

PRIMECLUSTER™

Global File Services

Configuration and Administration Guide 4.1

(Solaris™ Operating Environment)

Edition February 2003

Preface

The Global File Services File System and its optional products are generically called "GFS product" or "GFS" in the document.

This manual explains the functions, settings, and operations of the entire GFS products. For the available functions, refer to the installation guide of each GFS product.

Target Readers

This manual is intended for all the users operating the products with GFS File System (e.g. PRIMECLUSTER.)

To read this manual, readers will require a general knowledge of the UNIX and Solaris(TM) Operating Environment. (From here, it will be abbreviated as "Solaris OE.").

Because GFS File System uses the functions of PRIMECLUSTER, readers will also need a knowledge of cluster control as used in a PRIMECLUSTER system.

Organization

This manual consists of the following chapters and appendixes:

Because GFS File System uses the functions of PRIMECLUSTER, readers will also need knowledge of cluster control as used in a PRIMECLUSTER system.

Part 1 Overview of Global File Services

Chapter 1 Overview of GFS

This chapter explains overview of Global File Services file system.

Part 2 GFS Local File System

Chapter 2 GFS Local File System

This chapter explains the functions and facilities of the GFS Local File System.

Chapter 3 Structure of the GFS Local File System

This chapter explains the structure of the GFS Local File System.

Chapter 4 Management of the GFS Local File System

This chapter explains the procedure for managing the GFS Local File System and the basic commands that are used for management.

Chapter 5 Backing Up and Restoring the GFS Local File System

This chapter explains how to back up and restore the GFS Local File System.

Chapter 6 Commands in the GFS Local File System

This chapter describes commands for use by general users, which are provided by the GFS Local File System.

Chapter 7 Managing Restriction of GFS Local File System Usage

This chapter explains the quota functions used to limit file systems usage.

Chapter 8 Defragmenting GFS Local File System

This chapter describes the defragmentation function provided by the GFS Local File System.

Chapter 9 Performance and Tuning

This chapter explains how to optimize the GFS Local File System and how to use it effectively with various utilities.

-
- Chapter 10 Using GFS Local File System in RMS cluster Operation
This chapter explains how to use GFS Local File System in RMS cluster operation.
- Chapter 11 Changing Node information for GFS Local File System
This chapter explains the action to be taken when node name or host ID of a node using GFS Local File System changed.
- Appendix A Kernel Messages
This appendix explains the kernel messages for the GFS Local File System.
- Appendix B Reference Manual
This appendix contains the reference manual for the GFS Local File System.
- Appendix C Corrective Measures against I/O Errors in the GFS Local File System
This appendix explains corrective measures that are to be taken if an I/O error occurred in the GFS Local File System.
- Appendix D Command correspondence with GFS Local File System and UFS file system
This appendix explains the command correspondence with GFS Local File System and UFS file system (the Solaris OE standard).
- Part 3 GFS Shared File System
- Chapter 12 File System Functions
This chapter describes the functions and features of the GFS Shared File System.
- Chapter 13 File System Structure
This chapter describes the structure of the GFS Shared File System.
- Chapter 14 Failure Recovery
This chapter describes the following failure recovery functions of the GFS Shared File System.
- Chapter 15 File System Design
This chapter describes the main features of the operational design of the GFS Shared File System.
- Chapter 16 Management Partition
This chapter explains the management partition.
- Chapter 17 Starting and Exiting the Management View
This chapter explains how to start and exit the GFS Management view.
- Chapter 18 Operation Management View Screen Elements
This chapter describes the screen elements of the GFS Management view.
- Chapter 19 File System Operations (GUI)
This chapter explains the operations of the GFS Management View.
- Chapter 20 File System Operations (Command)
This chapter describes the GFS Shared File System executed by command.
- Chapter 21 File System Management
This chapter describes the procedures for managing the GFS Shared File System using basic commands.
- Chapter 22 File System Backing-up and Restoring
This chapter describes how to backing up and restoring data in the GFS Shared File System.
- Chapter 23 Performance and Tuning
This chapter describes how to use a variety of utilities to optimize and make effective use of the GFS Shared File System.
- Chapter 24 Migration to the GFS Shared File System
This chapter describes how to migrate from existing file systems such as the UFS and GFS Local File Systems to GFS Shared File Systems.
- Appendix E List of Messages
This appendix explains GFS Shared File System messages.
-

Appendix F Reference Manual

This appendix is intended for use as a reference manual for the GFS Shared File System.

Appendix G Troubleshooting

This appendix explains messages for which emergency action is required.

Glossary

The glossary defines the terms related to the GFS Shared File System.

Related documentation

The documents listed in this section contain information relevant to GFS file system. Before beginning to work with GFS file system, read the following document:

- PRIMECLUSTER Concept Guide
- PRIMECLUSTER Configuration and Administration Guide
- PRIMECLUSTER Cluster Foundation Configuration and Administration Guide
- PRIMECLUSTER RMS Configuration and Administration Guide
- PRIMECLUSTER Web-Based Admin View Operation Guide
- PRIMECLUSTER Global Disk Services Configuration and Administration Guide
- PRIMECLUSTER Global Link Services Configuration and Administration Guide (MPnet)
- PRIMECLUSTER Global Link Services Configuration and Administration Guide (HaNet)



Note

A related document of PRIMECLUSTER includes the following documents besides the above-mentioned manual.

- PRIMECLUSTER Installation Guide for Solaris
Installation instructions on the paper appended to each product of PRIMECLUSTER.

Manual Printing

Use the PDF file supplied in a CD-ROM for each product to print this manual.

Adobe Acrobat Reader Version 4.0 or higher is required to read and print this PDF file.

Online Manuals

To reference the online manual, use the Cluster management server to register the user name in user group wvroot, clroot, cladmin, or clmon.

To enter a user group and meaning thereof, see "PRIMECLUSTER Web-Based Admin View Operation Guide."

Notational Conventions

Notation

This document adopts the following notational conventions.

Prompts

Command line examples that require system administrator (or root) rights to execute are

preceded by the system administrator prompt, the hash sign (#). Entries that do not require system administrator rights are preceded by a dollar sign (\$).

Manual page section numbers





References to the Solaris(TM) operating system commands are followed by their manual page section numbers in parentheses __ for example, cp(1).

The keyboard

Keystrokes that represent nonprintable characters are displayed as key icons such as [Enter] or [F1]. For example, [Enter] means press the key labeled *Enter*; [Cntl]+[B] means hold down the key labeled *Cntl* or *Control* and then press the [B] key.

Symbol

Material of particular interest is preceded by the following symbols in this manual:

-  Describes the contents of an important point.
-  Describes the points the users should pay close attention to.
-  Provides useful information related to the topic.
-  Provides manuals for users' reference.

Abbreviated name

- Solaris(TM) Operating Environment is abbreviated as Solaris OE.
- Solaris(TM) 9 Operating Environment is abbreviated as Solaris 9 OE.
- Solaris(TM) 8 Operating Environment is abbreviated as Solaris 8 OE.
- Microsoft(R) Windows NT(R) Server operating System Version4.0 and Microsoft(R) Windows NT(R) Server workstation System Version4.0 are abbreviated as Windows NT(R).
- Microsoft(R) Windows(R) 98 operating system and Microsoft(R) Windows(R) 98 Second Edition operating system are abbreviated as Windows(R) 98.
- Microsoft(R) Windows(R) 2000 operating system is abbreviated as Windows(R) 2000.
- Microsoft(R) Windows(R) XP operating system is abbreviated as Windows(R) XP.
- Windows(R) 95, Windows(R) 98, Windows NT(R), Windows(R) 2000, and Windows(R) Me are abbreviated as Microsoft(R) Windows.

February 2003

Trademarks

Sun, Sun Microsystems, the Sun Logo, Solaris and all Solaris based marks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries, and are used under license.

Java and all Java-related products and logs are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Netscape and Netscape Navigator are trademarks of Netscape Communications Corporation in the United States.

Microsoft, Windows, Windows NT, Windows Me, Internet Explorer and Windows2000 are registered trademarks of Microsoft Corporation in the United States and other countries.

Screen images are used in accordance with the guidelines of Microsoft Corporation.

PRIMECLUSTER is a trademark of FUJITSU LIMITED.

Other product names are product names, trademarks, or registered trademarks of these companies.

Copyright (c) 1986-1997 Sun Microsystems, Inc.

Requests

- No part of this document may be reproduced or copied without permission of FUJITSU LIMITED.
- The contents of this document may be revised without prior notice.

Copyright (C) 2003 FUJITSU LIMITED.

All Rights Reserved.

Copyright (C) 2003 Fujitsu Siemens Computers GmbH.

All Rights Reserved.

Contents

Part 1 Overview of Global File Services	1
Chapter 1 Overview of Global File Services	3
1.1 Types of GFS File System	4
1.2 Features of the GFS Local File System	5
1.3 Features of the GFS Shared File System	6
Part 2 Global File Services Local File System	7
Chapter 2 GFS Local File System	9
2.1 GFS Local File System	10
2.2 Multi-partition Configuration	11
2.3 Extension of the File System.....	12
2.4 High-speed Recovery of the File System.....	13
2.5 Improvement of File System Performance by Allocation of Continuous Blocks	14
2.6 Unique Meta-data Cache Management	15
2.7 Statistical Information on the File System	16
2.8 Area Reallocation in case of Disk Block Failure.....	17
2.9 Setting of Extent Attribute Information	18
2.10 Upper Limits for the File System	19
2.11 Action Environment.....	20
Chapter 3 Structure of the GFS Local File System.....	21
3.1 Disk Structure of the GFS Local File System.....	22
3.2 Super Block.....	23
3.3 Partition Configuration Data	24
3.4 Update Log Area	25
3.5 Meta-data Area	26
3.5.1 Area management data	26
3.5.2 i-node area	26
3.5.3 V-data area.....	26
3.6 File Data Area.....	27
3.7 Multi-partition Configuration	28
Chapter 4 Management of the GFS Local File System	31
4.1 Commands for GFS Local File System Management.....	32
4.2 Notes on GFS Local File System Commands	34
4.3 Creating a GFS Local File System	35
4.3.1 Creating a GFS Local File System	35
4.3.2 Customizing the GFS Local File System	36
4.4 Mounting or Unmounting a GFS Local File System.....	37
4.4.1 Mounting a GFS Local File System	37
4.4.2 Unmounting the file system	37
4.5 Checking the File System for Consistency (fsck).....	38
4.5.1 Action method fsck terminates abnormally	38
4.6 Obtaining Various Types of File System Information	40
4.7 Extending a GFS Local File System	41
4.7.1 Adding new partition to extend file system.....	41
4.8 Displaying Partition and Node Information.....	43
4.9 Changing Partition or Node Information.....	44
Chapter 5 Backing Up and Restoring the GFS Local File System.....	45
5.1 Backing Up and Restoring by sfxdump and sfxrestore	46
5.1.1 Backing up by executing sfxdump command	46
5.1.2 Restoring by executing sfxrestore command.....	47
5.2 Backing Up and Restoring by Executing sfxmkbkup Command.....	50
5.3 Backing Up and Restoring by Executing dd Command	52
5.3.1 Backing up by executing dd command.....	52
5.3.2 Restoring by executing dd command	53
5.4 Change Settings after Restoreing	55
5.4.1 Re-setting of partition information.....	55

5.4.2 recovery of file systems.....	55
Chapter 6 Commands in the GFS Local File System.....	57
6.1 Setting the Extent Attribute Information.....	58
6.2 Reference of the Extent Attribute Information.....	59
6.3 Commands That Reflect the Extent Attribute Information.....	60
Chapter 7 Managing Restriction of GFS Local File System Usage.....	61
7.1 quota Function.....	62
7.2 quota Management Method for GFS Local File System.....	63
7.3 quota Function Commands.....	64
7.4 Setup of quota Function.....	65
7.4.1 Setting up quota function.....	65
7.5 Enabling and Disabling quota Function.....	67
7.5.1 Enabling at system boot.....	67
7.5.2 Enabling at command execution.....	67
7.5.3 Disabling quota function.....	67
7.6 Notes on Operation of quota Function.....	69
7.6.1 Matching quota data.....	69
7.6.2 external quotas file and internal quotas file.....	69
7.6.3 Handling users for which limit values are not specified.....	69
7.6.4 Tuning parameter.....	69
7.6.5 Kernel messages.....	70
7.7 Differences from UFS.....	71
7.7.1 quotas files.....	71
7.7.2 quotacheck(1M).....	71
7.7.3 quota(1M).....	71
7.7.4 sfxquotadm(1M).....	71
Chapter 8 Defragmenting GFS Local File System.....	73
8.1 Defragmentation Function.....	74
8.2 Defragmentation Policies.....	75
8.3 Defragmentation Command.....	76
Chapter 9 Performance and Tuning.....	77
9.1 Update Logs.....	78
9.1.1 Is the capacity of the update log area sufficient?.....	78
9.1.2 Is the update log area too large?.....	78
9.1.3 Is it better to allocate the update log area to another partition?.....	78
9.2 Performance and Tuning.....	79
9.2.1 Statistical information of file system.....	79
9.2.2 Performance Verification.....	80
9.2.3 Tuning Parameters.....	83
9.3 Monitoring the Free Area.....	86
Chapter 10 Using GFS Local File System in RMS cluster Operation.....	87
10.1 Before using a GFS Local File System in RMS cluster operation.....	88
10.2 When Using a GFS Local File System as an Alternate cluster file system.....	89
10.3 Setup flow for using GFS Local File System for the GDS volume as an alternate file system.....	90
10.4 Setup procedure for using GFS Local File System for the GDS volume as an alternate file system.....	91
10.5 Setup flow for adding a data partition in the GFS Local File System of the alternate file system.....	93
10.6 Setup procedure for adding a data partition to the GFS Local File System of the alternate file system.....	94
10.7 Using GFS Local File System as NFS Server in Stand-by Operation.....	96
Chapter 11 Changing Node information for GFS Local File System.....	97
11.1 When the Name of a Node Using GFS Local File System is Changed.....	98
11.2 When the Motherboard of a Node Using GFS Local File System is Replaced.....	99
Appendix A Kernel Messages.....	101
A.1 Diagnostics.....	102
A.1.1 NOTICE: sfxfs:0001: <i>mount_point</i> : file system full.....	102
A.1.2 NOTICE: sfxfs:0002: <i>mount_point</i> : out of inodes.....	102
A.1.3 NOTICE: sfxfs:0003: <i>mount_point</i> : file system meta-data full (Vdata) type = <i>number</i>	102

A.1.4 NOTICE: sfxfs:0004: <i>mount_point</i> : too much fragment (extent).....	102
A.1.5 NOTICE: sfxfs:0005: <i>mount_point</i> : daemon create failed	102
A.1.6 NOTICE: sfxfs:0010: illegal mode (recovered): <i>mount_point</i> <i>ino_num</i> mode = <i>file_mode</i>	103
A.1.7 NOTICE: sfxfs:0011: illegal free inode (recovered): <i>mount_point</i> <i>ino_num</i> had size <i>size</i>	103
A.1.8 NOTICE: sfxfs:0012: illegal mode (recovered): <i>mount_point</i> <i>ino_num</i> mode = <i>file_mode</i>	103
A.1.9 NOTICE: sfxfs:0013: illegal mode (recovered): <i>mount_point</i> <i>ino_num</i>	103
A.1.10 NOTICE: sfxfs:0014: illegal mode (recovered): <i>mount_point</i> <i>ino_num</i> new size = <i>size</i> blocks = <i>blocks</i>	103
A.1.11 PANIC: sfxfs:0015: <i>mount_point</i> : bad <i>v_count</i> == 0	103
A.1.12 PANIC: sfxfs:0016: <i>mount_point</i> : bad <i>v_count</i> == 0.....	104
A.1.13 NOTICE: sfxfs:0020: <i>mount_point</i> : bad <i>namelen</i> == 0 parent <i>ino</i> = <i>number</i>	104
A.1.14 PANIC: sfxfs:0021: <i>mount_point</i> : illegal operation: <i>op</i> = <i>code</i>	104
A.1.15 WARNING: sfxfs:0022: <i>mount_point</i> : bad dir <i>ino</i> <i>number</i> at offset <i>offset</i> : <i>info</i>	104
A.1.16 WARNING: sfxfs:0023: <i>mount_point</i> : target directory link count < 2 <i>ino</i> = <i>number</i>	104
A.1.17 WARNING: sfxfs:0024: <i>mount_point</i> : invalid offset <i>offset</i> <i>ino</i> = <i>number</i>	105
A.1.18 PANIC: sfxfs:0025: <i>mount_point</i> : invalid slot status <i>code</i>	105
A.1.19 NOTICE: sfxfs:0026: <i>mount_point</i> : bad <i>namelen</i> == 0 parent <i>ino</i> = <i>number</i>	105
A.1.20 WARNING: sfxfs:0027: <i>mount_point</i> : bad dir, <i>ino</i> = <i>number</i>	105
A.1.21 WARNING: sfxfs:0030: <i>mount_point</i> : meta-data read error. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	105
A.1.22 WARNING: sfxfs:0031: <i>mount_point</i> : meta-data read error. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	105
A.1.23 WARNING: sfxfs:0032: <i>mount_point</i> : bitmap read error. type = <i>number</i> dev= <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	106
A.1.24 WARNING: sfxfs:0033: <i>mount_point</i> : bitmap read error. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	106
A.1.25 WARNING: sfxfs:0034: <i>mount_point</i> : bitmap read error. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	106
A.1.26 WARNING: sfxfs:0035: <i>mount_point</i> : bitmap write error. information was lost. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	106
A.1.27 WARNING: sfxfs:0036: <i>mount_point</i> : bitmap write error. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	106
A.1.28 WARNING: sfxfs:0037: <i>mount_point</i> : meta-data write error. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	107
A.1.29 WARNING: sfxfs:0038: <i>mount_point</i> : fail to update super bock at first time dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	107
A.1.30 WARNING: sfxfs:0039: <i>mount_point</i> : bitmap write error. information was lost. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	107
A.1.31 WARNING: sfxfs:0040: <i>mount_point</i> : meta-data write error. information was lost. type = <i>number</i> dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	107
A.1.32 NOTICE: sfxfs:0041: <i>mount_point</i> : I/O error on log: dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i> error = <i>number</i>	107
A.1.33 WARNING: sfxfs:0042: <i>mount_point</i> : LOG daemon is dead. dev = <i>dev_num</i>	108
A.1.34 WARNING: sfxfs:0043: <i>mount_point</i> : super block write error. dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	108
A.1.35 WARNING: sfxfs:0044: <i>mount_point</i> : super block write error. information was lost. dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	108
A.1.36 WARNING: sfxfs:0045: <i>mount_point</i> : cannot access: too many I/O errors. dev = <i>dev_num</i>	108
A.1.37 WARNING: sfxfs:0046: <i>mount_point</i> : cannot access: too many I/O errors. dev = <i>dev_num</i>	109
A.1.38 NOTICE: sfxfs:0050: setting <i>sfxfs_ninode</i> to max value of <i>value</i>	109
A.1.39 NOTICE: sfxfs:0051: freeing wrong file system by <i>umount</i> <i>file_num</i>	109
A.1.40 PANIC: sfxfs:0052: <i>v_count</i> != 0 <i>ino_pointer</i> <i>ino_num</i>	109
A.1.41 PANIC: sfxfs:0053: <i>mount_point</i> : illegal transaction buffer. <i>bufno</i> = <i>number</i>	109
A.1.42 PANIC: sfxfs:0054: <i>mount_point</i> : illegal transaction buffer. <i>bufno</i> = <i>number</i>	109
A.1.43 PANIC: sfxfs:0055: <i>mount_point</i> : buffer overflow. size = <i>size</i>	110
A.1.44 PANIC: sfxfs:0056: <i>mount_point</i> : hold count is invalid. type = <i>number</i> number = <i>number</i>	110
A.1.45 PANIC: sfxfs:0057: <i>page_create</i> == NULL.....	110
A.1.46 PANIC: sfxfs:0058: <i>bn</i> == SFXFS_HOLE.....	110

A.1.47 PANIC: sfxfs:0059: bn == SFXFS_HOLE.....	110
A.1.48 NOTICE: sfxfs:0060: <i>mount_point</i> : read only mount, but modified (recovered):	110
A.1.49 PANIC: sfxfs:0062: bn == SFXFS_HOLE and rw == S_WRITE	111
A.1.50 WARNING: sfxfs:0063: <i>mount_point</i> : cannot delete extent. ino = <i>number</i> len = <i>len</i> width = <i>width</i> length = <i>len</i>	111
A.1.51 WARNING: sfxfs:0064: <i>mount_point</i> : no extent exists. ino = <i>number</i> osize = <i>len</i> len = <i>len</i>	111
A.1.52 WARNING: sfxfs:0065: <i>mount_point</i> : cannot delete extent. ino = <i>number</i> len = <i>len</i> width = <i>width</i> length = <i>len</i>	111
A.1.53 WARNING: sfxfs:0070: <i>mount_point</i> : super block read error. dev = <i>dev_num</i>	111
A.1.54 WARNING: sfxfs:0071: <i>mount_point</i> : primary super block read failed, but secondary read. dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	112
A.1.55 WARNING: sfxfs:0072: <i>mount_point</i> : secondary super block read failed, but primary read. dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	112
A.1.56 NOTICE: sfxfs:0073: <i>mount_point</i> : VTOC read error. dev = <i>dev_num</i>	112
A.1.57 WARNING: sfxfs:0074: <i>mount_point</i> : primary and secondary super blocks are different. dev = <i>dev_num</i> ..	112
A.1.58 WARNING: sfxfs:0075: <i>mount_point</i> : partition configuration read error. dev = <i>dev_num</i>	113
A.1.59 WARNING: sfxfs:0076: <i>mount_point</i> : primary partition configuration read failed, but secondary read. dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	113
A.1.60 WARNING: sfxfs:0077: <i>mount_point</i> : secondary partition configuration read failed, but primary read. dev = <i>dev_num</i> block = <i>block_num</i> size = <i>size</i>	113
A.1.61 WARNING: sfxfs:0078: <i>mount_point</i> : primary and secondary partition configurations are different. dev = <i>dev_num</i>	113
A.1.62 NOTICE: sfxfs:0079: <i>mount_point</i> : nodename mismatch. node <-> node	114
A.1.63 WARNING: sfxfs:0080: <i>mount_point</i> : node information not found for <i>node</i>	114
A.1.64 WARNING: sfxfs:0081: <i>mount_point</i> : super blocks differ. dev = <i>dev_num</i> to <i>dev_num</i>	114
A.1.65 WARNING: sfxfs:0082: <i>mount_point</i> : partition configurations differ. dev = <i>dev_num</i> to <i>dev_num</i>	114
A.1.66 sfxfs:0083: <i>mount_point</i> : over file hard limit (pid <i>pnum</i> , uid <i>unum</i>)	114
A.1.67 sfxfs:0084: <i>mount_point</i> : Warning: too many files (pid <i>pnum</i> , uid <i>unum</i>)	115
A.1.68 sfxfs:0085: <i>mount_point</i> : over file and time limit (pid <i>pnum</i> , uid <i>unum</i>).....	115
A.1.69 sfxfs:0086: <i>mount_point</i> : over hard disk limit (pid <i>pnum</i> , uid <i>unum</i> , inum <i>ino</i>)	115
A.1.70 sfxfs:0087: <i>mount_point</i> : Warning: over disk limit (pid <i>pnum</i> , uid <i>unum</i> , inum <i>ino</i>)	115
A.1.71 sfxfs:0088: <i>mount_point</i> : over disk and time limit (pid <i>pnum</i> , uid <i>unum</i> , inum <i>ino</i>)	115
A.1.72 WARNING: sfxfs:0089: <i>mount_point</i> : segmap_release write error. addr = <i>addr</i> , offset = <i>offset</i> , error = <i>error</i> , ino = <i>ino</i>	116
A.1.73 WARNING: sfxfs:0100: <i>mount_point</i> : cannot find specified volume information in partition configuration. dev = <i>dev_num</i>	116
A.1.74 NOTICE : sfxfs:0110: <i>tunename</i> is too small (<i>num1</i>). Setting <i>num2</i>	116
A.1.75 NOTICE : sfxfs:0111: <i>tunename</i> is too big (<i>num1</i>). Setting <i>num2</i>	116
A.1.76 NOTICE : sfxfs:0112: <i>mount_point</i> : <i>tunename</i> is too small (<i>num1</i>). Setting <i>num2</i>	116
A.1.77 NOTICE : sfxfs:0113: <i>mount_point</i> : <i>tunename</i> is too big (<i>num1</i>). Setting <i>num2</i>	117
A.1.78 WARNING : sfxfs:0120: <i>mount_point</i> : meta data cache grow failed because of size limitation	117
Appendix B Reference Manual.....	119
B.1 Commands for Common File System Management.....	120
B.1.1 <i>clri_sfxfs(1M)</i> – Clears an i-node	120
B.1.2 <i>df_sfxfs(1M)</i> - Reports the free disk area of the GFS Local File System	121
B.1.3 <i>ff_sfxfs(1M)</i> - Displays file names and statistical information	123
B.1.4 <i>fsck_sfxfs(1M)</i> - Checks and repairs the GFS Local File System	124
B.1.5 <i>fstyp_sfxfs(1M)</i> - Determines the type of a file system.....	126
B.1.6 <i>mkfs_sfxfs(1M)</i> - Builds the GFS Local File System	127
B.1.7 <i>mount_sfxfs(1M)</i> - Mounts the GFS Local File System.....	130
B.1.8 <i>ncheck_sfxfs(1M)</i> - Lists file names and i-node numbers	132
B.2 Management Commands Specific to the GFS Local File System	133
B.2.1 <i>sfxadd(1M)</i> - Extends the GFS Local File System	133
B.2.2 <i>sfxadm(1M)</i> - Changes GFS Local File System partition information settings	135
B.2.3 <i>sfxddr(1M)</i> - Restores backup data and checks device restoring.....	137
B.2.4 <i>sfxdefrag(1M)</i> - Defragments the GFS Local File System.....	139
B.2.5 <i>sfxdump(1M)</i> - Dumps a GFS local file system	142

