfsconfig(1M) command, 8
- sam-fsd daemon, 56
- samfsdump(1M) command, 8
- samfsinfo(1M) command, 8
- samfsrestore(1M) command, 8
- sam-genericd daemon, 4
- samgrowfs(1M) command, 8
- sam-ibm3494d daemon, 4
- sam-initd daemon, 4
- samload(1M) command, 9
- sammkfs(1M) command, 8
- samncheck(1M) command, 8
- samquota(1M) command, 8
- samquotastat(1M) command, 8
- sam-recycler(1M) command, 10, 162
- sam-releaser(1M) command, 10, 121
- sam-robotsd daemon, 4
- sam-rpcd daemon, 12
- samset(1M) command, 7, 10, 177
- samsharefs(1M) command, 8
- sam-sonyd daemon, 4
- sam-stkd daemon, 4
- samtrace(1M) command, 8
- samu(1M) invocation command, 7, 12
- samunhold(1M) command, 8
- sdu(1) command, 5
- SEF, 181
- sefdata file, 182
- sefreport(1M) command, 181
- segment(1) command, 5, 180
- Segmented files, 180
- set\_admin(1M) command, 11
- set\_state(1M) command, 11
- setfa(1) command, 5
- sfind(1) command, 6
- showqueue(1M) command, 10
- Slots, adding, 168
- sls(1) command, 6
- Small computer system interface
- Software upgrades, 167
- Sony automated library, 4
- Sony network-attached automated library
  - operations, 43
- Sony PetaSite automated library
  - operations, 40
  - overview, 40
- sort archiver directive, 87
- squota(1) command, 6
- ssum(1) command, 6
- stage archiver directive, 77
- stage(1) command, 6
- stageback.sh(1M) command, 11, 12
- Stager
  - archive role in staging, 145
  - bufsize directive, 141
  - defined, 2, 139
  - directives, 139
  - drives directive, 141
  - error processing of stage requests, 2

- logfile fields, 144
- logging activities, 142
- maxactive directive, 144
- overview, 2, 139
- partial stage, 124
- star(1M) command, 11
- startage archiver directive, 94
- startcount archiver directive, 94
- Starting StorageTek ASM, 16
- startsize archiver directive, 94
- Stopping StorageTek QFS, 16
- StorageTek
  - Customer Resource Center (CRC), xviii
  - e-Partners site, xviii
  - hardcopy publications, xviii
  - Web site, xviii
- StorageTek ACSLS-attached automated library
  - operations, 45
- StorageTek ASM, stopping, 16
- StorageTek ASM-QFS
  - see StorageTek ASM
- StorageTek automated library, 4
- System error facility, see SEF

## T

- tapenonstop archiver directive, 90
- tar(1) command, see star(1M)
- tarback.sh(1M) command, 12
- tplabel(1M) command, 9, 19
- trace\_rotate(1M) command, 8
- Troubleshooting
  - archiver, 117
  - recycler, 164
  - releaser, 137

## U

- unarchive(1M) command, 11
- Unarchiving, 81, 89
- undamage(1M) command, 11
- unload(1M) command, 9
- Unloading media with samu(1M), 18
- unreach(1M) command, 10
- unreserve(1M) command, 10
- Upgrading hardware and software, 167
- user archiver directive, 75

- User commands, 5

## V

- Volume overflow (ovflmin archiver directive), 69
- Volume Serial Name, see VSN
- VSN
  - association directives, 96
  - disk archiving directives, 99
  - minimum gain per VSN, 160
  - pool directives, 98
  - reserving, 90
  - using regular expressions, 97
- vsn\_priority preview.cmd directive, 147
- vsnpools archiver directive, 98
- vsns archiver directive, 96

## W

- wait archiver directive, 71
- warranties, ii
- Watermark directives, 147
- weight\_age releaser directive, 130
- weight\_age\_access releaser directive, 130
- weight\_age\_modify releaser directive, 130
- weight\_age\_residence releaser directive, 130
- weight\_size releaser directive, 131
- wm\_priority factor, 147







NEED MORE INFORMATION?  
[www.storagetek.com](http://www.storagetek.com)

## ABOUT STORAGETEK

Storage Technology Corporation (NYSE: STK) is a \$2 billion global company that enables businesses, through its information lifecycle management strategy, to align the cost of storage with the value of information. The company's innovative storage solutions manage the complexity and growth of information, lower costs, improve efficiency and protect investments. For more information, visit [www.storagetek.com](http://www.storagetek.com), or call 1.800.275.4785 or 01.303.673.2800.

## WORLD HEADQUARTERS

Storage Technology Corporation  
One StorageTek Drive  
Louisville, Colorado 80028 USA  
1.800.678.4430 or 01.303.673.4430

© 2004 Storage Technology Corporation, Louisville, CO. All rights reserved. Printed in USA. StorageTek and the StorageTek logo are registered trademarks of Storage Technology Corporation. Other names mentioned may be trademarks of Storage Technology Corporation or other vendors/manufacturers.

StorageTek equipment is manufactured from new parts, or new and used parts. In some cases, StorageTek equipment may not be new and may have been previously installed. Regardless, StorageTek's standard warranty terms apply, unless the equipment is specifically identified by StorageTek as "used" or "refurbished."

Replacement parts provided under warranty or any service offering may be either new or equivalent-to-new, at StorageTek's option. Specifications/features may change without notice.