B.2.6	sfxedquota(1M) - Edits user quotas for the GFS Local File System.....	147
B.2.7	sfxinfo(1M) - Displays GFS Local File System partition information.....	149
B.2.8	sfxmkbkup(1M), sfxmkrstr(1M) – Creates a shell script for backing up.....	151
B.2.9	sfxnewfs(1M) – Builds the GFS Local File System	154
B.2.10	sfxnode(1M) - Adds or changes GFS Local File System node configuration information.....	156
B.2.11	sfxquota(1M) - Displays a user's GFS Local File System disk quota and usage.....	158
B.2.12	sfxquotadm(1M) - Manages quotas file for the GFS Local File System	159
B.2.13	sfxquotaon(1M), sfxquotaoff(1M) - Turns GFS Local File System quotas on and off	161
B.2.14	sfxrepquota(1M) - Summarizes quotas for the GFS Local File System.....	162
B.2.15	sfxrestore(1M) - Restores from the dump created with sfxdump.....	163
B.2.16	sfxstat(1M) - Reports statistical information on the GFS Local File System.....	168
B.2.17	sfxtune(1M) - Tunes up the GFS Local File System.....	171
B.3	General Commands Specific to the GFS Local File System	172
B.3.1	sfxcp(1) - Copies files with extent attribute information.....	172
B.3.2	sfxcpio(1) - Extracts or restores files from an archive with extent attribute information.....	175
B.3.3	sfxgetext(1) - Displays extent attribute information	180
B.3.4	sfxmv(1) - Moves files with extent attribute information	181
B.3.5	sfxsetext(1) - Specifies extent attribute information.....	183
B.4	Miscellaneous	185
B.4.1	fs_sfxfs(4) – The partition management format of the GFS Local File System	185
Appendix C	Corrective Measures against I/O Errors in the GFS Local File System.....	187
C.1	If the message "sfxfs...: read error" or "sfxfs...:write error" is output.....	188
Appendix D	Command correspondence with GFS Local File System and UFS file system	189
D.1	Correspondence with GFS Local File System commands and UFS commands.....	190
Part 3	Global File Services Shared File System.....	193
Chapter 12	File System Functions.....	195
12.1	File System Overview	196
12.1.1	Basic hardware configuration for use of file system	197
12.1.2	File system sharing with PRIMEPOWER 800/900/1000/1500/2000/2500.....	198
12.2	Simultaneous Shared Access Maintaining Coherency	199
12.3	High Availability	200
12.3.1	Uninterrupted operation in case of node failure.....	200
12.3.2	High-speed file system recovery	201
12.3.3	Area reassignment in case of disk block failure.....	201
12.3.4	Operation Continuity If the LAN Fault Occurs.....	202
12.4	Performance.....	203
12.4.1	Data access performance.....	203
12.4.2	Contiguous block assignment.....	203
12.4.3	Multi-partition configuration	203
12.4.4	Individualized meta-cache management	204
12.5	Extendability	205
12.5.1	File system extension.....	205
12.5.2	Adding shared nodes.....	206
12.6	High-speed backup and restoration of a file system.....	207
12.7	File Extension Attribute Information	208
12.7.1	Extent Attribute Information	208
12.7.2	Ignore Update Last Access Time Attribute Information.....	208
12.7.3	Ignore Update Last Modify Time Attribute Information.....	208
12.7.4	File Data Allocation Attribute Information.....	208
12.8	Operability	209
12.8.1	GUI.....	209
12.9	Upper Limits of the File System	210
12.10	Notes 211	211
12.10.1	Relationships with Other Components of PRIMECLUSTER	211
12.10.2	Unavailable functions	211
12.10.3	Service ports used by the GFS Shared File System	212
12.10.4	Notes on use as loopback virtual file system(lofs).....	212

12.10.5 Notes on use of logical volumes of GDS	212
12.10.6 Note on stopping a node	212
Chapter 13 File System Structure	213
13.1 Disk Structure of File System	214
13.1.1 Super block	214
13.1.2 Partition configuration data.....	215
13.1.3 Meta-data area	215
13.1.4 Update log area.....	215
13.1.5 File data area	216
13.1.6 Partition configuration.....	216
13.2 Component configuration	221
13.2.1 MDS(Meta-data Server)	223
13.2.2 AC(Access Client)	223
13.2.3 File System Driver.....	223
13.2.4 Monitoring daemon.....	223
13.3 File Access and Token Management.....	224
Chapter 14 Failure Recovery	225
14.1 MDS Failure Recovery	226
14.1.1 Automatic recovery for primary MDS failures	226
14.1.2 Automatic recovery for secondary MDS failures.....	227
14.1.3 Automatic recovery in case primary MDS goes down during primary MDS operation only	228
14.1.4 Automatic recovery by standby MDS node.....	229
14.2 AC degeneration.....	230
Chapter 15 File System Design	231
15.1 Mount and unmount opportunity of the file system at the time of system starting and ending	232
15.2 System Design	233
15.2.1 Effects of file system operation and system load.....	233
15.2.2 Node layout.....	233
15.2.3 Disk layout.....	234
15.2.4 Work Disk for fsck_sfcfs(1M).....	235
15.2.5 LAN selection	236
15.2.6 quota function.....	236
15.2.7 For future expansion	238
15.3 Backup Design	239
Chapter 16 Management Partition	241
16.1 Management Partition	242
16.2 Resources Required for the Management Partition	243
16.3 Notes on Management	244
16.4 Setting up the Management Partition to Start the file system monitoring mechanism.....	245
16.5 Backup and restore of the management partition information	248
Chapter 17 Starting and Exiting the Management View.....	251
17.1 Starting Web-Based Admin View.....	252
17.2 Web-Based Admin View Top Menu	253
17.2.1 Web-Based Admin View Operation Menu Functions.....	253
17.2.2 Web-Based Admin View Tool Menu Functions	253
17.3 Starting GFS Management View	254
17.4 Exiting GFS Management View	256
17.5 Exiting Web-Based Admin View	257
Chapter 18 Operation Management View Screen Elements.....	259
18.1 Screen Configuration.....	260
18.2 Menu Configurations and Functions.....	263
18.2.1 General	263
18.2.2 Operation	264
18.2.3 View	267
18.2.4 Help.....	268
18.3 Icon Types and Object Status.....	269
Chapter 19 File System Operations (GUI)	271

19.1	Flow of Operations	272
19.1.1	Creation.....	272
19.1.2	Change (in quota function status).....	273
19.1.3	Change (in file system attributes)	274
19.1.4	Change (partition configuration)	275
19.1.5	Change (shared node information).....	276
19.1.6	Deletion	276
19.2	Creation.....	278
19.2.1	Creating a file system.....	278
19.3	Change.....	286
19.3.1	Changing in file system attributes.....	286
19.3.2	Changing the file system configuration (partition addition)	289
19.3.3	Changing the shared node information	291
19.4	Deletion	294
19.4.1	Deleting the file system	294
Chapter 20	File System Operations (Command)	295
20.1	Operation flow	296
20.1.1	Creation.....	296
20.1.2	Change (in quota function status).....	297
20.1.3	Change (in file system attributes)	297
20.1.4	Change (partition configuration)	298
20.1.5	Change (shared node information).....	299
20.1.6	Changes (re-creating a file system).....	300
20.1.7	Change (MDS operational information)	300
20.1.8	Deletion	301
20.2	Flow from file system creation to mount/unmount	302
20.2.1	Setting shared disks	302
20.2.2	Creating a file systems	302
20.2.3	Setting /etc/vfstab.....	312
20.2.4	mount/umount/all nodes (u)mount.....	313
20.2.5	Checking file system status.....	316
20.2.6	Notes applying when the partitions of a created file system are used	317
20.3	Changes (in quota function status)	318
20.3.1	Setting the quota function.....	318
20.3.2	Enabling the quota function	320
20.3.3	Disabling the quota function	320
20.4	Change (file system attributes)	321
20.4.1	Unmount.....	321
20.4.2	Change the file system attributes	321
20.4.3	Mount	321
20.5	Change (partition configuration).....	322
20.5.1	Changing an unmounted file system	322
20.5.2	Changing while mounted status	324
20.6	Change (shared node information)	327
20.6.1	Unmount.....	327
20.6.2	Changing the management partition information.....	327
20.6.3	Changing the shared node information	327
20.6.4	Mount	328
20.7	Change (re-creating a file system).....	329
20.7.1	Unmount.....	329
20.7.2	Re-creating the file system.....	329
20.7.3	Mount	329
20.8	Change (MDS operational information)	330
20.8.1	Unmount.....	330
20.8.2	Changing the MDS operational information.....	330
20.8.3	Mount	332

20.9 Deleting	333
20.9.1 Unmount.....	333
20.9.2 Deleting the file system	333
Chapter 21 File System Management.....	335
21.1 File System Management Commands	336
21.2 Checking a File System for Consistency and Repairing It	339
21.2.1 Action to be taken if fsck terminates abnormally.....	339
21.3 File Extension Attribute Information.....	342
21.3.1 Setting the Extent Attribute Information	342
21.3.2 Reference of the Extent Attribute Information.....	343
21.3.3 Ignore Update Time Attribute Information.....	343
21.3.4 File Data Allocation Attribute Information.....	344
21.3.5 Reference of the File Extension Attribute Information	344
21.3.6 Command for Reflecting File Extension Attribute Information	344
21.4 quota function.....	345
21.4.1 Quota Function.....	345
21.4.2 Quota Management Method for the GFS Shared File System	345
21.4.3 Quota Function Commands.....	345
21.4.4 Checking the usage conditions for user	346
21.4.5 Notes on Operation of the Quota Function.....	347
21.4.6 Differences from UFS.....	348
21.5 Extending a File System.....	349
21.6 Displaying File System Information	350
21.6.1 Acquiring file system information.....	350
21.6.2 Displaying partition/node information	351
21.6.3 Displaying file system management state	352
21.7 Multiple LAN Setup.....	353
21.7.1 Setup.....	353
21.7.2 Reference.....	353
21.7.3 Change.....	353
21.8 Using the GFS Shared File System for RMS Cluster Operation.....	354
21.8.1 Before using a GFS shared file system in RMS cluster operation	354
21.8.2 To enable use as a shared file system for RMS cluster operation	354
21.8.3 Setup flow when a GFS shared file system is to be used in an RMS cluster operation.....	355
21.8.4 Setup procedure when the GFS Shared File System is to be used in an RMS cluster operation	355
21.8.5 Setup flow for addition of a file data partition to a GFS shared file system that is used in an RMS cluster operation.....	357
21.8.6 Setup procedure for addition of a file data partition to a GFS shared file system that is used in an RMS cluster operation	357
Chapter 22 File System Backing-up and Restoring	359
22.1 Types of Backups and Restores.....	360
22.2 Backing Up and Restoring using sfdump(1M) and sfcstore(1M)	361
22.2.1 Backing up using the sfdump(1M)	362
22.2.2 Restoration using sfcstore(1M)	363
22.2.3 Notes on using differential backup	365
22.3 Backup/Restoration using sfcmbkbp	366
22.4 Backing Up using Standard Solaris OE commands	368
22.4.1 Backing up file by file.....	368
22.4.2 Backing up an entire file system.....	368
22.5 Restoring using Standard Solaris OE commands	371
22.5.1 Restoring File by file.....	371
22.5.2 Restoring an entire file system	372
22.6 Backing up and Restoring User Limit Values of the quota function	373
22.7 High speed backup and restoration of a GFS shared file system.....	374
22.7.1 Outline.....	374
22.7.2 Implementation methods	374
22.7.3 Precondition	377

22.7.4 Contents of the backup image.....	377
22.7.5 Mount Unmount of a backup place.....	377
22.7.6 Setup of the backup relationship.....	377
22.7.7 Operation at execution time.....	380
22.7.8 Operation at the time of cancellation.....	382
22.7.9 Restoring the backup source.....	383
22.7.10 Displaying the copy state.....	384
22.7.11 Actions following an error.....	385
22.7.12 Notes regarding file system changes.....	386
Chapter 23 Performance and Tuning.....	387
23.1 Update Log.....	388
23.1.1 Is the size of the update log area sufficient?.....	388
23.1.2 Is the size of the update log area too large?.....	388
23.1.3 Should the update log area be separated and placed in another partition?.....	389
23.2 Tuning Parameters.....	390
Chapter 24 Migration to the GFS Shared File System.....	391
24.1 Migration from a UFS file system.....	392
24.2 Migration from a GFS local file system.....	394
Appendix E List of Messages.....	395
E.1 AC Messages of GFS Shared File System.....	396
E.1.1 Panic messages.....	396
E.1.2 Warning messages.....	398
E.1.3 Information messages.....	409
E.2 GFS Shared File System Driver messages.....	414
E.2.1 Information messages.....	414
E.3 GFS Shared File System Daemon messages.....	415
E.3.1 Panic messages.....	415
E.3.2 Error messages.....	416
E.3.3 Warning messages.....	429
E.3.4 Information messages.....	439
E.4 Command messages for File System Common Management.....	445
E.4.1 df_sfscfs command.....	445
E.4.2 fsck_sfscfs command.....	449
E.4.3 mkfs_sfscfs command.....	449
E.4.4 mount_sfscfs command.....	458
E.4.5 umount_sfscfs command.....	463
E.5 Management Command messages Specific to GFS Shared File System.....	468
E.5.1 sfccadd command.....	468
E.5.2 sfccadm and sfccnode commands.....	478
E.5.3 sfccddr command.....	486
E.5.4 sfccdu commnad.....	488
E.5.5 sfccedquota command.....	490
E.5.6 sfccfrmstart command.....	493
E.5.7 sfccfrmstop command.....	496
E.5.8 sfccgetconf command.....	497
E.5.9 sfcchost command.....	500
E.5.10 sfccinfo command.....	505
E.5.11 sfccproxybreak, sfccproxyinfo, sfccproxyjoin, sfccproxyop and sfccproxyrestore command.....	508
E.5.12 sfccmkbkup command.....	512
E.5.13 sfccmkrstr command.....	513
E.5.14 sfccmntgl command.....	514
E.5.15 sfccnewfs command.....	518
E.5.16 sfccquota command.....	526
E.5.17 sfccquotadm command.....	528
E.5.18 sfccquotaon, sfccquotaoff command.....	532
E.5.19 sfccrepquota command.....	537
E.5.20 sfccrscinfo command.....	540

E.5.21 sfcsetup command.....	542
E.5.22 sfcstat command.....	554
E.5.23 sfcctune command.....	555
E.5.24 sfcumntgl command.....	560
E.6 General Command messages Specific to GFS Shared File System.....	565
E.6.1 sfccp and sfcmv commands.....	565
E.6.2 sfccpio command.....	570
E.6.3 sfcgetext command.....	586
E.6.4 sfcsetext command.....	587
E.7 GFS Shared File System Management View Messages.....	589
E.7.1 Error messages.....	589
E.7.2 Warning messages.....	590
E.7.3 Information messages.....	592
E.8 Web-Based Admin View Messages.....	594
E.9 Log Reference Messages.....	595
E.10 Operating Environment Setting Messages.....	596
Appendix F Reference Manual.....	597
F.1 Commands for Common File System Management.....	598
F.1.1 df_sfcfs(1M) Reports the free disk area of GFS Shared File System.....	598
F.1.2 fsck_sfcfs(1M) Checks the GFS Shared File System for consistency and recovers the file system.....	601
F.1.3 fstyp_sfcfs(1M) Determines the type of a file system.....	604
F.1.4 mkfs_sfcfs(1M) Constructs the GFS Shared File System.....	605
F.1.5 mount_sfcfs(1M) Mounts the GFS Shared File System on the self-mode.....	609
F.1.6 umount_sfcfs(1M) Unmounts the GFS Shared File System from the self-node.....	611
F.2 Management Commands Specific to the GFS Shared File System.....	612
F.2.1 sfccadd(1M) Adds file data partitions.....	612
F.2.2 sfccadm(1M) Changes partition information settings.....	614
F.2.3 sfccddr(1M) Restores backup data and checks device restoring.....	618
F.2.4 sfccdu(1M) Summarize disk usage.....	620
F.2.5 sfccdump(1M) Dumps a GFS shared file system.....	622
F.2.6 sfccedquota(1M) Edits quota limits for each user.....	627
F.2.7 sfccfrmstart(1M) Starts FsRM on the self node.....	629
F.2.8 sfccfrmstop(1M) Stops FsRM on the self-node.....	630
F.2.9 sfccgetconf(1M) Backs up the management partition.....	631
F.2.10 sfcchost(1M) Displays and changes a connection host name with MDS.....	633
F.2.11 sfccinfo(1M) Displays partition information.....	635
F.2.12 sfccmkbkup(1M), sfccmkrstr(1M) Generates a shell script for backing up.....	639
F.2.13 sfccmntgl(1M) Mounts the GFS Shared File System on all nodes.....	642
F.2.14 sfccnewfs(1M) Builds the GFS Shared File System.....	643
F.2.15 sfccnode(1M) Adds, deletes, and alters node configuration information.....	647
F.2.16 sfccproxybreak(1M) Cancels the relationship for high speed backup and restoration.....	649
F.2.17 sfccproxyinfo(1M) Displays information about the relationship for high speed backup and restoration.....	650
F.2.18 sfccproxyjoin(1M) Sets up the relationship for high speed backup and restoration.....	654
F.2.19 sfccproxyop(1M) Executes operation for high speed backup and restoration.....	657
F.2.20 sfccproxyrestore(1M) Restores a file system at high speed.....	660
F.2.21 sfccquota(1M) Displays the quota limit value and uses amount for each user.....	662
F.2.22 sfccquotadm(1M) Manages quotas files.....	664
F.2.23 sfccquotaon(1M), sfccquotaoff(1M) Validates and invalidates the quota function.....	666
F.2.24 sfccrepquota(1M) Displays the quota function limits and uses amounts of all users.....	668
F.2.25 sfccrestore(1M) Restores from the dump created with sfccdump.....	670
F.2.26 sfccrscinfo(1M) Displays file system information.....	675
F.2.27 sfccsetup(1M) Initializes the management partition; adds, deletes, displays path of the management partition.....	676
F.2.28 sfccstat(1M) Reports statistics on the GFS Shared File System.....	678
F.2.29 sfccctune(1M) Tunes the GFS Shared File System.....	686
F.2.30 sfccumntgl(1M) Unmounts the GFS Shared File System from all nodes.....	687

F.3 General Commands Specific to the GFS Shared File System.....	688
F.3.1 sfcop(1) Copies files with extent attribute information.....	688
F.3.2 sfcpio(1) Extracts or restores files from an archive with extent attribute information	691
F.3.3 sfcgetext(1) Displays extent attribute information	696
F.3.4 sfcgetfattr(1) Reports extent attribute information of a file	698
F.3.5 sfcmv(1) Moves files with extent attribute information	700
F.3.6 sfcsetext(1) Specifies extent attribute information	702
F.3.7 sfcsettime(1) Sets or deletes ignore update time attribute information	704
F.3.8 sfcsetvolume(1) Sets or deletes file data allocation attribute information	705
Appendix G Trouble shooting.....	707
G.1 Action for a Panic.....	708
G.2 Action for Error Messages	709
G.2.1 Action for I/O errors.....	709
G.2.2 Collecting Investigation Material	710
G.3 GFS Shared File System Management View Abnormality	711
Glossary	713

Part 1 Overview of Global File Services

Chapter 1 Overview of Global File Services

This chapter overviews the Global File Services (hereafter abbreviated to GFS).

1.1 Types of GFS File System

The GFS has two types of file system, the GFS Local File System and the GFS Shared File System.

The GFS Local File System is for local file system in a node, or switchover file system among two or more nodes. The GFS Shared File System can be shared from multiple nodes simultaneously. The most appropriate file system can be chosen depending on the application and the purpose.

The GFS Local File System is the local file system that features high-performance file access with contiguous block allocation of the file data, as well as a high-speed consistency recovery function after a system down, which is effective for high-speed file system switchover.

The GFS Shared File System is a shared file system that features simultaneous accessibility that maintains consistency between multiple Solaris OE to which a shared disk device is connected and enables the continuation of processing on another node even if a certain node fails.

1.2 Features of the GFS Local File System

This section describes the features of the GFS Local File System. For details, see [“Part 2 Global File Services Local File System”](#) of this manual.

The GFS Local File System is used to extend the UFS (a typical UNIX file system) and to improve availability and performance. The major functions of the GFS Local File System are:

- Single file system can use multiple partitions (slices) as a file data area.
- High-speed file system recovery (equivalent function to UFS logging)
- Contiguous block allocation by using extent-based data space management
- Dynamic file system size expansion
- Meta-data reallocation when an error occurred in the disk block
- Setting the extent attribute information to the file
- De-fragmentation function

The GFS Local File System is available for high-speed consistency recovery after a system failure, so that file systems can be switchover if a system fails on one side of a standby configuration.

Also, online data area expansion and the de-fragmentation function can be used without stopping the job.

The GFS Local File System is used on a specific node only, or switchover file system among two or more nodes.



See

Use the RMS when the local file system is used as a switchover file system. For details, see [Part 2, Chapter 10 “Using GFS Local File System in RMS cluster Operation”](#) of this manual.



Note

The GFS Local File System cannot be used as simultaneously directly shared file system among from multiple Solaris OE. For this purpose, use the GFS Shared File System.

1.3 Features of the GFS Shared File System

This section describes the features of the GFS Shared File System. For details, see [Part 3, “Global File Services Shared File System”](#) of this manual.

The GFS Shared File System has API compatibility with the UFS and the GFS Local File Systems and can be accessed simultaneously from multiple Solaris OE. The major functions of the GFS Shared File System are:

- Simultaneous shared access to the file or file system from multiple nodes
- Maintaining of consistency when file data is to be referenced or updated from multiple nodes
- File access is enabled by using the file cache of each node.
- File operation can be continued on another node if a node down occurs while maintaining file system consistency.
- High-speed file system recovery function
- High-speed input and output processing is enabled with contiguous block allocation to the file data area.
- I/O load distribution and mass storage files are enabled with multi partition configuration support.
- The file system data area expansion is enabled without file system reconstruction.
- File system operation with a GUI by using a Web browser



Note

The GFS Shared File System can use only shared volume managed by GDS (Global Disk Services). Other type of disk driver is not applicable.

The GFS Shared File System is shared file system from multiple Solaris OE, so both switchover applications among two or more nodes and scalable applications among several nodes can run on this file system with same contents. When scalable applications are to be executed on multiple nodes, the relevant application can transfer the data on the GFS file system by using the conventional UNIX file API as is.

Also, because file operation can be continued on another node even if a node fails, the GFS Shared File System is particularly suitable for environments in which high availability is required for the file system.

The GFS Shared File System can be used with switchover applications of RMS Cluster Operation. At this time, the file system can be accessed from standby node.



Note

To use the GFS Shared File System with switchover applications of RMS Cluster Operation, do not specify the GFS Shared File System as mpoin resource. For details, see [Part 3, Chapter 21.8 “Using the GFS Shared File System for RMS Cluster Operation”](#) of this manual.

Part 2 Global File Services Local File System

Chapter 2 GFS Local File System

This chapter explains the functions and facilities of the GFS Local File System.

2.1 GFS Local File System

The GFS Local File System is an **extent**-based file system that can be used under the following operating systems:

- 32-bit/64-bit of Solaris 8 OE
- 32-bit/64-bit of Solaris 9 OE

This file system is an extended version of UFS, a representative UNIX file system that provides improved applicability and performance.

The major functions of the GFS Local File System are as follows:

- Multi-partition configuration implemented by allocating multiple **partitions** (slices) in a single file system
- High-speed file system recovery (function equivalent to UFS logging)
- Allocation of continuous blocks by extent-based management
- Extension of file data area in the file system
- Area reallocation in case of disk block failures
- Setting of **extent attribute information** to a file

The following major UFS file system functions are supported by the GFS Local File System as well:

- **64-bit file system interface**
- **Direct I/O**
- **File lock**
- **quota function**
- **ACL**

The GFS Local File System generally supports UFS file system functions with the exception of the following:

- Use as the root file system, /usr, /var, and /opt
- Setting extended file attributes, which is added in Solaris 9 OE
- Mount specifying non-blocking mandatory locking option, which is added in Solaris 9 OE

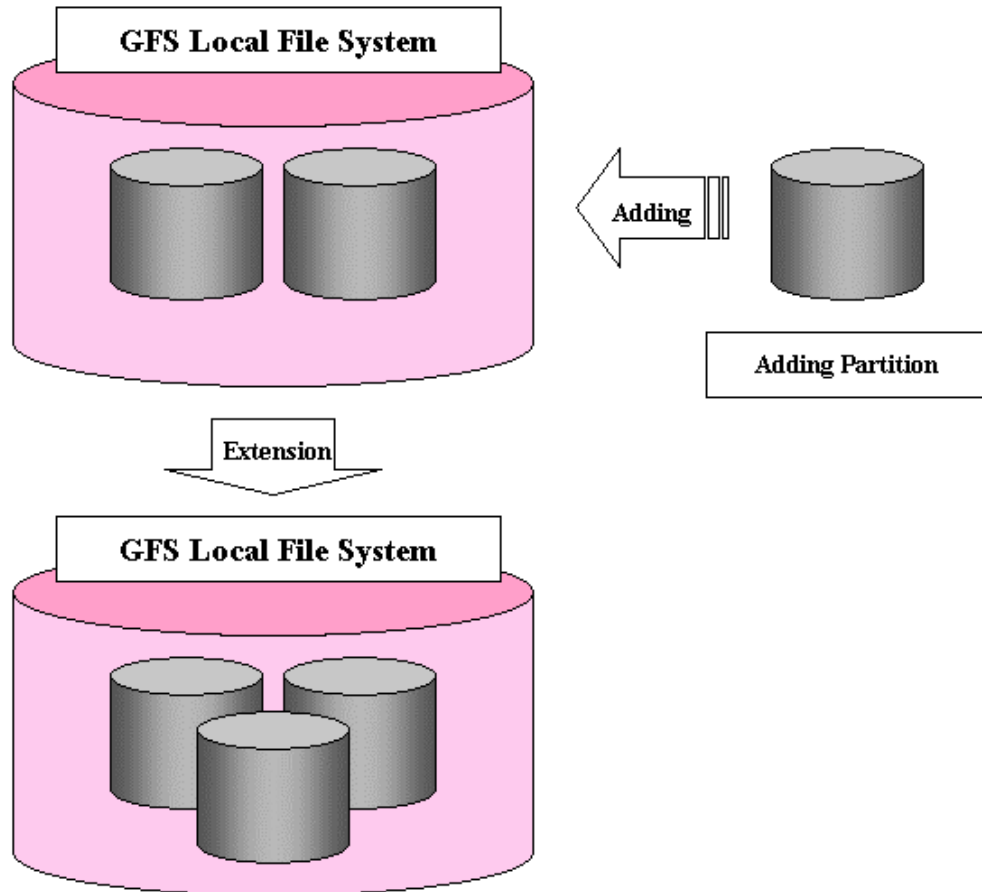
2.2 Multi-partition Configuration

The GFS Local File System supports the function for integrating multiple partitions in a single file system. This function makes it easy to add **file data partitions**, as explained later.

The GFS Local File System consists of multiple partitions. It employs **round robin allocation** for processing **file data areas** that is read from separate partitions for individual files. This function can thus distribute I/O loads over multiple partitions that consist of different disks, which improves file system performance.

2.3 Extension of the File System

Extension of an existing GFS Local File System does not require any data migration: It can be extended by simply adding partitions. This feature can solve area shortage problems without the need to re-create the file system.



2.4 High-speed Recovery of the File System

The GFS Local File System contains an **update log** function for recording updates that were made to the structure of the file system. This function enables the file system to recover from system failure within a few seconds to a minute or so.

When the GFS Local File System recovers from a system failure, the `fsck_sxfs(1M)` command of the file system scans the update log and invalidates or completes file system operations that were being performed at the time the system failure occurred. The file system can be mounted and used soon afterwards, without the need for a full check of the file system structure.



Note

The `fsck_sxfs(1M)` full check mode is supported as well. A full check by `fsck_sxfs(1M)` may be required when a disk has to recover from a hardware failure.

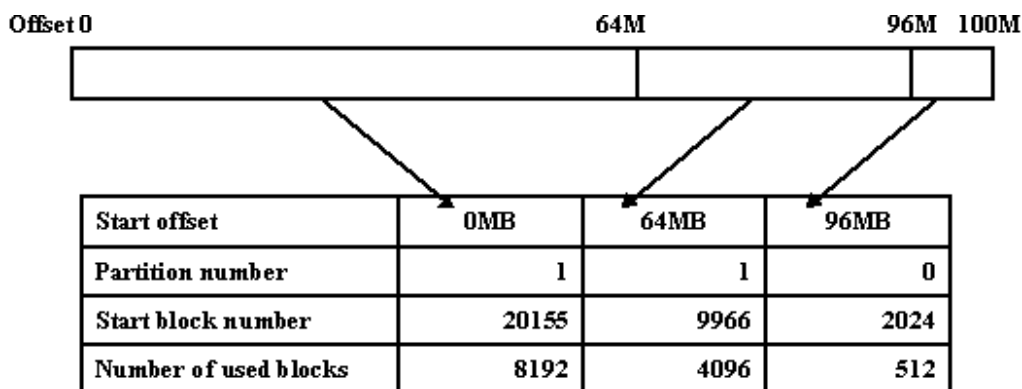
The size of the **update log area** can be specified by an option of `sfxnewfs(1M)`. The default value for this operation is 1% of the size of the file system. However, if 1% of the file system size is less than 5 MB, the default value is set to 5 MB. If it exceeds 50 MB, the default is set to 50 MB.

A larger update log area can improve file system performance, but may increase the time that is required for file system recovery. It is recommended to specify an update log area size of no more than 50 MB.

The update log area can be allocated in a separate partition to avoid that the I/O operations for log updates degrade file system performance. See the [sfxnewfs\(1M\)](#) option for this function.

2.5 Improvement of File System Performance by Allocation of Continuous Blocks

The performance of the GFS Local File System can be improved by allocating continuous blocks for file data and thereby increasing the possibility for I/O batch operations. GFS Local File System performs extent-based area management to support this kind of allocation. For blocks that can be allocated continuously, area allocation is managed by using a combination of start offset, start block number, and the number of used blocks.



Even for file extension after allocation, GFS Local File System arranges blocks as continuously as possible.

GFS Local File System also performs an extent-based management of free data areas. This makes it possible to create an optimum allocation of free areas at high speed.

2.6 Unique Meta-data Cache Management

The GFS Local File System performs a unique cache management for the **meta-data** part in the memory. Conventionally, all parts of meta-data have been processed centrally during cache management. In the GFS Local File System, however, the i-node, directory block, and indirect block areas on the disk are managed separately depending on access characteristics. This management approach improves the cache-hit ratio and reduces the amount of used resources.

2.7 Statistical Information on the File System

Generally speaking, file system performance can be greatly improved by proper tuning.

The GFS Local File System supports multi-partition and extent-based management that can improve the performance of conventional file systems significantly. In order to use these functions more effectively, the user can collect statistical information on the file system and tune the system based on this information.

2.8 Area Reallocation in case of Disk Block Failure

If a disk block hardware error occurs in a newly allocated meta-data area in the GFS Local File System, another disk block is allocated automatically.

This function makes it possible to continue file system processing provided that an error occurs only in a specific block on the disk.



Note

This function ensures that the block that contains the I/O error is temporarily skipped. Therefore, if a request for using this faulty block occurs, a new I/O error may occur. Since such an I/O error prolongs the time to respond to the corresponding request, it is important to remove the cause for the error as soon as possible. If block reallocation occurs due to an I/O error, recover the file system by backing up the file system, removing the cause for the hardware error, such as by replacing the disk, and restore the data that had been backed up.

2.9 Setting of Extent Attribute Information

The GFS Local File System has functions that allocate a data area beforehand to a file, extend a data area in a certain proportion, and guarantee data area continuity. The information that enables these functions is called extent attribute information. These functions enable a system offering higher performance, by ensuring data area allocation and data area continuity.

2.10 Upper Limits for the File System

The table below lists the upper limits that apply to each parts of the GFS Local File System.

Item	Upper limit
Maximum size of the file system	1 terabytes
Maximum size of a file	1 terabytes
Maximum size of a directory	2 gigabytes
Maximum number of partitions in a file system	32 partitions
Maximum number of i-nodes per file system	128 mega-nodes

2.11 Action Environment

GFS Local File System allocates memory cache area of meta-data part (explained at ["2.6 Unique Meta-data Cache Management"](#)) for each file system.

In other words, the following memory capacity is necessary for GFS Local File System as hardware requirements. Here, n means the number of GFS Local File System mounted.

(2.0 + 5.0 × n) megabytes or more than

The size of memory cache area of meta-data part can perform tuning for quantity of resource of file system.

There are two methods to perform tuning it.

- Specify an option in mount(1M)
- Edit /etc/system file

About the method to change size of memory cache area of meta-data part, refer to ["9.2 Performance and Tuning"](#)

And, in GFS Local File System, kernel stack area size is changed 16 Kbytes in package installation in 32-bit of Solaris OE.

Chapter 3 Structure of the GFS Local File System

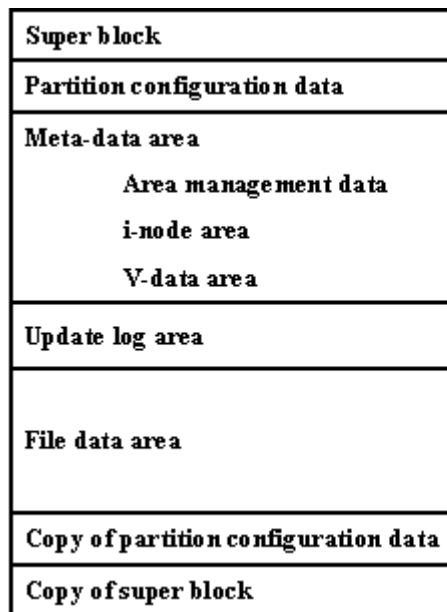
This chapter explains the structure of the GFS Local File System.

3.1 Disk Structure of the GFS Local File System

This section explains the disk structure of the GFS Local File System.

As shown in the figure below, the disk structure of the GFS Local File System consists of the following elements:

- **Super block**
- **Partition configuration data**
- **Update log area**
- **Meta-data area (area management data, i-node area, V-data area)**
- **File data area**



The GFS Local File System manages the V-data area in units of 1024 bytes and file data area in units of 8192 bytes.

While conventional file systems distribute the meta-data management areas over the disk, the GFS Local File System stores meta-data centrally. This not only improves the access speed for meta-data but also localizes updates and shortens the interval between the versions of update logs when the file system is recovered.

3.2 Super Block

The super block contains information on the file system type, dates of creation and updates, label information, size and layout of the file system, and the history of free area information.

The super block copy is located at the end of each partition so that processing can continue even if a disk block failure occurred in the area that contains the super block.

3.3 Partition Configuration Data

The GFS Local File System can consist of multiple partitions. The partition configuration information area contains information on the partitions that constitute the file system and information on the area allocation in each partition.

When the GFS Local File System is shared by multiple **nodes**, device information for identifying each of the nodes is recorded in this area. The `sfxadm(1M)`, `sfxadd(1M)`, or `sfxnode(1M)` commands can be used when the information in this area needs to be changed.

Similar to the super block, the partition configuration information, which is also important for file system operation, is copied at the end of the partition, so that the system remains operable in the event of a disk block failure.

3.4 Update Log Area

The update log area contains the history of updates that were made in the structure of the file system. This information is used to support high-speed file system recovery. Log data is overwritten cyclically.

The update log area contains information on the following events:

- i-node acquisition and release
- i-node information update
- Extent allocation and de-allocation
- V-data acquisition and release
- V-data update

The GFS Local File System ensures that information is written to the update log in advance each time the file system structure is updated. If a system error occurs, use `fsck_sxfs(1M)` to invalidate or complete update operations to the file system for which execution was interrupted: This ensures that the file system stays consistent with the update log. The update log area of the GFS Local File System records only the updates that were made to the file system structure, but does not record any data from files.

3.5 Meta-data Area

3.5.1 Area management data

This area contains i-node, V-data, and file data allocation information.

Management of file data allocation information is based on extents, by combining the part that contains extent information and the list that is used for management depending on the continuity degree.

When the `sfxnewfs(1M)` command is issued, the i-node, V-data, and area management data areas are allocated in fixed sizes.

3.5.2 i-node area

The i-node area is a data structure that contains information on files. Each i-node contains the file type, file length, file owner and group ID, access authority, and a pointer to extent information that includes the file data. There is a single i-node for each of the files.

3.5.3 V-data area

The directory block, symbolic link path name, and indirect block areas are collectively referred to as V-data area, and acquired as necessary. The V-data area is managed in units of 1024 bytes to ensure that directory blocks are used effectively.

3.6 File Data Area

The file data area contains file data. The file data area consists of blocks (with a minimum block size of 8,192 bytes), which are managed in such a way that they can be allocated to files continuously.

3.7 Multi-partition Configuration

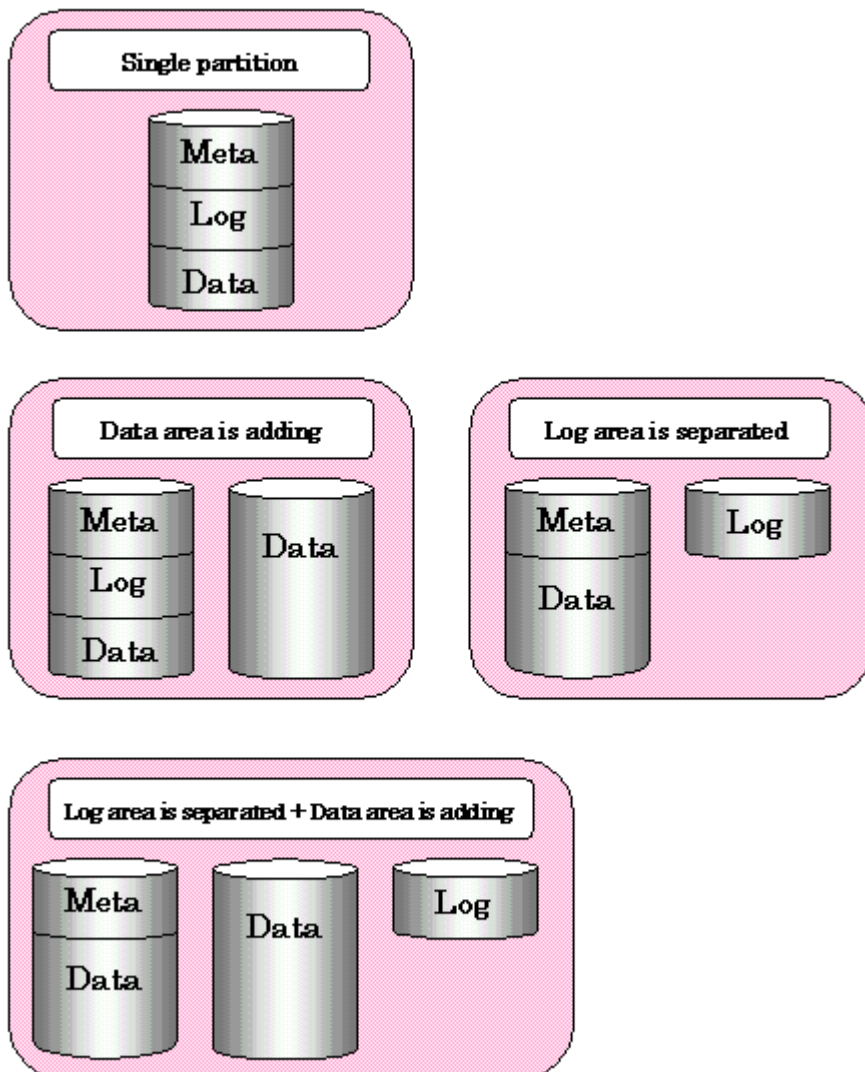
GFS Local File System supports not only file systems that consist only of a single partition, but also multi-partition configurations, in which two or more partitions are allocated to a single file system.

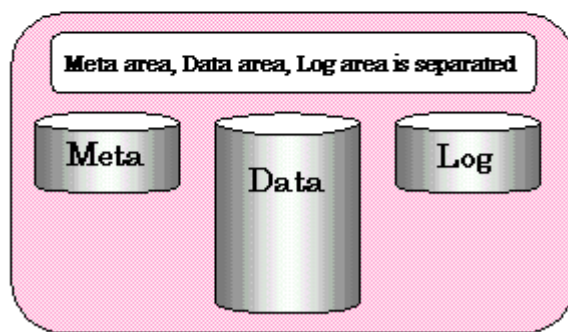
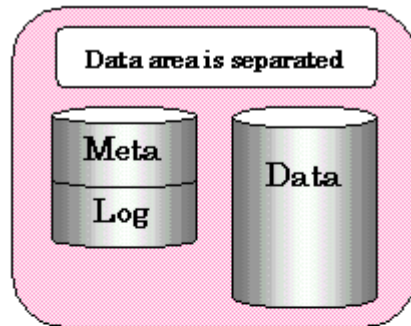
In a single partition configuration, update log, meta-data, and file data areas are all allocated in a single partition.

There are the following five patterns for multi-partition configurations:

- The update log area is reserved in a separate partition (Log area is separated).
- An additional file data area is reserved in a separate partition (Data area is adding).
- The update log area is reserved in a separate partition and an additional file data area is reserved in another file partition (Log area is separated + Data area is adding).
- A file data area is separated from the representative partition (Data area is separated).
- The meta-data area, file data area and update log area are individually independent (meta-data area, file data area, update log are separated).

Every partition contains its own super block and partition configuration information area.





Chapter 4 Management of the GFS Local File System

This chapter explains the procedure for managing the GFS Local File System and the basic commands that are used for management.

4.1 Commands for GFS Local File System Management

For each file system, there are two different types of management commands, namely, the general purpose management commands and the system specific management commands. The general purpose management commands that provided by Solaris OE are used for basic functions and the system specific commands are called by specifying options or file system types.

The GFS Local File System features an exclusive management commands that act as the general purpose management command and the system specific management command for the GFS Local File System's original functions.

Specify "sfxfs" for **-F** option as a file system type when using the general purpose management command to a GFS local file system. If you omit the **-F** option, a search is made of those entries matched with "special" or "mount_point" in "/etc/vfstab," after which the registered file system types are automatically selected.

General-purpose file system management commands

Command	Description
clri(1M)	Clears the i-node.
df(1M)	Displays the number of free disk blocks, number of files, etc.
ff(1M)	Displays file names and statistical information.
fsck(1M)	Checks a file system for consistency and repairs it.
fstyp(1M)	Determines the file system type.
mkfs(1M)	Creates a file system.
mount(1M)	Mounts the GFS Local File System.
ncheck(1M)	Lists file names and i-node numbers.
umount(1M)	Unmounts the GFS Local File System.



See

For details of the general-purpose management commands, see each command's page of "Solaris X Reference Manual."

GFS Local File System management commands

Command	Description
clri_sfxfs(1M)	Clears the i-node.
df_sfxfs(1M)	Displays the number of free disk blocks, number of files, etc.
ff_sfxfs(1M)	Displays file names and statistical information.
fsck_sfxfs(1M)	Checks a file system for consistency and repairs it.
fstyp_sfxfs(1M)	Determines the file system type.
mkfs_sfxfs(1M)	Creates a file system.
mount_sfxfs(1M)	Mounts the GFS Local File System.
ncheck_sfxfs(1M)	Lists file names and i-node numbers.

GFS Local File System specific management commands

Command	Description
sfxadd(1M)	Extends the GFS Local File System.

Command	Description
sfxadm(1M)	Changes GFS Local File System partition information settings.
sfxddr(1M)	Restores backup data and check device restoring.
sfxdefrag(1M)	Defragments the GFS Local File System.
sfxdump(1M)	Dumps a GFS local file system.
sfxedquota(1M)	Edits user quotas for the GFS Local File System.
sfxinfo(1M)	Displays GFS Local File System partition information.
sfxnewfs(1M)	Builds the GFS Local File System. (Front-end command of mkfs_sfxfs(1M))
sfxnode(1M)	Adds or changes GFS Local File System node configuration information.
sfxquota(1M)	Displays a user's GFS Local File System disk quota and usage.
sfxquotadm(1M)	Manages quotas file for the GFS Local File System.
sfxquotaoff(1M)	Turns GFS Local File System quotas on.
sfxquotaon(1M)	Turns GFS Local File System quotas off.
sfxrepquota(1M)	Summarizes quotas for the GFS Local File System.
sfxrestore(1M)	Restores from the dump created with sfxdump.
sfxstat(1M)	Reports statistical information on the GFS Local File System.
sfxtune(1M)	Tunes up the GFS Local File System.



Note

When the GFS Local File System is created, an entry is added to /etc/sfxfstab, which is used for file system management. If, for some reason, /etc/sfxfstab or the file entry is deleted, it must be restored from the backup file (/etc/.sfxfstab.back). Restore the deleted file according to the following procedure.

```
# cp -p /etc/.sfxfstab.back /etc/sfxfstab
```

If there is no backup file present, /etc/sfxfstab must be created. Create a new /etc/sfxfstab according to the following procedure.

```
# touch /etc/sfxfstab
# chmod 644 /etc/sfxfstab
```

Original commands for general users of the GFS Local File System original functions are listed below.

GFS Local File System management commands

Command	Description
sfxcp(1)	Copies files with extent attribute information.
sfxcpio(1)	Extracts or restores files from an archive with extent attribute information.
sfxgetext(1)	Displays extent attribute information.
sfxmv(1)	Moves files with extent attribute information.
sfxsetext(1)	Specifies extent attribute information.

4.2 Notes on GFS Local File System Commands

The GFS Local File System improves performance and operability with the following functions:

- Performance is improved by extent-based block allocation management
- High-speed recovery of the file system becomes possible because of the update log function
- Performance is improved by placing the update log area into another partition
- Performance and operability are improved by placing data in multiple partitions

The GFS Local File System permits to place the update log area in a separate partition. It also permits the allocation of multiple data partitions. The `sfxnewfs(1M)` command can be used for these types of allocation.

While the UFS file system supports the concept of cylinder groups and block fragmentation, the GFS Local File System does not support these functions.

Therefore, the options in connection with the settings for cylinder group and fragmentation that are supported by UFS are not included in the GFS Local File System commands.

4.3 Creating a GFS Local File System

The GFS Local File System can consist not only of a single partition, as in the case of a conventional UFS file system, but can also contain multiple partitions (a multi-partition configuration). The GFS Local File System supports the following two patterns of multi-partition configurations:

- Allocating multiple partitions for the use as file data area: For creating large file systems

4.3.1 Creating a GFS Local File System

Use the `sfxnewfs(1M)` command to create a GFS Local File System.

The default parameters of the `sfxnewfs(1M)` command are explained below.

Default parameters of the `sfxnewfs` command

Parameter	Default
Data block size	Fixed to 8,192 bytes
Minimum free area	10%
Optimization type	Time
Number of i-nodes	One per 8,192-byte disk area
Size of update log area	1% of the size of the file system (but at least 5 MB and not more than 50 MB)
Size of meta-data area	About 10% of the size of the file system
Maximum number of sharing nodes	16
Maximum number of partitions	32

The following shows an example of creating a typical file system.

To create a GFS Local File System in single partition configuration, specify the partition in which the file system is to be created. All types of areas (meta-data area, update log area, and file data area) will be allocated in this partition.

```
# sfxnewfs /dev/sfdsk/gfs/rdsk/vol0300
```

To place the update log area into a separate partition, specify both the **representative partition** and the update log partition. Meta-data area and file data area will be allocated in the representative partition.

```
# sfxnewfs -L /dev/sfdsk/gfs/rdsk/vol0310 /dev/sfdsk/gfs/rdsk/vol0300
```

To use multiple partitions for the data area, specify the representative partition and data partition(s). Meta-data area, update log area, and file data area are created in the representative partition.

```
# sfxnewfs -D /dev/sfdsk/gfs/rdsk/vol0310,/dev/sfdsk/gfs/rdsk/vol0320 /dev/sfdsk/gfs/rdsk/vol0300
```

To place the update log area into a separate partition and specify the multi-partition of the file data area, specify it as follows.

```
# sfxnewfs -L /dev/sfdsk/gfs/rdsk/vol0311 -D /dev/sfdsk/gfs/rdsk/vol0321,/dev/sfdsk/gfs/rdsk/vol0331 /dev/sfdsk/gfs/rdsk/vol0301
```

If specifying the multi-partition of the file data area, the `-A` option can be specified to select the configuration in which the file data area is not included in the representative partition.

```
# sfxnewfs -A -D /dev/sfdsk/gfs/rdsk/vol0321,/dev/sfdsk/gfs/rdsk/vol0331 /dev/sfdsk/gfs/rdsk/vol0301
```

If specifying that the update log area is to be included in the representative partition using `-A` option,

and specifying that the update log area is to be separated, the meta-data area, file data area and update log area can be independent of one another.

```
# sfxnewfs -A -L /dev/sfdsk/gfs/rdsk/vol0311 -D /dev/sfdsk/gfs/rdsk/vol0321,/dev/sfdsk/gfs/rdsk/vol0331 /dev/sfdsk/gfs/rdsk/vol0301
```

4.3.2 Customizing the GFS Local File System

The `sfxnewfs(1M)` command that is used to create a GFS Local File System contains the parameters for customizing the file system. The respective parameters are as follows:

Size of update log area

Specify the size of the update log area. The size that can be specified within a range from 5 to 100 megabytes. The default size is 1% of the size of the file system. However, if 1% of the file system size should be less than 5 megabytes, the value is set to 5 megabytes. If it should exceed 50 megabytes, the value is set to 50 megabytes.

Size of meta-data area

Specify the size of the meta-data area in the representative partition. The default is about 10% of the size of the file system. If the maximum size of file data area is specified, the default is set to about 10% of the specified size.

Maximum size of file data area

Specify the maximum total size of the file data area for the case that data partitions are added to the GFS Local File System. Data partitions can be added using the `sfxadd(1M)` command. This parameter has to be specified if data partitions are to be added. The maximum value that can be specified here is 1 terabytes.

Maximum number of partitions

Specify the maximum number of partitions that can be added to the GFS Local File System. A GFS Local File System consists of the representative partition, update log partition, and data partition(s). The maximum number that is to be specified here is the total number of these partitions. The number can range from 1 to 32. The default value is 16.

Maximum number of sharing nodes

Specify the maximum number of nodes that can directly share the specified GFS Local File System. This number can range from 1 to 16. The default value is 16.

4.4 Mounting or Unmounting a GFS Local File System

4.4.1 Mounting a GFS Local File System

To mount a file system manually, use the `mount_sfxfs(1M)` command.

To mount a file system automatically at system start, add an entry to the `/etc/vfstab` (virtual file system table) file.

For the special file name that is to be written in the `/etc/vfstab` file or specified to mount the file system, use the file name of the representative partition.

When the file system is mounted successfully, an entry is added to the `/etc/mnttab` file (table of mounted files systems).

The following shows an example of entries that have to be made in `/etc/vfstab` for mounting and for mounting a GFS Local File System.

- As special device that is to be specified in the "device to mount" and "device to fsck" fields, use the file name of the representative partition.
- Specify "sfxfs" in the "FS type" field.
- Specifying "yes" in the "mount at boot" field causes the `mountall` command to automatically mount the file system. Specifying "2" in the "fsck pass" field means that the file systems will be checked in random order.

See the description of the [mount_sfxfs\(1M\)](#) command in Appendix B for details on the options of the `mount_sfxfs(1M)` command.

- Example of coding in `/etc/vfstab`

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
/dev/sfdsk/gfs/dsk/vol0301	/dev/sfdsk/gfs/rdsk/vol0301	/mnt	sfxfs	2	yes	-

- Execution of the mount command

```
# mount -F sfxfs /dev/sfdsk/gfs/dsk/vol0301 /mnt
or
# mount /mnt
```

4.4.2 Unmounting the file system

To unmount the file system manually, use the `umount(1M)` command.

For the `umount(1M)` command, specify the mount point or the representative partition that was specified for mounting the file system.

When the file system is unmounted successfully, the corresponding entry is deleted from the `/etc/mnttab` file.

The commands in the following example unmount the file system that was mounted in the example above.

```
# umount /mnt
or
# umount /dev/sfdsk/gfs/dsk/vol0301
```

4.5 Checking the File System for Consistency (fsck)

If the file system is damaged or becomes inconsistent because of a system failure or other errors, it must be checked and recovered with the `fsck_sfxfs(1M)` command.

The GFS Local File System supports the update log function that makes high-speed recovery from a system failure or other faults possible. When the update log function is enabled, **update log replay** can recover the file system at high speed within a given time regardless of the size of the file system. Update log replay updates meta-data in the update log area that has not been updated.

When **"-o log"** is specified or the default is used, `fsck_sfxfs(1M)` recover the file system by update log replay.

If update log data is physically damaged, `fsck_sfxfs(1M)` does not perform update log replay but automatically performs a full check of the file system. However, to support a quick restart of system operation, `fsck_sfxfs(1M)` contains an option (**-o elog**) that suppresses file system recovery except for update by update log replay. If update log data is physically damaged while this option is specified, `fsck_sfxfs(1M)` quits immediately without checking and recovering the file system. This means, however, that the file system can be mounted only after it was checked and recovered by `fsck_sfxfs(1M)` afterwards. Enter `fsck_sfxfs(1M)` with the **"-o nolog"** option for performing a full check of the file system without update log replay. Recover the file system with this command, and mount it.

The command in the following example performs update log replay for recovering the file system.

```
# fsck -F sfxfs /dev/sfdsk/gfs/rdsk/vol0301
```

The command in the following example performs a full check of the file system without update log replay, and recovers the file system.

```
# fsck -F sfxfs -o nolog /dev/sfdsk/gfs/rdsk/vol0301
```

See the description of the [fsck_sfxfs\(1M\)](#) command in Appendix B for details on the options of the `fsck_sfxfs(1M)` command.

The following commands can also be used to clear invalid i-nodes and recover the file system consistently.

- `ff_sfxfs(1M)/ncheck_sfxfs(1M)`
- `clri_sfxfs(1M)`

4.5.1 Action method fsck terminates abnormally

Action method `fsck_sfxfs(1M)` terminates abnormally are explained below.

4.5.1.1 The memory demand is abnormal

The memory demand for `fsck_sfxfs(1M)` internal table is fails, the following messages are output.

```
Can't allocate memory for <xxxx>
xxxx: fsck internal table name, dup table, bta dup table, others.
```

Add the swap area.

4.5.1.2 Obtaining configuration information of GFS Local File System is fails

Obtaining configuration information of GFS Local File System is fails, the following message is output.

```
<device-name> is sfxfs_getfstab error!
```

Confirm the status of `/etc/sfxfstab` file, and is recovered.

4.5.1.3 The partition configuration information of GFS Local File System is abnormal

Each partition configuration information of GFS Local File System is disagreement, the following messages are output.

```
Can't open <device-name>: <errno>
Can't get the data of partition <device-name>
setup: Volume information error
setup: fsck quit due to unrecoverable error!
```

Recover the partition configuration information, use `sfxadm(1M)` command.
The file system is a single partition configuration, the following operations are done.

```
# sfxadm -r -a <device-name>
```

The file system is multi partition configurations, after the partition is confirmed by output of `sfxinfo(1M)`, recover from representative partition, use `sfxadm(1M)` command.

```
# sfxinfo <device-name>
# sfxadm -r <restoration origin device-name>,<device-name>,...
```

4.5.1.4 The node cannot be added

The node cannot be added, the following message are output.

```
No node volume entry for localhost,file system access forbidden!
```

Add the node, use `sfxnode(1M)` command.

4.5.1.5 File system damage which cannot be recovered

GFS Local File System data damage that cannot be recovered, the following messages are output.

```
No Meta volume information available!
No Log volume information available!
Only found <num> data volume(s), total <num> in superblock!
The data volume <device-name> is not found!
setup_data: fs_dsize < sum(dvol->sdata)!
```

It is not possible to recover. Restore from the backup if you buck up the file system.

4.6 Obtaining Various Types of File System Information

The `df_sfxfs(1M)` command can be used to obtain information about the disk block use status, free disk block status, and the number of files in the file system.

In the case of a file system that consists of multiple partitions, the `df_sfxfs(1M)` command can display information for each of the partitions.

With respect to the meta-data partition, the following information is displayed:

- i-node usage status (number of i-nodes, number of free i-nodes)
- Free extent status (maximum number of leaves and usage status)
- V-data usage status (maximum number of V-data blocks and usage status)

With respect to the data partitions, the following information is displayed:

- Total number of blocks and number of free blocks

For the update log partition., only the number of allocated blocks is displayed.

The following shows examples of using the `df_sfxfs(1M)` command for displaying the usage status of the file system.

See the description of the [df_sfxfs\(1M\)](#) command in Appendix B for details on the options of the `df_sfxfs(1M)` command.

```
# df -k /mnt
Filesystem            kbytes  used  avail  capacity  Mounted on
/dev/sfdsk/gfs/dsk/vol0301
                        113992  62664  39936   62%      /mnt

# df -F sfxfs -o v /mnt
Filesystem:/mnt
                inodes  free  vdata  free  leaf  free  Type
/dev/sfdsk/gfs/dsk/vol0301
                14248  12192  4781   4615  7128  7127  META
                kbytes  used  free  %used
/dev/sfdsk/gfs/dsk/vol0301
                113992  62664  51328   55%                DATA
                kbytes
/dev/sfdsk/gfs/dsk/vol0301
                5120                LOG

# df -F sfxfs -o v /mnt
Filesystem:/mnt
                inodes  free  vdata  free  leaf  free  Type
/dev/sfdsk/gfs/dsk/vol0301
                29686  27630  9922   9756  14848  14846  META
                kbytes  used  free  %used
/dev/sfdsk/gfs/dsk/vol0301
                105752  31832  73920   30%                DATA
                kbytes
/dev/sfdsk/gfs/dsk/vol0500
                5120                LOG
                kbytes  used  free  %used
/dev/sfdsk/gfs/dsk/vol0600
                131736  30832  100904   23%                DATA
```

4.7 Extending a GFS Local File System

An existing GFS Local File System can be simply extended by adding new partitions. Data in the files does not have to be saved or restored in this process.

Execute the `sfxadd(1M)` command to extend the file system. The file system can be extended while the file system is mounted (online extension) or unmounted (offline extension).

However, if the file system is extended while it is unmounted, the status of the file system must be normal. If an attempt is made to extend the file system while the status of the file system is not normal, the `sfxadd(1M)` command outputs the message, "partition-name is corrupted, run `fsck` manually." If this occurs, execute the `fsck_sfxfs(1M)` command to restore the status of the file system. Then, execute the `sfxadd(1M)` command again.



Note

If the file system is probably going to be extended, estimate the maximum size of the extension. Then, specify the data size of the file system when creating the file system (when the `mkfs_sfxfs(1M)` or `sfxnewfs(1M)` command is executed). The GFS Local File System records a free data area as an extent. For the free data management area, the area size is calculated automatically so that data in the specified data area can be managed when the file system is created. However, if the file system is extended, this management area can become insufficient. (Because a free data area is recorded as an extent, it can be managed if it is not fragmented. If fragmentation is severe, the free data area cannot be used because of the insufficient management area.)

If the management area seems to be insufficient for an existing file system, the free area may be rearranged for the management area of the file system by defragmenting. For information on defragmentation, see [Chapter 8, "Defragmenting GFS Local File System."](#) To check whether the management area is insufficient, specify the `-ov` option in the `df_sfxfs(1M)` command to display the total number of leaves and number of free leaves.



Note

If the `sfxadd(1M)` command has the `-pa` option specified to pre-register a partition addition, the additional partition is recognized as an existing configuration element for GFS Local File System. Then, the original file system itself is deleted. This occurs if the GFS Local File System is created as another GFS Local File System including the partition that has been pre-registered for addition to delete the addition specification. This also occurs if the `sfxadm(1M)` command with `-D` option is executed to delete a pre-registered partition from the file system.

Once a partition has been registered, including ones that have been pre-registered, only the additional partition can be deleted.

4.7.1 Adding new partition to extend file system

A file system with multiple partitions can be created for a GFS Local File System. The size of an existing file system can be extended by adding a new partition.



Note

The addition of a new partition changes the configuration information for the file system. To prepare for unexpected system failures during processing, or to operate as an alternate file system from multiple systems, execute preprocessing (execute the `sfxadd(1M)` command specified with the `-pa` option, as described below). If this preprocessing is not executed, insufficient data may be recovered in the event of a system failure during operation of an active node. As a result, normal recovery will not be possible.

If preprocessing has been executed, file system recovery by the `fsck_sxfs(1M)` command automatically extends the file system and deletes preprocessing data.

In the event of a system failure during file system extension, the `sfxinfo(1M)` command can be executed to confirm that processing has been completed. For information on `sfxinfo(1M)` command options, refer to the [sfxinfo\(1M\)](#) manual.



The maximum number of partitions for configuring a file system is set when the file system is created (if the `mkfs_sxfs(1M)` or `sfxnewfs(1M)` command is executed). Additional partitions cannot be added above this limit. The default value is 16.

4.7.1.1 Operation from one system

The following examples show how to execute the `sfxadd(1M)` command to add a data volume (`/dev/sfdsk/gfs/rdisk/vol0601`) to an existing file system (`/dev/sfdsk/gfs/rdisk/vol0301` is used as a representative partition).

For information on `sfxadd(1M)` command options, refer to the [sfxadd\(1M\)](#) manual.

- Preregistering the partition addition instruction

```
# sfxadd -pa -D /dev/sfdsk/gfs/rdisk/vol0601 /dev/sfdsk/gfs/rdisk/vol0301
```

- Adding the actual partition

```
# sfxadd -D /dev/sfdsk/gfs/rdisk/vol0601 /dev/sfdsk/gfs/rdisk/vol0301
```

4.7.1.2 Operation as alternate file system from multiple systems

The following examples show how to execute the `sfxadd(1M)` command to add a data volume (`/dev/sfdsk/gfs/rdisk/vol0601`) to an existing file system (`/dev/sfdsk/gfs/rdisk/vol0301` is used as a representative partition).

For information on `sfxadd(1M)` command options, refer to the [sfxadd\(1M\)](#) manual.

- Preregistering the partition addition instruction for all systems that use the file system

```
node1# sfxadd -pa -D /dev/sfdsk/gfs/rdisk/vol0601 /dev/sfdsk/gfs/rdisk/vol0301
node2# sfxadd -pa -D /dev/sfdsk/gfs/rdisk/vol0601 /dev/sfdsk/gfs/rdisk/vol0301
```

- Adding the actual partition on active systems

```
node1# sfxadd -D /dev/sfdsk/gfs/rdisk/vol0601 /dev/sfdsk/gfs/rdisk/vol0301
```

4.8 Displaying Partition and Node Information

The `sfxinfo(1M)` command can be used to display partition and node information for the partitions that constitute the GFS Local File System.

In the following example, the `sfxinfo(1M)` command is used for displaying partition and node information on the file system.



See

See the description of the [sfxinfo\(1M\)](#) command in Appendix B for details on the options of the `sfxinfo(1M)` command.

```
# sfxinfo /dev/sfdsk/gfs/dsk/vol0301
ID    special                                size  mount  Type
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)  25986 /mnt   META
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)  105751 /mnt   DATA
1     /dev/sfdsk/gfs/dsk/vol0302(3f40022)   5120 /mnt   LOG
2     /dev/sfdsk/gfs/dsk/vol0303(3f40023)  131736 /mnt   DATA

# sfxinfo -n /dev/sfdsk/gfs/dsk/vol0301
/dev/sfdsk/gfs/dsk/vol0301:
ID    hostID          hostname
0     807b1f69        node1
1     807b1f70        node2
```

4.9 Changing Partition or Node Information

The GFS Local File System stores partition information that constitutes the file system and node information related to the **nodes** that share the file system in each partition.

This configuration is for the purpose of supporting a multi-partition configuration and an environment in which the respective devices in individual nodes that share the file system can have different numbers.

Use the `sfxadm(1M)` command to change partition information.

Use the `sfxnode(1M)` command to change node information.

Partition and node information include the following items:

[Partition information]

- Usage status of the partition (meta-data, data, or update log)
- Partition ID
- Device number
- Partition size

[Node information]

- Host ID
- Host name

To change the file system to the HA configuration, use the `sfxadm(1M)` command to update partition information.

The following is an example procedure of updating the partition information and enabling the use of the file system that had previously been used for the normal configuration is changed to the HA configuration.

See the manual for the [sfxadd\(1M\)](#) command for details on the options of the `sfxadm(1M)` command.

```
# sfxadm /dev/sfdsk/gfs/rdsk/vol0301,/dev/sfdsk/gfs/rdsk/vol0400,/dev/sfdsk/gfs/rdsk/vol0500,/dev/sfdsk/gfs/rdsk/vol0601
```

When a node that is to share the file system is added, the node configuration information (host ID and name) has to be re-defined, as shown above. Set the number of special devices as seen from the sharing local node to the partition configuration information.

The following is an example specification for sharing the file system with the node by using the `sfxnode(1M)` command.

See the manual for the `sfxnode(1M)` command for details on the options of the `sfxnode(1M)` command.

```
# sfxnode /dev/sfdsk/gfs/rdsk/vol0301,/dev/sfdsk/gfs/rdsk/vol0400,/dev/sfdsk/gfs/rdsk/vol0500,/dev/sfdsk/gfs/rdsk/vol0601
```

Chapter 5 Backing Up and Restoring the GFS Local File System

This chapter explains how to back up and restore the GFS Local File System.

5.1 Backing Up and Restoring by sfxdump and sfxrestore

The sfxdump and sfxrestore commands are backup and restoration tools for GFS Local File Systems. The commands have the same functions as the ufsdump and ufsrestore commands in UFS. The following is the main features of the sfxdump and sfxrestore commands:

- Multiple backup and restoration
The entire file system can be backed up, Incremental backup or backup as individual files can also be executed. The file system can be restored from these backups.
- Multiple tape support
A file system can be backed up on multiple tapes. Therefore, if the contents of a file system or file to be backed up cannot be stored on one tape, the contents can be backed up on multiple tapes.
- Command line interface compatibility with ufsdump/ufsrestore
GFS Local File Systems can be backed up and restored using the same command line interface as that of the ufsdump and ufsrestore commands in UFS.



Note

Note the following about compatibility between the ufsdump/ufsrestore and sfxdump/sfxrestore commands:

- Backup data created by the ufsdump command is not compatible with that by the sfxdump command. The sfxrestore command cannot restore the data backed up by the ufsdump command. In the same way, the ufsrestore command cannot restore the data backed up by the sfxdump command.
- The sfxdump command can back up only GFS Local File Systems. In addition, the ufsdump command cannot back up GFS Local File Systems.

This section explains how to execute the sfxdump and sfxrestore commands to back up and restore GFS Local File Systems. For information on the sfxdump and sfxrestore commands, refer to the [Appendix B, "Reference Manual."](#)

5.1.1 Backing up by executing sfxdump command

This section contains an example of backing up an entire GFS Local File System specified in the /etc/vfstab file shown below. Here, /dev/rmt/0 is the backup destination tape drive.

```
% cat /etc/vfstab
#device          device          mount  FS    fck   mount  mount
#to mount        to fsck         point type  pass  at boot options
...
/dev/sfdsk/gfs/dsk/vol0a00 /dev/sfdsk/gfs/rdsk/vol0a00 /mnt sfxfs 2 yes -
...
```

The character special file in the representative partition of the GFS Local File System is /dev/sfdsk/gfs/rdsk/vol0a00. The mount point is /mnt.

1. Set for super-user.

```
% su
```

2. Unmount the GFS Local File System to be backed up, or set the system to single-user mode so that the GFS Local File System is not updated. In the following example, the GFS Local File System is unmounted.

```
# umount /mnt
```

**Note**

The GFS Local File System can be backed up while the file system is mounted. However, the GFS Local File System must be set to non-updated status. If the GFS Local File System is updated during backup, the contents of the backup may be invalid.

3. Set a tape to back up the GFS Local File System.

```
# sfxdump 0ucf /dev/rmt/0 /dev/sfdsk/gfs/rdsk/vol0a00
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Tue May 23 11:20:14 2000
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/sfdsk/gfs/rdsk/vol0a00 (kaizer1:/mnt) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 282 blocks (141KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 250 blocks (125KB) on 1 volume at 1838 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Tue May 23 11:20:14 2000
```

Here, each character of 0ucf in the first parameter has the following meaning:

- 0
This is the backup level. The number 0 means a full backup. A number between 1 and 9 means an incremental backup. In this example, the file system is backed up as an incremental backup from the latest backup at a lower level than the specified backup level.
- u
The dump record file, /etc/sfxfs_dumpdates, is updated. Specify this character for full backup or incremental backup.
- c
The contents are backed up on tape.
- f
The write destination is specified. Here, /dev/rmt/0 is specified as the write destination.

5.1.2 Restoring by executing sfxrestore command

This section explains the procedure for fully restoring the GFS Local File System from the backup tape described in the previous section.

**Note**

In this example, the backup data is restored to the GFS Local File System. If the representative partition of the GFS Local File System is different than that of the restoration destination, the /etc/vfstab file must be corrected to match the representative partition.

**Note**

If a different GFS Local File System is to be restored, note the capacity of the restoration destination. Restoration may fail because of insufficient capacity.

1. Set for super-user.

```
% su
```

2. If the GFS Local File System of the restoration destination is mounted, unmount the GFS Local File System.

```
# umount /mnt
```

3. Execute the `sfxnewfs` command to create the GFS Local File System of the restoration destination. The following example shows a single partition configuration where options are not specified.

```
# sfxnewfs /dev/sfdsk/gfs/rdsk/vol0a00
```



Note

When the GFS Local File System is created, the contents of the previously created GFS Local File System are lost.

4. Mount the GFS Local File System. If an entry exists in the `/etc/vfstab` file shown above, the GFS Local File System can be simply mounted by specifying only the mount point.

```
# mount /mnt
```

5. Move the current working directory to the mount point.

```
# cd /mnt
```

6. Insert the appropriate tape, and execute the `sfxrestore` command to restore the GFS Local File System from the tape.

```
# sfxrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump   date: Wed May 24 13:28:08 2000
Dumped from: the epoch
Level 0 dump of /mnt on kaizer1:/dev/sfdsk/gfs/dsk/vol0a00
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Warning: ./lost+found: File exists
Make node ./etc
Make node ./etc/default
...
extract file ./etc/fs/nfs/mount
extract file ./etc/fs/ufs/mount
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
```

Here, each character of `rvf` in the first parameter has the following meaning:

- r
All contents of the backup media are restored. Specify this character to restore the entire contents of the backup data.
- v
Details are displayed. The name of each file is displayed when it is restored.
- f
The read source is specified. Here, `/dev/rmt/0` is specified as the read source.



Information

To restore the incremental backup, restore from the backup tape with level 0 according to step 6. Then, repeat step 6 in ascending order based on the backup level.

7. Delete the restoresymtable file, which is a temporary file created by the sfxrestore command.

```
# rm /mnt/restoresymtable
```

8. To restore the GFS Local File System using a quota function, execute the sfxquotadm command with the -i option specified to reflect the contents of an external quotas file to an internal quotas file. For information on the sfxquotadm command, refer to the [Appendix B, "Reference Manual."](#) This step is not required if a quota function is not used.

```
# sfxquotadm -i /mnt
```

9. Move to a different directory.

```
# cd /
```

10. Unmount the newly restored GFS Local File System.

```
# umount /mnt
```

11. Execute the sfxdump command to fully back up the newly restored GFS Local File System. Insert a tape, and execute the sfxdump command to back up the GFS Local File System.

```
# sfxdump 0ucf /dev/rmt/0 /mnt
```



Note

Back up the entire GFS Local File System immediately after it has been restored. This must be done because the assignment of the i-node numbers changes after restoration by the sfxrestore command. An incremental backup cannot thereby be correctly executed from the GFS Local File System before restoration.

12. Mount the restored GFS Local File System.

```
# mount /mnt
```

5.2 Backing Up and Restoring by Executing sfxmkbkup Command

The sfxmkbkup command is a support tool for backing up and restoring by executing the dd command.

For the GFS Local File System, a file system can be created to have a multi-partition configuration. In this configuration, however, operation is complicated because the dd command must be executed several times to back up the GFS Local File System. In addition, the each backup of the multiple partitions must be restored to its correct partition at restoration. Otherwise, the file system may not be restored if the partitions are incorrect.

The sfxmkbkup command is a support tool that solves this problem. The sfxmkbkup command can be executed to create shell scripts for backing up and restoring the GFS Local File System. These commands thus enable safe backup and restoration of the GFS Local File System.

This section shows how to execute the sfxmkbkup command to back up and restore the GFS Local File System.

For information on the sfxmkbkup command, refer to [Appendix B.2.8 "sfxmkbkup\(1M\)."](#)

Super-user authority (root) is required to execute the sfxmkbkup command and run shell scripts created by the sfxmkbkup command.

Assume the GFS Local File System has the multi-partition configuration shown below.

```
# sfxinfo /dev/sfdsk/gfs/dsk/vol0301
ID   special                                size   mount   type
0    /dev/sfdsk/gfs/dsk/vol0301(3f40021)  25986  /mnt    META
0    /dev/sfdsk/gfs/dsk/vol0301(3f40021)  105751 /mnt    DATA
1    /dev/sfdsk/gfs/dsk/vol0302(3f40022)   5120   /mnt    LOG
2    /dev/sfdsk/gfs/dsk/vol0303(3f40023)  131736 /mnt    DATA
```

To generate the shell script for backing-up and restoring the GFS Local File System, specify in the sfxmkbkup command the representative partition and a device for performing the backup.

```
# sfxmkbkup /dev/sfdsk/gfs/rdisk/vol0301 /dev/rmt/0
```

After executing the sfxmkbkup command, the following two shell scripts are generated in the current working directory:

- sfxfs.bkup Shell script for performing backup
- sfxfs.rstr Shell script for restoring

If the shell script indicated below already exists in the current working directory, the shell script will not be overwritten.

If you wish to change the shell script, delete it first and then execute the sfxmkbkup command. Execute each of these scripts for backing up and restoring the GFS Local File System.

Execute backup by using the sfxfs.bkup shell script.

```
# ./sfxfs.bkup
```

Execute the restoring by using the sfxfs.rstr shell script.

The partition for the restoring must be the same one for the backing-up.

```
# ./sfxfs.rstr
```

The device size required for backup can be estimated by using the sfxmkbkup command before performing the backup. Prepare the required medium for backup in accordance with the estimate.

Specify the representative partition to execute the sfxmkbkup command.

```
# sfxmkbkup -e /dev/sfdsk/gfs/rdisk/vol0301
special          size(KB)
/dev/sfdsk/gfs/dsk/vol0301    131760
/dev/sfdsk/gfs/dsk/vol0302    131760
/dev/sfdsk/gfs/dsk/vol0303    131760
```



Note

If the `sfxadd` command is executed to add a volume, execute the `sfxmkbkup` command again to create the backup and restoration shell scripts. Also, restoration from a backup media is disabled before the volume is added. Therefore, Fujitsu recommends that the backup shell script created after the volume is added be run again for another backup.

5.3 Backing Up and Restoring by Executing dd Command

5.3.1 Backing up by executing dd command

This section explains the procedure for backing up the GFS Local File System by using dd command.

The entire file system can be backed up by using dd command. The dd command is most suitable for backing up the entire GFS Local File System, since it can back up the entire file system faster than a backup in units of single files. However, the file system cannot be restored in units of single files in this case.

When performing a backup with the procedure below, confirm first that the GFS Local File System has been unmounted.

- Backup for a single-partition configuration

Use the sfxinfo command to display partition information and check that a single partition is used (all of the respective file names are identical) is used and that it is smaller than the total tape capacity. Estimate the total capacity for the partition by adding 1 megabyte to the total of the values that are displayed in the size field (in kilobytes) .

Example: The following is an example for the case of a single-partition configuration:

```
# sfxinfo /dev/sfdsk/gfs/dsk/vol0301
ID    special                size    mount    type
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)  12626  /mnt     META
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)  113992 /mnt     DATA
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)   5120  /mnt     LOG
# dd if=/dev/sfdsk/gfs/rdsk/vol0301 of=/dev/rmt/0 bs=1024k
```

- Backup for a multi-partition configuration

The dd command performs a backup for single partitions. Therefore, when a multi-partition configuration is used, the file system must be backed up in units of single partitions. After checking the partition configuration with the sfxinfo command, back up each of the partitions in the same way as for a single-partition configuration.

Example: The following is an example of a 3-partition (META+DATA, DATA, and LOG) configuration:

```
# sfxinfo /dev/sfdsk/gfs/dsk/vol0301
ID    special                size    mount    type
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)  25986  /mnt     META
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)  105751 /mnt     DATA
1     /dev/sfdsk/gfs/dsk/vol0302(3f40022)   5120  /mnt     LOG
2     /dev/sfdsk/gfs/dsk/vol0303(3f40023)  131736 /mnt     DATA

# dd if=/dev/sfdsk/gfs/rdsk/vol0301 of=/dev/rmt/0 bs=1024k -> Tape 1
Tape replacement
# dd if=/dev/sfdsk/gfs/rdsk/vol0302 of=/dev/rmt/0 bs=1024k -> Tape 2
Tape replacement
# dd if=/dev/sfdsk/gfs/rdsk/vol0303 of=/dev/rmt/0 bs=1024k -> Tape 3
```



Note

- All the partitions making up the desired GFS Local File System must be backed up. Only backing up only some of the partitions cannot restore a file system.
- The tapes used for backing-up must be labeled so that the backed up partitions can be identified. They must then be stored as a set.

**Note**

The dd command does not support multi-volume tapes. If the size of a file system is larger than the capacity of a single tape volume, divide the file system into multiple units and back these units up separately. In this case, set bs (block size) and count (block count) to an adequate value and increase iseek (input-file offset [block count]) by the count (block count).

Example: A partition is divided into one-gigabytes units to be backed up.

```
# dd if=/dev/sfdsk/gfs/rdisk/vol0301 of=/dev/rmt/0 bs=1024k count=1024
->Tape 1
Tape replacement
# dd if=/dev/sfdsk/gfs/rdisk/vol0301 of=/dev/rmt/0 bs=1024k count=1024 iseek=1024
->Tape 2
Tape replacement
# dd if=/dev/sfdsk/gfs/rdisk/vol0301 of=/dev/rmt/0 bs=1024k count=1024 iseek=2048
->Tape 3
Tape replacement
# dd if=/dev/sfdsk/gfs/rdisk/vol0301 of=/dev/rmt/0 bs=1024k count=1024 iseek=3072
->Tape 4
...

```

5.3.2 Restoring by executing dd command

This section explains the procedure for restoring backup tapes described in the previous section.

Use the dd command to restore each of the partitions that were saved on the backup tape back to the disk.

When restoring with the procedure below, check first that the GFS Local File System has been unmounted:

- Restoring a single-partition configuration

Example: The following is an example of a single-partition configuration:

```
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdisk/vol0301 bs=1024k
```

- Restoring a multi-partition configuration

Example: The following is an example for the case of a 3-partition (META+DATA, DATA, and LOG) configuration:

```
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdisk/vol0301 bs=1024k <- Tape 1
Tape replacement
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdisk/vol0302 bs=1024k <- Tape 2
Tape replacement
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdisk/vol0303 bs=1024k <- Tape 3

```

**Note**

The dd command does not support multi-volume tapes. When restoring the file system from separate volumes of backup tape, set the bs and count parameters of the dd command by the same value as for backup. Also, specify the oseek parameter by the same value as that of iseek during backup.

Example: The following is an example for restoring a file system from separated one-gigabyte volumes of backup tape:

```
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/vol0301 bs=1024k count=1024
  <- Tape 1
Tape replacement
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/vol0301 bs=1024k count=1024 oseek=1024
  <- Tape 2
Tape replacement
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/vol0301 bs=1024k count=1024 oseek=2048
  <- Tape 3
Tape replacement
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/vol0301 bs=1024k count=1024 oseek=3072
  <- Tape 4
...
```

5.4 Change Settings after Restoring

This section explains two necessary works after restore the GFS Local File System.

When restoring to a partition other than the partition that is the source of the backup data, it is necessary to re-set up the partition information of GFS Local File System.

Moreover, when restoring the data which backed up during mounting file system, recovery of a file system is still more nearly required.

5.4.1 Re-setting of partition information

GFS Local File Systems retains partition information within the each partition so that the partition can be checked for consistency. When a GFS Local File System is restored to another partition with the following means, it is not in agreement with partition composition with actual partition information.

- Restoration to a partition different the backup origins by dd(1) command.
- The snapshot created by the snapshot function of GDS etc.

In order to protect user's data, a GFS Local File System is made not to be able to use the file system in the state that partition information and the partition composition are different like the above-mentioned and it.

When you restore a GFS Local File System to other partitions used the above-mentioned, please perform a re-setup of partition information. sfxadm(1M) performs a re-setup of partition information.

The following example shows how to perform a re-setup of partition information by sfcadm(1M) after a GFS Local File System restores to /dev/sfdsk/gfs/rdsk/vol0301, /dev/sfdsk/gfs/rdsk/vol0400, and /dev/sfdsk/gfs/rdsk/vol0500. The representative partition is /dev/sfdsk/gfs/rdsk/vol301. Please execute it by super user.

```
# sfxadm /dev/sfdsk/gfs/rdsk/vol0301,/dev/sfdsk/gfs/rdsk/vol0400,/dev/sfdsk/gfs/rdsk/vol0500
```

5.4.2 recovery of file systems

When the mounted file system was backed up, it was backed up in the "*mounting*" status.

In order to protect user's data, a GFS Local File System is made not to be able to use the file system in "*mounting*" status. Because to avoid multiple mountings of a file system at the same time. Should multiple mounting occur, file system consistency cannot be maintained. (Pay special attention to the shared disk environment accessed from multiple nodes.)

To release this status, the administrator must execute fsck_sfxfs(1M) command to clear the flag indicating "*mounting*." When a copy has been made normally, it takes several seconds to one minute to complete fsck_sfxfs(1M) execution.

The following example shows how to release "*mounting*" status by fsck_sfxfs(1M). The representative partition is /dev/sfdsk/gfs/rdsk/vol301. Please execute it by super user after checking not mount from any node.

```
# fsck -F sfxfs -y /dev/sfdsk/gfs/rdsk/vol0301
```

Chapter 6 Commands in the GFS Local File System

This chapter describes commands for use by general users, which are provided by the GFS Local File System.

6.1 Setting the Extent Attribute Information

The GFS Local File System has functions that allocate a data area beforehand to a file, extend a data area in a certain proportion, and guarantee data area continuity. The information that enables these functions is called extent attribute information. These functions enable a system offering higher performance, by ensuring data area allocation and data area continuity.

Use **sfxsetext(1)** to set the extent attribute information. However, you cannot use it to set this information in an existing file. To set the extent attribute information in an existing file, execute **sfxsetext(1)** with an alias, copy the data with **sfxcp(1)** and change the name of alias to the original with **sfxmv(1)**.

An example for using **sfxsetext(1)** is given below:

```
# sfxsetext -a 100,25 -f contig /mnt/file
```



See

For details of the **sfxsetext(1)** command, see [Appendix B.3.5 “sfxsetext\(1\).”](#)

6.2 Reference of the Extent Attribute Information

Use **sfxgettext(1)** to reference the extent attribute information of the file set by **sfxsettext(1)**. This command can also be used for files for which the extent attribute information has not been set. An example is given below:

% sfxgettext -H /mnt/file				
Blocksize(KB)	init-blocks	inc-blocks	flags	filename
8	100	25	contig	/mnt/file



See

For details of the **sfxgettext(1)** command, see [Appendix B.3.3 "sfxgettext\(1\)."](#)

6.3 Commands That Reflect the Extent Attribute Information

The extent attribute information is specific to the GFS Local File System. The standard command of **tar(1)**, **cpio(1)**, **cp(1)** or **mv(1)** cannot be used to store, restore, and copy the extent attribute information.

Accordingly, use **sfxcpio(1)**, **sfxcp(1)**, or **sfxmv(1)** that reflects the extent attribute information for storing, restoring or copying a file. The command equivalent to **tar(1)** is not provided.



See

For details about each command, see [Appendix B "Reference Manual."](#)

Chapter 7 Managing Restriction of GFS Local File System Usage

This chapter explains the quota functions used to limit file systems usage.

7.1 quota Function

The quota function is used to limit the total number of files or blocks that can be created by a general user. This total number can be limited for each user.

There are two types of limit values: soft limit value and hard limit value. The soft limit value is used to output a warning message to the user if the number of files or blocks exceeds the limit. The hard limit value is the limit where further files cannot be created or data added. In addition, if a specific interval (time limit) has elapsed after the number of files or blocks has exceeded the soft limit value, file creation or data addition is disabled.

7.2 quota Management Method for GFS Local File System

To manage the limit values, number of usage files, and number of usage blocks for each general user, the UFS quota function uses a quotas file in the root directory of the file system. In the GFS Local File System, a quotas file hidden in the file system, called an external quotas file, is used in addition to the above quotas file, called an internal quotas file. The external quotas file is used to manage the limit values for each general user. The internal quotas file is used to manage the limit values, number of usage files, and number of usage blocks for each general user.

Using two files to manage the usage of general users has the following benefits:

- Even if a command such as the `rm(1)` command deletes the external quotas file by mistake, the internal quotas file remains. As a result, quota data does not have to be set up again or matched. In addition, the external quotas file can easily be restored from the internal quotas file.
- Because the external quotas file manages only the limit values, the number of usage files or blocks is not changed if the external quotas file is backed up or restored during operation. As a result, quota data does not have to be matched.
- Update data in the internal quotas file is written to the update log area. As a result, the integrity of the file system can be quickly restored in the event of a system failure.

7.3 quota Function Commands

The table below lists the quota-related commands provided by GFS Local File System.

Command	Function outline
sfxedquota	Edits quota data for each user.
sfxquota	Displays the limit values and usage amount of users.
sfxrepquota	Displays quota data in a file system.
sfxquotaon	Enables quota function in a file system.
sfxquotaoff	Disables quota function in a file system.
sfxquotadm	quota management command



See

For information on each command, refer to [Appendix B “Reference manual.”](#)

7.4 Setup of quota Function

7.4.1 Setting up quota function

1. Create a quotas file.

With super-user authority, execute the **touch(1)** command to create a quotas file in the root directory of the GFS Local File System that uses the quota function. In addition, execute the **chmod(1)** command to set the access authority of the quotas file to [rw...]. In the following example, a quotas file is created in the mount point /sfxfs directory.

Mount the target file system in advance.

```
# cd /sfxfs
# touch quotas
# chmod 0600 quotas
```

2. Specify the user limit values.

With super-user authority, execute the **sfxedquota(1M)** command to specify the user limit values. In the following example, user limit values are specified for user1.

```
# sfxedquota user1
```

After the above command is executed, the editor displays the following contents. Specify each limit value.

```
fs /sfxfs  blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

- blocks: Specify the limit value of the number of blocks (unit: kilobytes.)
- inodes: Specify the limit value of the number of files (including the directory).
- soft: Specify the soft limit value (for outputting a warning message).
- hard: Specify the hard limit value (actual limit).

In the following example, the maximum number of blocks that can be used is specified as five megabytes (warning message output at three megabytes) and the maximum number of files that can be created is specified as 99 (warning message output at 80 files).

```
fs /sfxfs  blocks (soft = 3072, hard = 5120) inodes (soft = 80, hard = 100)
```

3. Copy the user limit values.

To copy the user limit values specified for user1 to another user, execute the **sfxedquota(1M)** command specified with the -p option. In the following example, user limit values specified for user1 are copied for user2 and user3.

```
# sfxedquota -p user1 user2 user3
-----
|           |
|           | Copy destination user name
|           | Copy source user name
```

4. Specify the time limit when the soft limit value is exceeded.

Execute the **sfxedquota(1M)** command with the -t option specified to specify the time limit. This setting applies to all users except for the super user. In the following, the time limit is specified.

```
# sfxedquota -t
```

After the above command is executed, the editor displays the following contents. Specify each time limit.

```
fs /sfxfs  blocks time limit = 0 (default),files time limit = 0 (default)
```

For default value 0, one week is displayed. Specify the unit after the value. The units that can be

specified are month, week, day, hour, min, and sec.

In the following example, the time limit for the number of blocks is specified as two weeks and the time limit for the number of files is specified as 16 days.

```
fs /sfxfs blocks time limit = 2 week, files time limit = 16 day
```

5. Enable the quota function.

With super-user authority, execute the **sfxquotaon(1M)** command to enable the quota function. In the following example, /sfxfs is enabled.

```
# sfxquotaon /sfxfs
```

6. Check the usage conditions of quota data.

- Checking the usage conditions for each user

Execute the **sfxquota(1M)** command to check the usage conditions for each user. In the following example, the usage conditions are checked for user1.

```
# sfxquota -v user1
Disk quotas for user1 (uid 1000):
Filesystem      usage  quota  limit  timeleft  files  quota  limit  timeleft
/sfxfs          2258  3072  5120                71    80    100
```

- Checking the usage conditions in units of file systems

Execute the **sfxrepquota(1M)** command to check the usage conditions in units of file systems. In the following example, the usage conditions are checked for mount point /sfxfs.

```
# sfxrepquota -v /sfxfs
/dev/sfdsk/gfs/dsk/vol1400 (/sfxfs):
                Block limits
User           used  soft  hard  timeleft  used  soft  hard  timeleft
user1  --  2258  3072  5120                71    80    100
user2  --  1345  3072  5120                38    80    100
user3  ++  3424  3072  5120  7.0 days    81    80    100  7.0 days
:
```

7. Edit the /etc/vfstab file.

Add the entries related to the target file system to the /etc/vfstab file. For more information, see [Section 7.5.1, "Enabling at system boot."](#)

The /etc/vfstab file can be edited before quotas files are created. Editing the /etc/vfstab file beforehand is effective if the quota function is set up for multiple GFS Local File Systems.

7.5 Enabling and Disabling quota Function

7.5.1 Enabling at system boot

Edit the `/etc/vfstab` file to enable the quota function when the system is booted. Add `[rq]` or `[quota]` in the `[mount options]` field on the setup line of the file system for which the quota function is to be enabled.

In the following example, the GFS Local File System of `/dev/sfdsk/gfs/dsk/vol1200` is enabled.

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
<code>/dev/sfdsk/gfs/dsk/vol1200</code>	<code>/dev/sfdsk/gfs/rdisk/vol1200</code>	<code>/sfxfs</code>	<code>sfxfs</code>	<code>2</code>	<code>yes</code>	<code>rq</code>



Note

For HA configuration (active and standby) systems, select `[no]` in the `[mount at boot]` field.

7.5.2 Enabling at command execution

The following two methods can enable the quota function when a command is executed:

- Executing the `mount(1M)` command
- Executing the `sfxquotaon(1M)` command

1. Executing the `mount(1M)` command

If the `mount(1M)` command is executed, specify the quota option in the `mount(1M)` command. In the following example, `/dev/sfdsk/gfs/dsk/vol1200` is mounted in `/sfxfs` while the quota function is concurrently enabled.

```
# mount -F sfxfs -o quota /dev/sfdsk/gfs/dsk/vol1200 /sfxfs
```



Note

The quota option and `-r` option (read only) cannot be specified at the same time.

If the `/etc/vfstab` file has been edited so that the quota function is enabled when the system is booted, specify the following to enable the quota function:

```
# mount /sfxfs
```

2. Executing the `sfxquotaon(1M)` command

Execute the `sfxquotaon(1M)` command if the target file system has already been mounted. Also, specify the mount point of the file system for which the quota function is to be enabled. In the following example, mount point `/sfxfs` is enabled.

```
# sfxquotaon /sfxfs
```

7.5.3 Disabling quota function

Execute the `sfxquotaoff(1M)` command to disable the quota function. Specify the same mount point as that specified in the `sfxquotaon(1M)` command. In the following example, mount point `/sfxfs` is disabled.

```
# sfxquotaoff /sfxfs
```



Note

If the file system is updated after the `sfxquotaoff(1M)` command is executed, quota data is matched internally if the `sfxquotaon(1M)` command is executed again. The processing time

required to match quota data increases in proportion to the number of files. Therefore, the quota function must not be disabled if possible once it has been enabled.

7.6 Notes on Operation of quota Function

7.6.1 Matching quota data

When the quota function is enabled for the first time, the number of usage files and number of usage blocks for each user is matched automatically in the GFS Local File System. The processing time required to match quota data depends on the number of files.

A log of the internal quotas file is collected if the quota function is enabled. Therefore, quota data does not have to be matched even in the event of a system failure.

If the quota function is disabled and the system subsequently updated for any reason (e.g., file creation, deletion), the numbers of files and blocks used by a user do not match the values in the internal quotas file. Therefore, additional time is required to match the data if the quota function is enabled again.

Specifically, the **sfxquotaon(1M)** or **mount(1M)** command takes additional time to return.

Therefore, the quota function must not be disabled if possible once it has been enabled.

7.6.2 external quotas file and internal quotas file

1. File structure

A 64-byte area per user is used for quota data. For the structure of each file, quota data is stored at an offset obtained by multiplying the UID value by 64 bytes.



Note

The UIDs of users must be consecutive values (if possible). If the values are not consecutive, unused areas that do not contain quota data result, causing inefficient use of space. In addition, the quota data may not be hit in the cache if access to the data is tried. Therefore, with real input-output, system performance can be reduced.

2. Storage area

The area of the internal quotas file is allocated from the V-data area. The area of the external quotas file is allocated from the data area.

The area from the V-data area is allocated in units of one kilobyte. The area from the data area is allocated in units of eight kilobytes.

For example, assuming consecutive UID values in a range of 1,000 to 2,999, the internal quotas file requires 125 kilobytes and the external quotas file requires 128 kilobytes.



Note

If the quota function is used in an existing GFS Local File System, check for sufficient free area in the V-data area.

7.6.3 Handling users for which limit values are not specified

If the quota function has been enabled, the number of usage files and number of user blocks is managed using the internal quotas file, even for users for which limit values have not been specified. As a result, limit values can be specified for the relevant user without having to match quota data.

7.6.4 Tuning parameter

The quota function has the following tuning parameter:

- Amount of quota data cached in memory (sfxfs_ncache_quota)

For the default, the value is automatically obtained based on the amount of installed memory and corresponding allocated area. However, a value can be specified using `/etc/system`.

For this value, an area of one kilobyte multiplied by the specified value is allocated to kernel memory for each file system.

Execute the statistics command **sfxstat(1M)** exclusive to the GFS Local File System to check the cache hit ratio and perform tuning accordingly.

- Setup example

```
set sfxfs:sfxfs_ncache_quota=256
```

7.6.5 Kernel messages

The kernel messages related to the quota function have message numbers in a range of 0083 to 0088. For information on the kernel messages, refer to [Appendix A “Kernel Messages.”](#)

The quota related messages are displayed on the terminal of the relevant user. The messages are not displayed on the console. The messages, however, are stored in `/var/adm/messages`.

7.7 Differences from UFS

7.7.1 quotas files

For GFS Local File System, the format of the UFS quotas file has been extended. As a result, the UFS quotas file cannot be used.

7.7.2 quotacheck(1M)

GFS Local File System does not provide a **quotacheck(1M)** command. Quota data is automatically matched in GFS Local File System as required.

7.7.3 quota(1M)

If the **quota(1M)** command of UFS is executed on an NFS client, the quota data on the NFS server can be collected and displayed.

The **sfxquota(1)** command of GFS Local File System cannot collect or display the quota data on an NFS server.

7.7.4 sfxquotadm(1M)

The **sfxquotadm(1M)** command is an exclusive GFS Local File System command. This command is executed to match the limit values of all users between the external quotas and internal quotas files. This command can also delete the internal quotas file. For more information, refer to [Appendix B.2.12 "sfxquotadm\(1M\)."](#)

Chapter 8 Defragmenting GFS Local File System

This chapter describes the defragmentation function provided by the GFS Local File System.

8.1 Defragmentation Function

If many files are repeatedly created and deleted in the GFS Local File System, the extents allocated to the files and free extents of the file system may be fragmented and dispersed. This state is referred to as fragmentation. If the file system becomes highly fragmented, the device seek time overhead may increase, and file access performance may be adversely affected.

The GFS Local File System enables defragmentation for eliminating these fragments.

Defragmentation makes contiguous the fragmented extents allocated to the files and changes the storage locations of file extents to create large free areas in the file system.

The following results can be expected by defragmentation:

- Insufficient meta-data areas in the file system can be eliminated by making the fragmented extents contiguous.
- File access efficiency can be improved by making the fragmented extents contiguous.



Note

The following operations cannot be performed if defragmenting in the mounted state (online):

- File system cannot be unmounted (**umount(1M)**).
- File system size cannot be extended (**sfxadd(1M)**).

The following operations cannot be performed if defragmenting in the unmounted state (offline):

- File system cannot be mounted (**mount(1M)**).
- File system size cannot be extended (**sfxadd(1M)**).
- File system integrity cannot be recovered (**fsck(1M)**).
- Files cannot be manipulated.

Similarly, in the unmounted state (offline), if an option enabling disk quotas in `/etc/vfstab` have been specified, defragmentation is performed but the disk quotas are not enabled. As a result, if disk quotas are enabled after offline defragmentation has been performed, quota data is updated.

8.2 Defragmentation Policies

Defragmentation has the following policies based on usage:

- simple policy
The number of extents is reduced and file access performance improved by making contiguous the extents allocated to files.
If an entire file or part of a file area is allocated again, the area is allocated from large free extents.
If free extents are sufficiently large, defragmentation can be performed faster compared with that done using other policies (adjacent and evacuate).
- adjacent policy
The number of extents is reduced and file access performance improved by making contiguous the extents allocated to files.
If free extents exist after the extents to be defragmented, the free extents are allocated contiguously.
- evacuate policy
The number of extents is reduced and file access performance improved by making contiguous the extents allocated to files.
If the extents after the extents to be defragmented are used by other files, they are allocated forcibly moved and allocated contiguously.
Although time is required to forcibly move the extents used by other files, this policy provides ideal defragmentation.
- free policy
Large free extents are created by making the fragmented free extents contiguous in the file system.
- dircomp policy
Unnecessary entries in the directory are deleted and the directory compressed.

8.3 Defragmentation Command

Execute the **sfxdefrag(1M)** command to defragment GFS Local File Systems.

To defragment the GFS Local File System by executing the **sfxdefrag(1M)** command, specify the file system defined by the block special file name of the representative partition or the mount point and policy.

The **sfxdefrag(1M)** command can be executed even if the target file system is in the mounted (online) or unmounted (offline) state. If defragmenting in the unmounted state, an entry to `/etc/vfstab` must be specified in advance.



Note

If the **sfxdefrag(1M)** command is executed, the mounted (online) or unmounted (offline) state of the file system must be specified.

An error occurs if offline (-f option) is specified in the mounted state (online) or online (-f option) is not specified in the unmounted state (offline).

The following are usage examples of the **sfxdefrag(1M)** command:

- Improving the file system of mount point `/sfxfs` using simple policy

```
# sfxdefrag /sfxfs simple
```

- Improving the file system of block special file `/dev/sfdsk/gfs/dsk/vol1100` in the unmounted state using adjacent policy

```
# sfxdefrag -f /dev/sfdsk/gfs/dsk/vol1100 adjacent
```

- Improving directory `target_dir` under mount point `/sfxfs` using evacuate policy

```
# sfxdefrag /sfxfs evacuate target_dir
```

- Improving the file system of block special file `/dev/sfdsk/gfs/dsk/vol1100` using free policy

```
# sfxdefrag /dev/sfdsk/gfs/dsk/vol1100 free
```

- Improving directory `/sfxfs/target_dir` of the file system of block special file `/dev/sfdsk/gfs/dsk/vol1100` using dircomp policy

```
# sfxdefrag /dev/sfdsk/gfs/dsk/vol1100 dircomp target_dir
```

Chapter 9 Performance and Tuning

This chapter explains how to optimize the GFS Local File System and how to use it effectively with various utilities.

9.1 Update Logs

In GFS Local File System, an update log that contains information on changes in the file system structure is written to the disk in order to support a high-speed recovery of the file system. This seems to suggest that GFS Local File System require more I/O operations than conventional file systems. However, the information on changes is actually written collectively and updating the entries on changes is delayed in order to reduce the volume of file data I/O. This makes it possible to achieve file system performance that is equal to or higher than the performance of a file system that uses no log file. In order to achieve this, check the update log characteristics and actual accesses of the file system with respect to the following questions:

1. Is the capacity of the update log area sufficient?
2. Is the update log area too large?
3. Is it better to allocate the update log area to another partition?

How to check the above questions and what actions are to be taken depending on the outcome is described below:

When the settings for the update log area are changed, the file system will be re-created with `sfxnewfs(1M)`.

9.1.1 Is the capacity of the update log area sufficient?

In the GFS Local File System, writing data from the **update log buffers** in memory to the update log area on the disk is delayed as long as possible. The latest update log information for the same meta-data is written in order to avoid writing outdated log information to the update log area. To effectively delay the writing of log information, a sufficiently large update log area must be reserved on the disk.

As a general rule, reserve an update log area of 1-megabytes for a file system of 100 megabytes (`sfxnewfs(1M)` default).

The GFS Local File System records the actual and the average number of updates by way of the meta-data on the disk in a set of unique statistical information.

If file system performance is low and the average number of updates is close to 1, try to expand the update log area.

9.1.2 Is the update log area too large?

If a system failure occurs, GFS Local File System performs a high-speed recovery of the corresponding file system based on the information in the update log area. Specifically, it reads update log information and reflects updates in the meta-data. The capability for high-speed file system recovery depends on the size of the update log area.

When the time that is allowed for file system recovery is limited, the appropriate capacity for the log area has to be estimated.

It can be estimated that 5 seconds correspond to one megabyte of the update log area.

In the standard `sfxnewfs(1M)` setting, the update log area size is set to less than 50 megabytes regardless of the file system capacity.

9.1.3 Is it better to allocate the update log area to another partition?

GFS Local File System attempts to avoid decreases in file system performance by writing to the update log area. However, I/O bottlenecks may nevertheless occur depending on operation states. In this case, create a partition for the update log area on a disk with relatively low I/O load and separate the I/O operations for the file system from those for the update log area in order to improve the performance of the file system.

When checking whether writing to the update log area causes an I/O bottleneck, check with the GFS Local File System-specific statistical command the number of I/O accesses per second to the update log area and the average waiting time. The amount of I/O accesses for the entire file system can also be investigated with this command.

9.2 Performance and Tuning

This section describes how to use the utilities to optimize a GFS Local File System for most efficient use.

9.2.1 Statistical information of file system

The commands for checking a GFS Local File System include a GFS Local File System-specific command, `sfxstat(1M)`, as well as the standard Solaris OE commands, `iostat(1M)`, `vmstat(1M)`, and `sar(1M)`. These commands enable the user to determine the status of system activity and troubleshoot performance problems.

This section describes the use of the `sfxstat(1M)` command and the information displayed by the command. The method of tuning performance by setting the values of the items is explained in the next section.

Format: `sfxstat [-abcdfilmA] [interval [count]] [mount_point ...]`

After reporting statistical information accumulated since the file system was mounted, the `sfxstat(1M)` command reports differences from previous information at specified intervals.

If `interval` is specified, this command reports statistical information at intervals of `interval` seconds. If `count` is specified, this command repeats the report of statistical information for the number of times indicated by `count`.

If `mount_point` is specified, this command reports on only the specified file system. Otherwise, this command reports information on all GFS Local File Systems currently mounted.

The meanings of the optional parameters are listed below. Note that all information is reported if the options are omitted:

- a
Reports information on assigned extent manipulation.
The retrieval request count, acquisition count, number of acquired blocks, unsuccessful acquisition count, release count, number of released blocks, released block successful connection count, free extent management division occurrence count, and free extent management link occurrence count are reported.
- b
Reports the meta-data management buf acquisition count, dynamic buf acquisition count, and sleep count during a request to acquire buf.
- c
Reports the access count for the meta-data cache, cache hit count, update count, read count for each meta-data type, write count, write time, and average time of write processing.
- d
Reports information on direct I/O.
The direct I/O read count, write count, number of blocks read, number of blocks written, normal I/O read count, normal I/O write count, page flash count after read, and page flash count after write are reported.
- f
Reports information on free extent manipulation.
The retrieval request count, acquisition count, number of acquired blocks, unsuccessful acquisition count, release count, number of released blocks, released block successful connection count, free extent management division occurrence count, and free extent management link occurrence count are reported.
- i

Reports the actual data area read I/O count, actual write I/O count, number of actual I/O blocks read, and number of actual I/O blocks written.

- l

Reports information on the log daemon.

The log daemon start count, sync request count, fsync request count, log buffer full count, request count from meta-data write daemon, sleep count for excessive transactions, sleep count for excessively large preceding transaction, log daemon I/O issue count, write time, number of write blocks, number of valid blocks, log-oriented processing request count, log-oriented processing parallel operation count, average parallel operation count, large buffer use count, and subtransaction occurrence count are reported.

- m

Reports information on the meta-data daemon.

The meta-data daemon start count, I/O wait buf count for meta-data daemon, disk full-driven start count, and meta-data cache full-driven start count are reported.

- A

Reports all information.



Note

Some of the information displayed by the sar(1) command does not contain the status of GFS Local File System activity. The following items are not included:

- Cache hit rates (bread, bwrit, lread, lwrit, rcache, wcache)
- UFS-specific information (iget, dirblk, ufs_ipf)

9.2.2 Performance Verification

Each GFS Local File System is divided into the following internal components:

- a. File system operation component
- b. File operation component
- c. File data management component
- d. Meta-data cache management component
- e. Meta-data disk update component (meta-data write daemon: Meta-data daemon)
- f. Update log memory recording component
- g. Disk update component of update log (log write daemon: Log daemon)

While monitoring the performance of the file system, check for performance bottlenecks in each of these components.

a) File system operation component

The file system operation component is responsible for operations such as mount, unmount, and sync.

The mount and unmount operations have no relationship with general performance problems. In GFS Local File System, the sfxstat(1M) command can be used to see the number of times the log write daemon is requested to reflect disk updates at sync request. If the user explicitly specifies frequent execution of the sync command or sync(2) system call, review the number of times the call is issued if it is primarily executed to request start of the log write daemon and log write I/O is the bottleneck.

b) File operation component

For the file operation component, the number of times read and write system calls are issued in the entire system can be determined by specifying the -c option of the sar command.

Specific individual information cannot be obtained for other system calls.

c) File data management component

For the file data management component, the number of times read and write system calls are issued and the number of blocks (in units of 8 kilobytes) can be determined by specifying the `-i` option of the `sfxstat(1M)` command.

GFS Local File System assigns the data in the file data part so that reading or writing is as contiguous as possible. The area may not be contiguous, however, depending on the area acquisition status.

If the value of `readblks/readcnts` or `writeblks/writecnts` is near 1, allocation of the data part may be fragmented, lowering the performance of sequential read or write operations.

If possible, the area should be defragmented, or the file should be copied by a command such as `cp`, the file deleted, and the data copied back to a new file. This method enables contiguous reallocation of the area.

If the direct I/O function is used, the `-d` option of the `sfxstat(1M)` command can be specified to confirm the direct I/O read count, write count, number of read blocks, number of write blocks, number of times a read becomes normal I/O, number of times a write becomes normal I/O, number of times page flash is executed after read, and number of times page flash is executed after a write.

If the direct I/O function is used, two methods of tuning are assumed: Improvements on the application side such as changing I/O size and boundary condition, and improvements by defragmenting the area to be contiguous as stated above.

d) Meta-data cache management component

For the meta-data cache management component, the access count, cache hit count, update count for each cache type of meta-data, and read count for each type of meta-data can be confirmed by the `-c` option of the `sfxstat(1M)` command.

If the ratio of cache hit counts to meta-data cache access counts is low, the cache area for the system meta-data might be insufficient.

The current GFS Local File System allows the user to set the following items:

- Number of caches for extent area management (LEAF, NODE)
- Number of caches for directory data (DIR)
- Number of caches for disk i-nodes (DINODE)
- Number of caches for quota data (QUOTA)

If the cache hit rate for any of the above areas ($\text{cachehit/access} * 100$) is less than 30%, performance may be improved by resetting the following tuning parameters.

e) Meta-data disk update component (meta-data write daemon)

The write count, write time, and average write time for each meta-data cache can be confirmed by the `-c` option of the `sfxstat(1M)` command.

If the update count per second (`mod`) is 100 or more and the write count (W-I/O) ratio is near 1, the holding time for the actual disk update after update may be restricted by factors such as the update log area.

First, if the `mlogfull` item exceeds 10 per second for more than one minute, the performance may be improved by using the `-m` option of the `sfxstat(1M)` command to take the following actions:

- Expand the update log area (if the update log area is in another partition).
- Assign the update log area to another partition (if the update log area is in the representative partition).

To expand the update log area, use the `sfxadm(1M)` command. To assign the update log area to another partition, re-create the file system. Refer to the command reference for details of the `sfxadm(1M)`, `mkfs_sxfs(1M)` and `sfxnewfs(1M)` commands.

If the update log area is not causing the problem, the cache area for meta-data in the system is assumed insufficient. The area full count for each meta-data cache is displayed for `METAFULL`, which is displayed by the `-m` option of the `sfxstat` command.

The current GFS Local File System enables the user to set the following items:

- Number of caches for extent area management (LEAF, NODE)
- Number of caches for directory data (DIR)

- Number of caches for disk i-nodes (DINODE)
- Number of caches for quota data (QUOTA)

If any of the above areas frequently goes into METAFULL state (an area shortage occurs one or more times), the performance may be improved by resetting the following tuning parameters.

f) Update log memory recording component

The following items of the update log memory recording component can be confirmed by the -l option of the sfxstat(1M) command:

- Log buffer full count (lbfll)
- Sleep count for excessive transactions (ltfll)
- Log disk full sleep count (ldfll)
- Sleep count for large preceding transaction (logbig)
- Sleep count for excessively large preceding transaction (logbigx)
- Log-oriented processing request count (ltrans), log-oriented processing parallel operation (lpara)
- Average parallel operation count (lave), large buffer use count (lbigbuf)
- Subtransaction occurrence count (lsubtrn)

If ldfll frequently indicates a value other than 0, performance may be improved by reviewing the disk update component. This is discussed in the next few paragraphs.

g) Disk update component of update log (log write daemon)

The following items on the update log disk update component (log write daemon) can be confirmed by the -l option of the sfxstat(1M) command:

- Log daemon start count (lwakeup)
- Log daemon I/O issue count (lnwrite)
- Write time (lms)
- Number of write blocks (lnblks)
- Number of valid blocks (loccupy)

If the number of times the log daemon issues an I/O request exceeds 10 per second and the average I/O time (lms/lnwrite) exceeds 100 ms, performance may be improved by assigning the update log area to another independent partition.

To assign the update log area to another partition, re-create the file system. Refer to the command reference for details of the mkfs_sfxfs(1M) and sfxnews(1M) commands.

If ldfll frequently indicates a value other than 0, performance may be improved by expanding the update log area (for the update log area in another partition). To expand the update log area, unmount the file system currently being used and specify an appropriate log size by the -L option of the sfxadm(1M) command. Refer to the command reference for details of the sfxadm(1M) command.

9.2.3 Tuning Parameters

This section discusses the tuning parameters of the GFS Local File System.

9.2.3.1 Tuning the size of the meta-data cache area

The GFS Local File System enables tuning of the following items:

- Amount of free extent management data to be cached in memory (sfxfs_ncache_extent)
- Amount of directory data to be cached in memory (sfxfs_ncache_directory)
- Amount of i-nodes on the disk to be cached in memory (sfxfs_ncache_inode)
- Amount of quota data to be cached in memory (sfxfs_ncache_quota)

These settings can be changed by `/etc/system` or options (mount options field of `/etc/vfstab` allowed) at `mount(1M)`.

If the settings are changed by `/etc/system`, an area of the specified value is secured on the kernel memory for each file system regardless of the sizes of the file systems.

If the settings are specified by option at `mount(1M)`, an area of the specified value is secured on the kernel memory for the corresponding file system (the values specified by `/etc/system` are ignored)

Therefore, to tune the items according to the amounts of individual file system resources, the setting values should be changed by option at `mount(1M)`. See `mount_sfxfs(1M)` for details.

These settings are used to secure an area whose size is determined by multiplying the specified value by 1 kilobyte for extent management and directory data plus an area whose size is determined by multiplying the specified value by 4 kilobytes for an i-node on the kernel memory for each file system.

Use the GFS Local File System -dedicated command, `sfxstat(1M)`, to check the extent, directory, and hit rate of the i-node cache on the disk for tuning. Refer to the command reference for details of the `sfxstat(1M)` command.



If the size of a directory exceeds the size of the GFS Local File System directory cache, access to and update of entry data of the corresponding directory become extremely slow. In an environment containing a large directory, Fujitsu strongly recommends that the size of the cache for the directory be larger than the expected size of the directory.

9.2.3.2 Change method

There are two methods of changing the tuning parameters. One method is to edit `/etc/system` and the other method is to specify options in `mount(1M)`.

- Specifying options in `mount(1M)`
`mount(1)` has the following options for changing the values of the tuning parameters. Specify appropriate values for the options in `mount(1M)`:
 - `textent=<val>`
Amount of extent management data to be cached in memory, in 1-kilobyte units.
 - `tdir=<val>`
Amount of directory data to be cached in memory, in 1-kilobyte units.
 - `tinode=<val>`
Amount of i-nodes on the disk to be cached in memory, in 4 kilobyte units.
 - `tquota=<val>`
Amount of quota information to be cached in memory, in 1-kilobyte units.

The following is an example of specification:
 Example of specification

```
# mount -F sfxfs -o textent=512,tidir=1024,tinode=512,tquota=256 /dev/dsk/c2t1d0s0 /sfxfs
```

- Editing the /etc/system file
Use a text editor such as vi to edit the /etc/system file to set each tuning parameter to an appropriate value.

The following is an example of settings:
Example of settings

```
set sfxfs:sfxfs_ncache_extent=512
set sfxfs:sfxfs_ncache_directory=1024
set sfxfs:sfxfs_ncache_inode=512
set sfxfs:sfxfs_ncache_quota=256
```



Note

Since the values are automatically set according to the amount of memory installed in the system, do not set up the tuning parameters unless a performance problem occurs. Be careful when modifying the /etc/system file because its settings are important to the system.

9.2.3.3 Range of values

The minimum and maximum values of each tuning parameter are listed below.

Tuning parameter	Minimum value	Maximum value
sfxfs_ncache_extent	192	32768
sfxfs_ncache_directory	512	16384
sfxfs_ncache_inode	128	4096
sfxfs_ncache_quota	64	4096

The values of the system settings can be confirmed by executing the crash command with one or more GFS Local File Systems mounted.

The method of confirmation is as follows:

```
# crash <<!
od -d sfxfs_ncache_extent
od -d sfxfs_ncache_directory
od -d sfxfs_ncache_inode
od -d sfxfs_ncache_quota
!
```

The values are displayed as in the following example:


```
> dumpfile = /dev/mem, namelist = /dev/ksyms, outfile = stdout
> 61bf37e8: 0000000576
> 61bf37e4: 0000002048
> 61bf3800: 0000000256
> 61bf380c: 0000000128
```

9.2.3.4 Tuning the upper limit of the size of the meta-data cache area

When the following meta-data cache area for each file system used all, these area is grow automatically.

- disk inode area
- V-data area (directory information, extent information (node/leaf))
- allocate information of file data

There is a maximum value in the size of the meta-data cache area in the whole node, and it is made not to occupy kernel memory area by automatic expansion. The meta-data cache area size upper limit is 64MB.

When grow size arrived at the upper limit size of meta-data cache area during use, the following messages are indicated. At this time, automatic extension of a meta-data cache area is not performed, but a meta-data cache area is replaced and used.

```
sfxfs:0120: mntpoint: meta data cache grow failed because of size limitation
```

When implementation memory is large, grow upper limit size of meta-data cache area can be change greatly.

The meta-data cache area grow limit size sets it in a `/etc/system` file by Kbytes unit.

Please note that the memory is used up to the setting value when it tunes up the meta-data cache area grow limit size.

Please specify in the range exceeding the memories which deducted the kernel memories used by the system, and memories which application needs from real memories not to press the memories required for the whole node to carry out normal operation.

The example of setting the meta-data cache area grow limit size to 128MB is shown as follows.

```
set sfxfs:sfxfs_metagrow_limit=0x20000
```

9.3 Monitoring the Free Area

If file system resources become insufficient, new files cannot be created and existing files cannot be extended. For GFS Local File System, it is recommended to monitor whether there is sufficiently much free space in the following areas:

- i-node area
- file data area
- V-data area
- Extent management information area

Information on the i-node area, data, V-data, and extent management information areas can be displayed with the corresponding **df_sfxfs(1M)** option.

Even if the available free data area becomes insufficient, GFS Local File System can extend the corresponding file system by unmounting it first and adding a new, free partition to it. For details, see [Appendix B.2.1, "sfxadd\(1M\)."](#)

If the free space in i-node, V-data, or extent management information area become insufficient, back up the corresponding file system temporarily, create a new file system area by changing the **sfxnewfs(1M)** options or preparing a larger partition, then restore the file system from the backup tape.

The management area for V-data and extent management information areas may increase because of area fragmentation. In this case, the shortage can be resolved by simply backing up the corresponding file system, re-creating the file system area, and restoring the file system.

Chapter 10 Using GFS Local File System in RMS cluster Operation

This chapter explains how to use GFS Local File System in RMS cluster operation.

10.1 Before using a GFS Local File System in RMS cluster operation

When using GFS Local File System in RMS cluster, set up the environment to use RMS with each node beforehand.

If use RMS, the following environment setting is necessary.

- Registration of RMS host name to /etc/hosts with each node
RMS recognizes each node constituting a RMS cluster with host name to be accompanied with the string "RMS" like the following setting.

192.168.0.1	sunnyRMS
192.168.0.2	moonyRMS

- Setting of cluster.config file or .rhosts file with each node
In order to distribute RMS composition definition information to each node, the userApplication Configuration Wizard uses CF remote services or rcp (1).
For this reason, a setup of the RMS host name to a setup, .rhost, or hosts.equiv of a cluster.config file is needed.



[See](#)

For details on the cluster operation, refer to "security of CF" of a "*PRIMECLUSTER Cluster Foundation Configuration and Administration Guide*."

10.2 When Using a GFS Local File System as an Alternate cluster file system

GFS Local File System has the name and host-id of each node that is using the file system, and information on each device on the file system.

When the file system is used as an alternate file system, information on all **nodes** that are using the file system must be stored in the file system.

Specifically, execute the `sfxnode` command on a node other than the one on which the file system was created by the `sfxnewfs` command.

Before executing the `sfxnode` command, unmount the file system from the node using the file system.

Because the file system cannot be used from two nodes simultaneously, specify "mount at boot" as "no" in `/etc/vfstab`.



See

For details on the setup of the cluster system, refer to "*PRIMECLUSTER Configuration and Administration Guide*" and "*PRIMECLUSTER RMS Configuration and Administration Guide*."

In the following example, the GFS Local File System created on node `sunny` is set up as an alternate file system on node `moony`.

- Confirm that the file system that is to be added as an alternate file system is not currently mounted.

```
sunny # mount | grep /dev/sfdsk/gfs/dsk/vol0301
```

Confirm that no data is currently output.

If the file system is mounted, perform unmount processing.

- Set up host information so that the file system can be used from the node "moony."

```
moony # sfxnode /dev/sfdsk/gfs/rdsk/vol0301
```

- Confirm that both nodes have been set.

```
moony # sfxinfo -n /dev/sfdsk/gfs/dsk/vol0301
/dev/sfdsk/gfs/dsk/vol0301
ID      hostID      hostname
0       809fxxxx   sunny
1       809fyyyy   moony
sunny # sfxinfo -n /dev/sfdsk/gfs/dsk/vol0301
/dev/sfdsk/gfs/dsk/vol0301
ID      hostID      hostname
0       809fxxxx   sunny
1       809fyyyy   moony
```

10.3 Setup flow for using GFS Local File System for the GDS volume as an alternate file system

The following table shows the setup flow for using GFS Local File System for the GDS volume on the RMS cluster as an alternate file system.

Common operation (Performed on an active node)	Active node	Standby node
-	Creating the GDS local file system (See Procedure 1 in Section 10.4) Setting up the alternate file system (See Procedure 2 in Section 10.4) Executing sfxnode command	-
-	-	Setting up the alternate file system (See Procedure 2 in Section 10.4) Executing sfxadm command
-	Adding the GFS local file system to /etc/vfstab (See Procedure 3 in Section 10.4)	Adding the GFS local file system to /etc/vfstab (See Procedure 3 in Section 10.4)
Defining the RMS configuration (mount point for alternate file system and switchover definition for RMS application) (See Procedure 4 in Section 10.4)	-	-
-	Starting the RMS, and checking the settings (See Procedure 5 in Section 10.6)	Starting the RMS, and checking the settings (See Procedure 5 in Section 10.6)

10.4 Setup procedure for using GFS Local File System for the GDS volume as an alternate file system

This section explains the procedure for creating the GFS Local File System on the GDS volume on the RMS cluster and using it as an alternate file system.



For details on setup, refer to "*PRIMECLUSTER Configuration and Administration Guide*," "*PRIMECLUSTER RMS Configuration and Administration Guide*," and "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

In addition to commands, a GDS volume can be also created by using the GDS Management View (GUI). For details to setup the GDS, refer to "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

Procedure 1 Creating a GFS Local File System (Active node)

Create a GFS Local File System on the volume from the node in which an active node exists. To create a GFS Local File System, use `sfxnewfs` command.

Procedure 2 Setting up the alternate file system (Active node, Standby node)

By the following methods, the alternate file system is set up.

That procedure is explained with `sunny` as that host name of an active node, and `moony` as host name of a standby node.

- At `moony` side, confirm host ID of `moony`.

```
moony # hostid
809fyyyy
```

- At `sunny` side, to set up host name and host ID of `moony`, execute `sfxnode` command. In `sfxnode` command, specify the representative partition of GFS Local File System.

```
sunny # sfxnode -a -n moony -i 809fyyyy /dev/sfdsk/gfs/rdsk/vol1000
```

- At `moony` side, to register device information to GFS Local File System, execute `sfxadm` command.

In `sfxadm` command, specify all partitions constituting GFS Local File System with device name seen from `moony` side, and first of all specify the representative partition of GFS Local File System.

```
moony # sfxadm -A /dev/sfdsk/gfs/rdsk/vol1000,/dev/sfdsk/gfs/rdsk/vol1100
```

Procedure 3 Adding a GFS Local File System to `/etc/vfstab` (Active node, Standby node)

Add a mount information of a GFS Local File System to `/etc/vfstab`.

Add the mount information prefixed by "`#RMS#`" or "`#RMS:application#`" to `/etc/vfstab` on the node in which the file system was created. At this time, specify "no" for the "mount at boot" specification in the added mount information.

In addition to above, "`application`" means the RMS application name created by later the userApplication Configuration Wizard.

```
#RMS#/dev/sfdsk/gfs/dsk/vol1000 /dev/sfdsk/gfs/rdsk/vol1000 /sxfxs sxfxs 2 no -
```

Procedure 4 Setting a mount point resource and a switchover definition for user applications as the RMS configuration (Active node, Standby node)

By the following methods, the RMS configuration is set up using the userApplication Configuration

Wizard.



For details on setup, refer to "*PRIMECLUSTER Configuration and Administration Guide*" and "*PRIMECLUSTER RMS Configuration and Administration Guide*."

Procedure 5 Checking the settings (Active node, Standby node)

Start the RMS, and check the following points to confirm configuration of /etc/vfstab.

- In the active node, the GFS Local File System can be mounted as described in /etc/vfstab.
- The operation side is switched when the active node is stopped. And the file system is mounted correctly to the switched node.



If RMS application is failed in the above procedures, it may have occurred for the following reasons. Take the action described below.

- Type of the file system that is specified in the /etc/vfstab is incorrect.
Action: Correct /etc/vfstab.
- After the GFS Local File System was created, the alternate file system was not set up.
Action: Set up the alternate file system referring to Procedure 2.

Still more, to set up an alternate file system in the cascade operation environment, perform the above steps on standby nodes.

10.5 Setup flow for adding a data partition in the GFS Local File System of the alternate file system

The following table shows the flow for adding a data partition in the GFS Local File System of the alternate file system.

Common operation (Performed on an active node)	Active node	Standby node
-	Stopping the RMS (See Procedure 1 in Section 10.6)	Stopping the RMS (See Procedure 1 in Section 10.6)
-	Adding a data partition to a GFS local file system (See Procedure 2 in Section 10.6) Setting up the alternate file system (See Procedure 3 in Section 10.6) Executing sfxnode command	-
-	-	Setting up the alternate file system (See Procedure 3 in Section 10.6) Executing sfxadm command
Changing the RMS configuration (Adding a disk class of GDS volume) (See Procedure 4 in Section 10.6)	-	-
-	Starting the RMS, and checking the settings (See Procedure 5 in Section 10.6)	Starting the RMS, and checking the settings (See Procedure 5 in Section 10.6)

10.6 Setup procedure for adding a data partition to the GFS Local File System of the alternate file system

This section explains the procedure for adding a data partition to the GFS Local File System of the alternate file system.



See

For details on setup, refer to "*PRIMECLUSTER Configuration and Administration Guide*," "*PRIMECLUSTER RMS Configuration and Administration Guide*," and "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

Procedure 1 Stopping the RMS (Active node, Standby node)

When the RMS is active, stop the RMS.

Procedure 2 Adding a data partition to the GFS Local File System (Active node)

Add a data partition from the node in which an active node exists to the GFS Local File System with the `sfxadd` command.

```
sunny # sfxadd -D /dev/sfdsk/gfs/rdsk/vol1200 /dev/sfdsk/gfs/rdsk/vol1000
```

Procedure 3 Setting up the alternate file system (Active node, Standby node)

By the following methods, the alternate file system is set up.

That procedure is explained with `sunny` as that host name of an active node, and `moony` as host name of a standby node.

- At `moony` side, confirm host ID of `moony`.

```
moony # hostid  
809fyyyy
```

- At `sunny` side, to set up host name and host ID of `moony`, execute `sfxnode` command.
In `sfxnode` command, specify the representative partition of GFS Local File System created by `sfxnewfs` command.

```
sunny # sfxnode -a -n moony -i 809fyyyy /dev/sfdsk/gfs/rdsk/vol1000
```

- At `moony` side, to register device information to GFS Local File System, execute `sfxadm` command.

In `sfxadm` command, specify all partitions constituting GFS Local File System with device name seen from `moony` side, and first of all specify the representative partition of GFS Local File System.

```
moony # sfxadm -A /dev/sfdsk/gfs/rdsk/vol1000,/dev/sfdsk/gfs/rdsk/vol1100,/dev/sfdsk/gfs/rdsk/v  
ol1200
```

Procedure 4 Setting a mount point resource and a switchover definition for user applications as the RMS configuration (Active node, Standby node)

By the following methods, the RMS configuration is set up using the userApplication Configuration Wizard if a GDS volume in other disk class is added.



See

For details on setup, refer to "*PRIMECLUSTER Configuration and Administration Guide*" and "*PRIMECLUSTER RMS Configuration and Administration Guide*."

Procedure 5 Checking the settings (Active node, Standby node)

Restart the RMS, and check the following points to confirm the setting of the RMS configuration.

- In the active node, the GFS Local File System can be mounted as described in the /etc/vfstab file.
- The operation side is switched when the active node is stopped. And the file system is mounted correctly to the switched node.



Note

If RMS application is failed in the above procedures, it may occur for the following reasons. Take the action described below.

- After the data partition of the GFS Local File System was added, the alternate file system was not set up.

Action: Set up the alternate file system referring to Procedure 3.

Still more, to add a file data partition to an alternate file system in the cascade operation environment, perform the above standby nodes' steps on standby nodes.

10.7 Using GFS Local File System as NFS Server in Stand-by Operation

The following conditions must be satisfied to use the GFS Local File System as an alternate cluster file system and the NFS server in stand-by operation:

1. The device numbers of all partitions composing the GFS Local File System must match in both nodes used for stand-by operation.
2. The specification for forcibly loading the GFS Local File System module must be able to be registered in /etc/system for both nodes.
3. The GFS Local File System module forcibly loaded on both nodes must have the same slot number.

First, execute the ls command to confirm that the device major and minor numbers on each node match. If the numbers do not match, check the cluster environment again.

Next, as explained in the previous section, setup the GFS Local File System for use as an alternate cluster file system. Then, execute the vi command to add the following statement to the /etc/system file at the position where "forceload" is specified.

```
forceload: fs/sfxfs
```

After adding the above statement to the files on both nodes, reboot the system.

Finally, execute the following command to confirm that the slot number of kernel module sfxfs of GFS Local File System is the same on both nodes. If the numbers do not match, check the forceload statement on both nodes.

```
# crash <<!  
vfssw  
!
```

The above command leads to an output of the following list:

```
dumpfile = /dev/mem, namelist = /dev/ksyms, outfile = stdout  
> FILE SYSTEM SWITCH TABLE SIZE = XX  
SLOT  NAME    FLAGS  
  1  specfs    0  
  2   ufs     0  
  3  fifofs    0  
  4  namefs    0  
  5   proc    0  
  6   s5fs    0  
  7   nfs     0  
  8  sfxfs     0  
....
```

In the above example, the slot number is 8.

Chapter 11 Changing Node information for GFS

Local File System

This chapter explains the action to be taken when node name or host ID of a node using GFS Local File System changed.

11.1 When the Name of a Node Using GFS Local File System is Changed

GFS Local File System stores the name of each node that uses the file system, the respective host-ids, and information on all devices in the file system. When a node name is changed after the GFS Local File System has been created, the change must be incorporated at the required places in the GFS Local File System. Change the respective settings with the `sfxnode` command; otherwise, the GFS Local File System cannot be mounted.

The following steps have to taken when the node name is changed from "sunny" to "moony":

1. Check the current settings.

```
# sfxinfo -n /dev/sfdsk/gfs/dsk/vol0301
/dev/sfdsk/gfs/dsk/vol0301:
ID      hostID      hostname
0       807exxxx   sunny
```

2. Set the new node information.

```
# sfxnode /dev/sfdsk/gfs/rdsk/vol0301
hostid: 807exxxx      nodename: moony
```

3. Check the node information.

```
# sfxinfo -n /dev/sfdsk/gfs/dsk/vo0301
/dev/sfdsk/gfs/dsk/vol0301:
ID      hostID      hostname
0       807exxxx   moony
```

11.2 When the Motherboard of a Node Using GFS Local File System is Replaced

GFS Local File System stores the name of each node that uses the file system, the respective host-ids, and information on all devices in the file system. When a motherboard of node is replaced after the GFS Local File System has been created, the change must be incorporated at the required places in the GFS Local File System. Change the respective settings with the `sfxnode` command; otherwise, the GFS Local File System cannot be mounted.

About the procedure, see the [section 11.1 “When the Name of a Node Using GFS Local File System is Changed.”](#)

Appendix A Kernel Messages

This appendix explains the kernel messages for the GFS Local File System.

A.1 Diagnostics

A.1.1 NOTICE: sfxfs:0001: *mount_point*: file system full

Explanation

The data area in the file system (*mount_point*) is full. (For common users, this refers to cases where the remaining free space of the data area is smaller than "minfree")

Response

Increase the size of the file system, delete unnecessary files, or move files to another file system that has enough free space in the data area. Use the `sfxadd(1M)` to increase the size of the file system.

A.1.2 NOTICE: sfxfs:0002: *mount_point*: out of inodes

Explanation

The i-node area in the file system (*mount_point*) is full.

Response

Delete unnecessary files or move files to a file system that has enough free space in the i-node area.

A.1.3 NOTICE: sfxfs:0003: *mount_point*: file system meta-data full (Vdata) type = *number*

Explanation

The V-data area in the meta-data area of the file system (*mount_point*) is full.

Response

Delete unnecessary files or directories, or move files to a file system that has enough free space in the V-data area.

A.1.4 NOTICE: sfxfs:0004: *mount_point*: too much fragment (extent)

Explanation

The area that is used to manage free space in the file system (*mount_point*) is full because of area fragmentation. In this case, it may have no effect to create free areas by area release.

Response

In order to re-arrange the fragmented areas in the file system, back up the file system, then recreate and restore it.

A.1.5 NOTICE: sfxfs:0005: *mount_point*: daemon create failed

Explanation

Daemon creation failed during mounting the file system (*mount_point*).

Response

The memory load is thought as a cause. Please examine the increase of a real memory.

A.1.6 NOTICE: sfxfs:0010: illegal mode (recovered): *mount_point* | *ino_num* mode
= *file_mode*

Explanation

An error was detected in the i-node of the file system, but the i-node was automatically recovered.

Response

No response is required since the system continues processing after correcting the error of the i-node automatically.

A.1.7 NOTICE: sfxfs:0011: illegal free inode (recovered): *mount_point* | *ino_num*
had size *size*

Explanation

An error was detected in the i-node of the file system (*mount_point*) but the i-node was recovered automatically.

Response

No response is required since the system continues processing after correcting the error of the i-node automatically.

A.1.8 NOTICE: sfxfs:0012: illegal mode (recovered): *mount_point* | *ino_num* mode
= *file_mode*

Explanation

An error was detected in the i-node of the file system (*mount_point*) but the i-node was recovered automatically.

Response

No response is required since the system continues processing after correcting the error of the i-node automatically.

A.1.9 NOTICE: sfxfs:0013: illegal mode (recovered): *mount_point* | *ino_num*

Explanation

An error was detected in the i-node of the file system (*mount_point*) but the i-node was recovered automatically.

Response

No response is required since the system continues processing after correcting the error of the i-node automatically.

A.1.10 NOTICE: sfxfs:0014: illegal mode (recovered): *mount_point* | *ino_num* new
size = *size* blocks = *blocks*

Explanation

An error was detected in the i-node of the file system (*mount_point*) but the i-node was recovered automatically.

Response

No response is required since the system continues processing after correcting the error of the i-node automatically.

A.1.11 PANIC: sfxfs:0015: *mount_point*: bad v_count == 0

Explanation

An error was detected during i-node management of the file system (*mount_point*).

Response

Collect a crash dump and contact your local Customer Support.

A.1.12 PANIC: sfxfs:0016: *mount_point*: bad v_count == 0

Explanation

An error was detected during i-node management of the file system (*mount_point*).

Response

Collect a crash dump and contact your local Customer Support.

A.1.13 NOTICE: sfxfs:0020: *mount_point*: bad namelen == 0 parent ino = *number*

Explanation

An illegal path name request whose name length is zero was issued during directory manipulation.

Response

No response is required because the system rejects the illegal request and continues processing. If this error occurs frequently, determines the target directory for the operation from "parent ino" so as to determine the operation that caused the illegal request, then act accordingly.

A.1.14 PANIC: sfxfs:0021: *mount_point*: illegal operation: op = *code*

Explanation

An illegal operation was detected during processing in the file system (*mount_point*).

Response

Collect a crash dump and contact your local Customer Support.

A.1.15 WARNING: sfxfs:0022: *mount_point*: bad dir ino *number* at offset *offset*.

info

Explanation

A directory entry error was detected.

Response

Unmount the corresponding file system, and execute the fsck_sfxfs(1M) with the specification "-o nolog."

A.1.16 WARNING: sfxfs:0023: *mount_point*: target directory link count < 2 ino =

number

Explanation

An error was detected in directory management information.

Response

Unmount the corresponding file system, and execute the fsck_sfxfs(1M) with the specification "-o nolog."

A.1.17 WARNING: sfxfs:0024: *mount_point*: invalid offset *offset* ino = *number***Explanation**

A directory entry error was detected.

Response

Unmount the corresponding file system, and execute the `fsck_sfxfs(1M)` with the specification "`-o nolog.`"

A.1.18 PANIC: sfxfs:0025: *mount_point*: invalid slot status *code***Explanation**

An error was detected during processing in the file system (*mount_point*).

Response

Collect a crash dump and contact your local Customer Support.

A.1.19 NOTICE: sfxfs:0026: *mount_point*: bad namelen == 0 parent ino = *number***Explanation**

An illegal path name request whose name length is zero was issued during directory manipulation.

Response

No response is required because the system rejects the illegal request and continues processing. If this error occurs frequently, determines the target directory for the operation from "parent ino" so as to determine the operation that caused the illegal request, then act accordingly.

A.1.20 WARNING: sfxfs:0027: *mount_point*: bad dir, ino = *number***Explanation**

A directory entry error was detected.

Response

Unmount the corresponding file system, and execute the `fsck_sfxfs(1M)` with the specification "`-o nolog.`"

A.1.21 WARNING: sfxfs:0030: *mount_point*: meta-data read error. type = *number*

dev = *dev_num* *block* = *block_num* *size* = *size*

Explanation

An error occurred while reading from the meta-data area on the disk.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.22 WARNING: sfxfs:0031: *mount_point*: meta-data read error. type = *number*

dev = *dev_num* *block* = *block_num* *size* = *size*

Explanation

An error occurred while reading from the meta-data area on the disk.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.23 WARNING: sfxfs:0032: *mount_point*: bitmap read error. type = *number*

dev = *dev_num* block = *block_num* size = *size*

Explanation

An error occurred during reading from the bitmap area on the disk. Mount processing failed.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.24 WARNING: sfxfs:0033: *mount_point*: bitmap read error. type = *number* dev

= *dev_num* block = *block_num* size = *size*

Explanation

An error occurred during reading from the bitmap area on the disk.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.25 WARNING: sfxfs:0034: *mount_point*: bitmap read error. type = *number* dev

= *dev_num* block = *block_num* size = *size*

Explanation

An error occurred during reading from the bitmap area on the disk. The freed area has been invalidated.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.26 WARNING: sfxfs:0035: *mount_point*: bitmap write error. information was

lost. type = *number* dev = *dev_num* block = *block_num* size = *size*

Explanation

An error occurred during writing to the bitmap area on the disk while the file system (*mount_point*) was unmounted. Some part of the file system information was lost.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.27 WARNING: sfxfs:0036: *mount_point*: bitmap write error. type = *number* dev

= *dev_num* block = *block_num* size = *size*

Explanation

An error occurred during writing to the bitmap area on the disk.

Response

If message 0035 or 0039 is not output during unmount or shutdown processing, no response is required because

writing was successfully retried.

A.1.28 WARNING: sfxfs:0037: *mount_point*: meta-data write error. type = *number*

dev = *dev_num* block = *block_num* size = *size*

Explanation

An error occurred during writing to the meta-data area on the disk.

Response

If message 0040 is not output during unmount or shutdown processing, no response is required because writing was successfully retried.

A.1.29 WARNING: sfxfs:0038: *mount_point*: fail to update super block at first time

dev = *dev_num* block = *block_num* size = *size*

Explanation

An attempt to update the super block failed during mount processing. Mount processing failed.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.30 WARNING: sfxfs:0039: *mount_point*: bitmap write error. information was

lost. type = *number* *dev* = *dev_num* block = *block_num* size = *size*

Explanation

An error occurred during writing to the bitmap area on the disk while the file system (*mount_point*) was unmounted. Some part of the file system information was lost.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.31 WARNING: sfxfs:0040: *mount_point*: meta-data write error. information was

lost. type = *number* *dev* = *dev_num* block = *block_num* size = *size*

Explanation

An error occurred during writing to the meta-data area on the disk while the file system (*mount_point*) was unmounted. Some part of the file system information was lost.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.32 NOTICE: sfxfs:0041: *mount_point*: I/O error on log: *dev* = *dev_num* block =

block_num size = *size* error = *number*

Explanation

Writing to the update log failed.

Response

If message 0042 is not output during unmount or shutdown processing, no response is required because writing was successfully retried.

A.1.33 WARNING: sfxfs:0042: *mount_point*: LOG daemon is dead. dev =
dev_num

Explanation

The log daemon stopped processing because a failure in writing to the update log area could not be resolved.

Response

The update log daemon quit, as it could not operate because of an I/O error. This failure prevented the use of the update log and therefore degraded file system access performance. Back up the file system, then act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.34 WARNING: sfxfs:0043: *mount_point*: super block write error. dev =
dev_num block = *block_num* size = *size*

Explanation

An error occurred while a super block was written to the disk.

Response

If message 0044 is not output during unmount or shutdown processing, no response is required because writing was successfully retried.

A.1.35 WARNING: sfxfs:0044: *mount_point*: super block write error. information
was lost. dev = *dev_num* block = *block_num* size = *size*

Explanation

An error occurred while a super block was written to the disk.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.36 WARNING: sfxfs:0045: *mount_point*: cannot access: too many I/O errors.
dev = *dev_num*

Explanation

The file system (*mount_point*) has shut down because an error occurred while writing in the Vdata area. The access to this file system continues to return an error.

Response

After system reboot or forcibly unmount the file system, act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.37 WARNING: sfxfs:0046: *mount_point*: cannot access: too many I/O errors.

dev = dev_num

Explanation

The file system (*mount_point*) has shut down because an error occurred while writing to the super block. The access to this file system continues to return an error.

Response

After system reboot or forcibly unmount the file system, act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.38 NOTICE: sfxfs:0050: setting sfxfs_ninode to max value of *value***Explanation**

A value that exceeded the specified range was originally specified in *sfxfs_ninode*, but this value was changed appropriately.

Response

No response is required. Because the *sfxfs_ninode* is set to the maximum value and the system continues processing. If the value of *sfxfs_ninode* was specified in */etc/system*, review the value.

A.1.39 NOTICE: sfxfs:0051: freeing wrong file system by umount *file_num***Explanation**

The file system management detected an attempt to release the wrong file system.

Response

No response is required. Because the system ignores the request that free file system and continues processing.

A.1.40 PANIC: sfxfs:0052: *v_count* != 0 *ino_pointer* *ino_num***Explanation**

An abnormality was detected in the i-node for releasing the file system.

Response

Collect a crash dump and contact your local Customer Support.

A.1.41 PANIC: sfxfs:0053: *mount_point*. illegal transaction buffer. *bufno* = *number***Explanation**

An error during update log management was detected.

Response

Collect a crash dump and contact your local Customer Support.

A.1.42 PANIC: sfxfs:0054: *mount_point*. illegal transaction buffer. *bufno* = *number***Explanation**

An error during update log management was detected.

Response

Collect a crash dump and contact your local Customer Support.

A.1.43 PANIC: sfxfs:0055: *mount_point*: buffer overflow. size = size

Explanation

An error during update log management was detected .

Response

Collect a crash dump and contact your local Customer Support.

A.1.44 PANIC: sfxfs:0056: *mount_point*: hold count is invalid. type = *number*

number = *number*

Explanation

An error was detected in the reference count for file system data.

Response

Collect a crash dump and contact your local Customer Support.

A.1.45 PANIC: sfxfs:0057: page_create == NULL

Explanation

An error was detected during i-node management of the file system.

Response

Collect a crash dump and contact your local Customer Support.

A.1.46 PANIC: sfxfs:0058: bn == SFXFS_HOLE

Explanation

An error was detected in file system i-node management.

Response

Collect a crash dump and contact your local Customer Support.

A.1.47 PANIC: sfxfs:0059: bn == SFXFS_HOLE

Explanation

An error was detected during i-node management of the file system.

Response

Collect a crash dump and contact your local Customer Support.

A.1.48 NOTICE: sfxfs:0060: *mount_point*: read only mount, but modified

(recovered):

Explanation

An error was detected in the file system (*mount_point*) but was resolved automatically.

Response

No response is required. Because the system ignores the request that modify file system and continues processing.

A.1.49 PANIC: sfxfs:0062: bn == SFXFS_HOLE and rw == S_WRITE

Explanation

An error was detected during i-node management of the file system.

Response

Collect a crash dump and contact your local Customer Support.

A.1.50 WARNING: sfxfs:0063: *mount_point*: cannot delete extent. ino = *number*

len = *len* *width* = *width* *length* = *len*

Explanation

An error was detected during extent management of the file system (*mount_point*).

Response

Unmount the corresponding file system, and execute the `fsck_sfxfs(1M)` with "`-o nolog`" specified.

A.1.51 WARNING: sfxfs:0064: *mount_point*: no extent exists. ino = *number* *osize*

= *len* *len* = *len*

Explanation

An error was detected during extent management of the file system (*mount_point*).

Response

Unmount the corresponding file system, and execute the `fsck_sfxfs(1M)` with "`-o nolog`" specified.

A.1.52 WARNING: sfxfs:0065: *mount_point*: cannot delete extent. ino = *number*

len = *len* *width* = *width* *length* = *len*

Explanation

An error was detected during extent management of the file system (*mount_point*).

Response

Unmount the corresponding file system, and execute the `fsck_sfxfs(1M)` with "`-o nolog`" specified.

A.1.53 WARNING: sfxfs:0070: *mount_point*: super block read error. dev =

dev_num

Explanation

An error occurred during reading from the super block on the disk while the file system (*mount_point*) was mounted.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.54 WARNING: sfxfs:0071: *mount_point*: primary super block read failed, but secondary read. *dev = dev_num* *block = block_num* *size = size*

Explanation

The primary super block in the indicated device could not be read but the secondary super block was read successfully, and therefore the system continues processing.

Response

File system manipulation can be continued. However, since a block error occurred, promptly act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.55 WARNING: sfxfs:0072: *mount_point*: secondary super block read failed, but primary read. *dev = dev_num* *block = block_num* *size = size*

Explanation

The secondary super block in the indicated device could not be read but the primary super block was read successfully, and therefore the system continues processing.

Response

File system manipulation can be continued. However, since a block error occurred, promptly act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.56 NOTICE: sfxfs:0073: *mount_point*: VTOC read error. *dev = dev_num*

Explanation

The size of the specified partition in the indicated device could not be read but the primary super block was read successfully, and therefore the system continues processing. Only when the file system is done in the mount by using `mount(2)` directly, this message is generated. In the mount of the file system, by `mount_sfxfs(1M)`, it is not generated.

Response

Use `mount_sfxfs(1M)` instead of `mount(2)` system call for mounting the file system. If this message is output, first unmount the file system, then execute the `sfxadm(1M)` with `-r` option so as to make the two super blocks consistent.

A.1.57 WARNING: sfxfs:0074: *mount_point*: primary and secondary super blocks are different. *dev = dev_num*

Explanation

An inconsistency was detected between the primary and secondary super blocks in the indicated device. Mount processing failed.

Response

Display partition information by `fstyp_sfxfs(1M)` with `-v` option, and confirm to the file system composition errorless it. Check whether the configuration of the specified file system is normal. If it is abnormal, make the file system again. If it is normal, execute the `sfxadm(1M)` with no option to set up the information on file system configuration again, then retry mounting after recover the file system with `fsck_sfxfs(1M)`.

A.1.58 WARNING: sfxfs:0075: *mount_point*: partition configuration read error. dev = *dev_num*

Explanation

Partition configuration information could not be read while the file system (*mount_point*) was mounted. Mount processing failed.

Response

Act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.59 WARNING: sfxfs:0076: *mount_point*: primary partition configuration read failed, but secondary read. dev = *dev_num* block = *block_num* size = *size*

Explanation

The primary set of partition configuration information for the indicated device could not be read but the secondary set of partition configuration information was read successfully, and the system therefore continues processing.

Response

File system manipulation can be continued. However, since a block error occurred, promptly act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.60 WARNING: sfxfs:0077: *mount_point*: secondary partition configuration read failed, but primary read. dev = *dev_num* block = *block_num* size = *size*

Explanation

The secondary set of partition configuration information for the indicated device could not be read, but the primary set of partition configuration information was read successfully, and therefore the system continues processing.

Response

File system manipulation can be continued. However, since a block error occurred, promptly act according to the [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File System."](#)

A.1.61 WARNING: sfxfs:0078: *mount_point*: primary and secondary partition configurations are different. dev = *dev_num*

Explanation

An inconsistency was detected between the primary and secondary set of partition configuration information for the indicated device. Mount processing failed.

Response

Check whether the specified file system configuration is normal. If it is normal, execute the `sfxadm(1M)` with no option to set up the information on file system configuration again, then retry mounting after recover the file system with `fsck_sfxfs(1M)`.

A.1.62 NOTICE: sfxfs:0079: *mount_point*: nodename mismatch. node <-> node

Explanation

A node registered in the file system (*mount_point*) has hostid that matches that of an existing computer system. However, the node has a node name that differs from any of the existing node names.

Explanation

No response is required unless message 0080 is output.

A.1.63 WARNING: sfxfs:0080: *mount_point*: node information not found for *node*

Explanation

Information on the indicated node was not found in the file system (*mount_point*).

Response

Use the sfxnode(1M) to set up node information correctly.

A.1.64 WARNING: sfxfs:0081: *mount_point*: super blocks differ. dev = *dev_num* to *dev_num*

Explanation

An inconsistency was detected between the two super blocks in the indicated file system (*mount_point*) for a multi-partition configuration. Mount processing failed.

Response

Check whether the specified file system configuration is normal. If it is normal, execute the sfxadm(1M) with no option to set up the information on file system configuration again, and then retry mounting.

A.1.65 WARNING: sfxfs:0082: *mount_point*: partition configurations differ. dev = *dev_num* to *dev_num*

Explanation

An inconsistency was detected between the two super blocks in the indicated file system (*mount_point*) in a multi-partition configuration. Mount processing failed.

Response

Check whether the specified file system configuration is normal. If it is normal, execute the sfxadm(1M) with no option to set up the information on file system configuration again, then retry mounting after recover the file system with fsck_sfxfs(1M).

A.1.66 sfxfs:0083: *mount_point*: over file hard limit (pid *pnum*, uid *unum*)

Explanation

The number of the files which user *unum* owns on *mount_point* was reached to the hard limit value of the number of files set up by the quota function. User *unum* cannot own a file any more.

Response

In order for user *unum* to enable create a new file, please delete unnecessary files which user *unum* owns on *mount_point* until the number of files is less than the soft limit value and hard limits value set by quota function.

A.1.67 sfxfs:0084: *mount_point*: Warning: too many files (pid *pnum*, uid *unum*)**Explanation**

The number of files which user *unum* owns on *mount_point* exceeds the soft limit value set by quota function.

Response

In order for user *unum* to enable create a new file, please delete unnecessary files which user *unum* owns on *mount_point* until the number of files is less than the soft limit value and hard limits value set by quota function.

A.1.68 sfxfs:0085: *mount_point*: over file and time limit (pid *pnum*, uid *unum*)**Explanation**

The number of the files which user *unum* owns on *mount_point* was not less than the soft limit value in the time limit set up by the quota function, after it exceeding the soft limit value. User *unum* cannot create a new file until it is less than a soft limit value.

Response

In order for user *unum* to enable create a new file, please delete unnecessary files which user *unum* owns on *mount_point* until the number of files is less than the soft limit value and hard limits value set by quota function.

A.1.69 sfxfs:0086: *mount_point*: over hard disk limit (pid *pnum*, uid *unum*, inum *ino*)**Explanation**

The total of the number of use blocks of a file which user *unum* owns on *mount_point* reached to the hard limit value of the number of blocks set up by the quota function. User *unum* cannot use a block any more.

Response

In order for user *unum* to enable allocate a new block, please delete unnecessary files which user *unum* owns on *mount_point* until the number of use blocks is less than the soft limit value and hard limits value set by quota function.

A.1.70 sfxfs:0087: *mount_point*: Warning: over disk limit (pid *pnum*, uid *unum*, inum *ino*)**Explanation**

The total of the number of use blocks of a file which user *unum* owns on *mount_point* reached to the soft limit value of the number of blocks set up by the quota function.

Response

In order for user *unum* to enable allocate a new block, please delete unnecessary files which user *unum* owns on *mount_point* until the number of use blocks is less than the soft limit value and hard limits value set by quota function.

A.1.71 sfxfs:0088: *mount_point*: over disk and time limit (pid *pnum*, uid *unum*, inum *ino*)**Explanation**

The total of the number of use blocks of a file which user *unum* owns on *mount_point* was not less than the soft limit value in the time limit set up by the quota function, after it exceeding the soft limit value. User *unum* cannot get a new block until it is less than a soft limit value.

Response

In order for user *unum* to enable allocate a new block, please delete unnecessary files which user *unum* owns on *mount_point* until the number of use blocks is less than the soft limit value and hard limits value set by quota function.

A.1.72 WARNING: sfxfs:0089: *mount_point*: segmap_release write error. addr =
addr, offset = *offset*, error = *error*, ino = *ino*

Explanation

During defragmentation, a write error occurred while the extents were being moved.

Response

Take the action indicated in [Appendix C, "Corrective Measures against I/O Errors in the GFS Local File Systems."](#)

A.1.73 WARNING: sfxfs:0100: *mount_point*: cannot find specified volume
information in partition configuration. dev = *dev_num*

Explanation

Information for the device specified for mounting cannot be found in the partition configuration information. Mounting the file system *mount_point* failed.

Response

Display partition information by `fstyp_sfxfs(1M)` with `-v` option, and confirm to the file system composition errorless it. The changed device management information may not have been reflected to the device volume. If the volume configuration is determined as correct, execute the `sfxadm(1M)` with no option to set the file system configuration information again. Then, retry mounting.

A.1.74 NOTICE : sfxfs:0110: *tunename* is too small (*num1*). Setting *num2*

Explanation

The value of *tunename* (*num1*) specified in `/etc/system` is too small. It is changed in minimum value (*num2*).

Response

Use of file system can be continued in order to modify set value automatically at node booted, but review the value of *tunename*, and set the value larger than minimum value (*num2*) again. Then, retry rebooting.

A.1.75 NOTICE : sfxfs:0111: *tunename* is too big (*num1*). Setting *num2*

Explanation

The value of *tunename* (*num1*) specified in `/etc/system` is too big. It is changed in maximum value (*num2*).

Response

Use of file system can be continued in order to modify set value automatically, but review the value of *tunename*, and set the value smaller than maximum value (*num2*) again. Then, retry rebooting.

A.1.76 NOTICE : sfxfs:0112: *mount_point*: *tunename* is too small (*num1*). Setting
num2

Explanation

The value of *tunename* (*num1*) specified in `mount(1M)` is too small. It is changed in minimum value (*num2*).

Response

Use of file system can be continued in order to modify set value automatically, but review the value of tunename, and set the value larger than minimum value (*num2*) again. Then, retry mounting.

A.1.77 NOTICE : sfxfs:0113: *mount_point* : *tunename* is too big (*num1*). Setting
num2

Explanation

The value of tunename (*num1*) specified in mount(1M) is too big. It is changed in maximum value (*num2*).

Response

Use of file system can be continued in order to modify set value automatically, but review the value of tunename, and set the value smaller than maximum value (*num2*) again. Then, retry mounting.

A.1.78 WARNING : sfxfs:0120: *mount_point* : meta data cache grow failed
because of size limitation

Explanation

Since the maximum value of the meta-data cache area size in the whole node was reached, extension of a meta-data cash area was not performed.

Response

Even if this message is outputted, there is no problem in file access operation.

When the access performance to a file system falls chronically and it has influence on operation, a performance fall may be able to mitigate by extending a meta-data cache area.

A meta-data cache area is extensible by changing the each value of meta-data cache area size or meta-data cache area grow limit size.

For detail of change method, refer to "[9.2.3 Tuning Parameters.](#)"

Appendix B Reference Manual

This appendix contains the reference manual for the GFS Local File System.

B.1 Commands for Common File System Management

B.1.1 `clri_sfxfs(1M)` – Clears an i-node

SYNOPSIS

`clri -F sfxfs [-V] special i-number`

DESCRIPTION

The `clri` command clears the i-node numbers of the specified GFS Local File System.

After the `clri` command was executed, the `fsck_sfxfs(1M)` command with the parameter *special* reports an error with respect to the respective block on file.

The file system device requires permission to read and write.

The `clri` command is for deleting entries on a file that can not be found in any directory.

When a file system is configured with two or more partitions, specify the special file name under the path name for the representative partition.

OPTIONS

`-V`

This option displays the command line, but does not execute the command.

EXAMPLE

```
# clri -F sfxfs /dev/sfdsk/gfs/dsk/vol0301 i-number
```

SEE ALSO

[fsck_sfxfs\(1M\)](#)

[vfstab\(4\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

B.1.2 df_sfxfs(1M) - Reports the free disk area of the GFS Local File System

SYNOPSIS

```
df -F sfxfs [ generic_options ] [ -o specific_options ] [ directory | special ]
```

DESCRIPTION

The **df** command displays the number of free disk blocks and of free i-nodes within the GFS Local File System. Note that the total area that is reported as being used or being ready for use is smaller than the actual area for the entire file system. This is because the system reserves a certain part of the file system area that is to be used for processing purposes of the file system allocation routine. This reserved area is generally 10% of the entire available area, but this size can be adjusted by [sfxxtune\(1M\)](#). If the entire file system area is used with the exception of the reserved area, only the superuser can allocate the data blocks of new and existing files. If excessive space is allocated for the file system area in this way, the **df** command reports that the file system area is used by more than 100%.

The special file or mounting point name can be specified. If it is omitted, information on each of the mounted file systems is displayed.

When a file system is configured with two or more partitions by the multi-partition function, specify the special file name with the path name of the representative partition.

When the GFS Local File System is configured with two or more partitions, information on each partition can be displayed.

The following information is displayed for each meta partition:

- i-node usage status (total number of i-nodes and number of free i-nodes)
- Free extent status (total number of leaf and number of free leaf)
- V data usage status (total number of blocks and number of free blocks)

The following information is displayed for each data partition:

- Free space in the data area

The following information is displayed for each log partition:

- Update log area size

The option *generic_options* is supported by the generic **df** command.

See the description of **df(1M)** options.

OPTIONS

-F sfxfs

This option specifies sfxfs-FSType.

-a

This option outputs information on file systems including on file system for which the ignore option was set in the respective */etc/mnttab* entry.

-b

This option outputs the usable size in kilobytes.

-e

This option outputs only the number of usable files.

-g

This option outputs the statvfs structure.

-k

This option outputs the allocation status in kilobytes.

-l

This option outputs information on local file systems only.

B.1 Commands for Common File System Management

- n** This option outputs file system types only.
- t** This option outputs a complete list including total values.
- V** This option displays the command line, but does not execute the command.
- o** This option specifies sfxfs local options. Delimit these options with a comma ",".
- i** This option reports the number of used and free i-nodes.
- v** This option displays the information on each partition for a multi-partition configuration.

EXAMPLES

```
# df -F sfxfs -o v /mnt
Filesystem:/mnt
              inodes   free  vdata  free   leaf   free   Type
/dev/sfdsk/gfs/dsk/vol0301
              14248  12192   4781  4615   7128   7127  META
              kbytes  used   free  %used
/dev/sfdsk/gfs/dsk/vol0301
              113992 62664  51328   55%
              kbytes
/dev/sfdsk/gfs/dsk/vol0301
              5120
              LOG
# df -F sfxfs -o v /mnt
Filesystem:/mnt
              inodes   free  vdata  free   leaf   free   Type
/dev/sfdsk/gfs/dsk/vol0301
              29686  27630   9922  9756  14848  14846  META
              kbytes  used   free  %used
/dev/sfdsk/gfs/dsk/vol0301
              105752 31832  73920   30%
              kbytes
/dev/sfdsk/gfs/dsk/vol0500
              5120
              LOG
              kbytes  used   free  %used
/dev/sfdsk/gfs/dsk/vol0600
              131736 30832 100904   23%
              DATA
```

SEE ALSO

[sfx tune\(1M\)](#), [fs_sfxfs\(4\)](#)
[df\(1M\)](#), [mnttab\(4\)](#), [attributes\(5\)](#)

B.1.3 ff_sfxfs(1M) - Displays file names and statistical information

SYNOPSIS

```
ff -F sfxfs [ generic_options ] [ -o specific_options ] special ...
```

DESCRIPTION

The **ff** command outputs the GFS Local File System file name that was specified with the parameter *special*, and its i-node number.

When a file system is configured with two or more partitions, specify the special file name under the path name for the representative partition.

The option *generic_options* is supported by the generic **ff** command.

See the description of **ff(1M)** options.

OPTIONS

-V

This option displays the command line, but does not execute the command.

-l

This option does not output any i-node number.

-l

This option outputs the path names for all multilink files.

-p prefix

This option adds the specified prefix to the path name.

-s

This option outputs file size.

-u

This option outputs the owner's login name.

-a -n

This option selects files that were accessed within n days.

-m -n

This option selects files that were modified or created within n days.

-c -n

This option selects files whose status changed within n days.

-n file

This option selects files that were modified after the specified file.

-i i-node-list

This option outputs i-nodes that were specified with i-node-list.

-o

This option specifies **sfxfs** local options. Delimits these options with a comma ",".

a

This option displays "." and ".." directory entries.

m

This option displays mode information (valid with -i).

s

This option displays special files and files with set user ID mode.

EXAMPLE

```
# ff -F sfxfs /dev/sfdsk/gfs/dsk/vol0301
```

SEE ALSO

find(1), ff(1M), ncheck(1M), attributes(5)

B.1.4 fsck_sfxfs(1M) - Checks and repairs the GFS Local File System

SYNOPSIS

fsck -F sfxfs [*generic_options*] [**-o** *specific_options*] [*special ...*]

DESCRIPTION

fsck audits and repairs inconsistent GFS Local File System conditions.

The GFS Local File System can be recovered at high speed with the update log function. **fsck** specifies whether to execute high-speed recovery by update log replay. If the update log replay cannot be performed because of a damaged update log, the **fsck** command performs a complete check of the file system. Specify the file system that is to be checked with the character special device or the mounting point of that file system that contains the entries for `/etc/vfstab`.

Specify the parameter *special* for the character special device in which the file system has been set up or with the mounting point. A block special device should not be specified here. The **fsck** command is not executed when the block device is mounted.

If no special device is specified **fsck** checks all GFS Local File Systems that contain `fsckdev` entries in `/etc/vfstab`. If the `p` (`preen`) option is specified, **fsck** checks the GFS Local File Systems with `fsckpass` 1 or more.

If a fatal error is found, **fsck** prompts the operator for confirmation before correcting the error. If the operator is not authorized to write to a file system, **fsck** operates with the `-n` option (no correction).

fsck automatically corrects the following minor errors:

- Some i-nodes are not referenced.
- An i-node contains too many links.
- The free list contains an incorrect block.
- A block is listed both as "free" and as a file.
- A superblock contains an invalid value.

Whenever one of these errors is corrected, a corresponding message is displayed.

When a file system was corrected successfully, the **fsck** command displays the following items:

- Number of files on the file system
- Number of used and free blocks
- Percentage of fragmentation

fsck checks for the following inconsistencies:

- A block was acquired by two or more i-nodes or free lists
- A block was requested by i-nodes outside the file system or the free list
- Invalid link count
- Invalid directory size
- i-nodes with an invalid format
- Non-referenced blocks
- Directory check: Files that refer to unallocated i-nodes, i-node numbers that exceed the range, directory names that start with "." or ".." .
- Superblock check: Number of i-nodes exceeds the number of file system blocks
- Free block list contains an invalid format
- Incorrect total number of free blocks or free i-nodes

Disconnected files and directories (that were allocated but not referenced) are placed and re-linked in the `lost+found` directory after confirmation by the operator. i-node numbers are assigned as name for these files. If the `lost+found` directory does not exist, it is newly created. Should this directory become insufficient, it is extended.

When a file system is configured with two or more partitions, specify the special file name under the path name for the representative partition.

OPTIONS

-v

This option displays the command line, but does not execute the command.

- m** This option checks the file system, but does not correct any errors.
- n|N** This option answers no to all inquiries. It does not open the file system in write mode.
- y|Y** This option answers yes to all inquiries.
- o** This option specifies local options for the GFS Local File System. Delimit these options by comma ",".
- f** This option forcibly checks the file system, regardless of the superblock flag status.
- p** This option checks and corrects the file system in non-conversational mode.
- w** This option checks only write-enabled file systems.
- log|elog|nolog** This option specifies whether to perform update log replay. The default value is log
- log** This option performs update log replay. If log data is physically damaged, the entire file system is checked.
- elog** This option performs update log replay. If log data is physically damaged, the command ends immediately, without checking and correcting the file system.
- nolog** This option checks the entire file system without performing update log replay.

EXAMPLES

```
# fsck -F sfxfs /dev/sfdsk/gfs/rdisk/vol0301
# fsck -F sfxfs -o nolog /dev/sfdsk/gfs/rdisk/vol0301
```

EXIT STATUS

- 0 File system correction ended normally or no correction was required.
- 1 The specified parameter is incorrect.
- 32 File system mounting was canceled and a check is required. (for fsck -m only)
- 33 The file system is mounted.
- 34 stat(2) for a device failed.
- 36 Processing ended because an important problem, which requires operator intervention, occurred during execution of the p (preen) option.
- 37 An interrupt occurred during command execution.
- 39 An irregularity occurred during command execution and processing ended.
- 50 Update log replay could not be executed, since an update log was damaged physically during execution of the elog option.

SEE ALSO

[mkfs_sfxfs\(1M\)](#), [mount_sfxfs\(1M\)](#), [sfxnews\(1M\)](#), [fs_sfxfs\(4\)](#), [cli\(1M\)](#), [crash\(1M\)](#), [fsck\(1M\)](#), [fsirand\(1M\)](#), [fstyp\(1M\)](#), [mkfs\(1M\)](#), [mountall\(1M\)](#), [reboot\(1M\)](#), [vfstab\(4\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

B.1.5 fstyp_sfxfs(1M) - Determines the type of a file system

SYNOPSIS

`fstyp [-v] special`

DESCRIPTION

The **fstyp** command determines the type for each of the file systems.

The **fstyp** command executes modules that check the type for each of the file systems.

When the module is applicable for the GFS Local File System, it displays the file system name "**sfxfs**" over the standard output and ends with the exit status is 0.

If none of the modules succeed, it displays the error message `unknown_fstyp` (no matches) and ends with exit status is 1.

If more than one module succeeds, it displays the error message `unknown_fstyp` (multiple match) and ends with exit status is 2.

When a file system is configured with two or more partitions, specify the special file name with the path name for the representative partition.

OPTIONS

-v

This option outputs the information highly detailed.
It displays superblock and partition information.

EXAMPLES

```
# fstyp /dev/sfdsk/gfs/dsk/vol0301
sfxfs
```

SEE ALSO

[mkfs_sfxfs\(1M\)](#), [sfx tune\(1M\)](#), [fs_sfxfs\(4\)](#)
[attributes\(5\)](#), [largefile\(5\)](#), [hsfs\(7FS\)](#), [pcfs\(7FS\)](#)

B.1.6 mkfs_sfxfs(1M) - Builds the GFS Local File System

SYNOPSIS

mkfs -F sfxfs [*generic_options*] [**-o** *specific_options*] *raw_device_file*

DESCRIPTION

The **mkfs** command configures the GFS Local File System on the specified disk.

In GFS Local File System, update logs can be allocated to another partition so as to enhance file system performance. In this case, it is recommended to use a dedicated physical device for the update logs.

In GFS Local File System, multi-partition file systems can be specified. This makes it possible to configure a large file system independent of the physical disk size.

In GFS Local File System, the data block size for each file system is specified as 8,192 bytes.

An entry is registered with `/etc/sfxfstab` that a GFS Local File System is configured. `/etc/sfxfstab` is used for management of a GFS Local File System.

The Parameter `generic_options` is supported for the generic **mkfs** command.
See the description of **mkfs(1M)** options.

OPTIONS

-o

Specifies the GFS Local File System specific suboptions, commas delimit suboptions.

N

This option specifies a parameter that is used to create a file system.

The file system is not actually created by this option.

data=raw_device_file

This option specifies a character special device of data partition

When configuring a data partition with multiple partitions, specify this parameter multiple times as required.

log=raw_device_file

This option specifies a character special device of update log partition.

Specify a partition that accommodates the size of the update log area, because, except the area used as an update log area, the area of update log partition remains unused.

free=n

This option specifies the minimum percentage of free areas in the file system.

The default value is 10%.

Values from 0 to 99 can be used here.

mfree=n

This option specifies the minimum percentage of free areas in the V-data area.

The default value is 10%.

Values from 0 to 99 can be used here.

nbpi=n

This option specifies the data size per i-node in units of bytes.

The default value is 8,192.

Specify a larger value when saving a metadata area.

Values from 512 to 1,048,576 can be used here.

opt=s|t

This option specifies whether to optimize space or time.

The parameter `s` indicates space, `t` indicates time.

The default value is `t`.

B.1 Commands for Common File System Management

maxnode=*n*

This option specifies the maximum number of nodes sharing the file system, including also the case that the file system is in the future expanded with **sfxnode**(1M).

The default value is 16.

Values from 1 to 16 can be used here.

maxvol=*n*

This option species the maximum number of file system component partitions, including also the case that the file system is in the future expanded with the **sfxadd** command.

The default value is 16.

Values from 1 to 32 can be used here.

maxdsz=*n*

This option specifies the maximum size of the data areas for each file system, including also the case that a data partition is to be added.

This parameter is to be specified in megabyte.

A value larger than 1 terabyte (1,048,576 megabyte) cannot be used here.

metasz=*n*

This option specifies the size of each metadata area.

This parameter is to be specified in megabyte.

A value that exceeds the size of the representative partition cannot be used here.

logsz=*n*

This option specifies the size of each update log area.

This option is ignored when an update log partition is specified.

This parameter is to be specified in megabyte.

A value from 5 to 100 can be used here.

nblkpext=*n*

This option specifies the average number of blocks that have to be managed in each extent.

This value can be used to calculate data area fragmentation.

The default value is 1.

A value from 1 to 100 can be used here.

dataopt=*y|n*

This option specifies whether or not a data area is included in a representative partition if a data partition is specified.

If *y* is specified, a data area is not included in a representative partition.

If *n* is specified, a data area is included.

If a data partition has not been specified, *n* is used regardless of the specification of this option.

The default is *n*.

OPERANDS

raw_device_file

This file contains the names of each of the file system component partitions.

EXAMPLES

```
mkfs -F sfxfs /dev/sfdsk/gfs/rdsk/vol0301
mkfs -F sfxfs -o log=/dev/sfdsk/gfs/rdsk/vol0601 /dev/sfdsk/gfs/rdsk/vol0301
mkfs -F sfxfs -o data=/dev/sfdsk/gfs/rdsk/vol0403,data=/dev/sfdsk/gfs/rdsk/vol0504 /dev/sfdsk/gfs/rdsk/vol0305
```

FILE

/etc/sfxfstab

This file contains the names of each of the file system component partitions.

SEE ALSO

[sfxadd\(1M\)](#), [sfxnews\(1M\)](#), [sfxnode\(1M\)](#), [sfx tune\(1M\)](#), [fs_sfxfs\(4\)](#)
fscck(1M), mkfs(1M), attributes(5)

B.1.7 mount_sxfs(1M) - Mounts the GFS Local File System

SYNOPSIS

```
mount -F sxfs [ generic_options ] [ -o specific_options ] special mount_point
mount -F sxfs [ generic_options ] [ -o specific_options ] {special|mount_point}
```

DESCRIPTION

The **mount** command mounts the GFS Local File System at the location `mount_point` within the specified path.

When only `special` or `mount_point` is specified, omitted arguments are retrieved from `/etc/vfstab`.

If the GFS Local File System is configured with two or more partitions using the multi-partition function, specify `special` at mounting with the path name for the representative partition.

The option *generic_options* is supported by the generic mount command.

See the description of **mount(1M)** options.

OPTIONS

-a [mount_points...]

This option is for mounting each of the file systems. If the mount point is omitted, a file system is mounted in `/etc/vfstab` that carries the indicating "yes" in the "mount at boot" field.

-p

This option displays the list of mounted file systems in `/etc/vfstab` format.

-v

This option displays the list in verbose format.

-V

This option displays the command line, but does not execute the command.

-m

This option mounts a file system without creating a `/etc/mnttab` entry.

-r

This option performs mounting in read only mode.

-O

This option performs remounting at the same point where the file system had already been mounted.

-o

This option specifies `sxfs` local options. Delimit these options with a comma ",".

noatime

This option does not update an access time of files.

closetsync

This option causes all unupdated data for the file to be updated the last time the file is closed.

noforcedirectio | forcedirectio

This option specifies whether to execute forced direct I/O. (Direct I/O selection) By default, `noforcedirectio` is used.

nolargefiles | largefiles

This option specifies whether to support large-scale files of 2 gigabytes or more.

Mounting with the specification `nolargefiles` is disabled in file systems that contain large-scale files. By default, `largefiles` is used.

quota

This option turns on quotas for a file system.

m

This option mounts a file system without creating a `/etc/mnttab` entry.

rw|ro

This option specifies read-write or read-only mode. By default, `rw` is used.

nosuid

This option inhibits `setuid` execution and performs mounting.

remount

This option mounts a file system that was mounted in read-only mode again, this time in read-write mode.

intr|nointr

This option specifies whether a process that hung up in a hard-mounted file system can be killed by a keyboard interrupt. By default, **intr** is used.

textent=<val>

This option specifies amount of extent management information cached in memory. If this command succeeds, kernel memory of specified value * 1 kilobyte is allocated to the file system that is mounted by this command.

tdir=<val>

This option specifies amount of directory information that is cached in memory. If this command succeeds, kernel memory of specified value * 1 kilobyte is allocated to the file system that is mounted by this command.

tinode=<val>

This option specifies amount of i-nodes on the disk that is cached in memory. If this command succeeds, kernel memory of specified value * 4 kilobytes is allocated to the file system that is mounted by this command.

tquota=<val>

This option specifies amount of quota information that is cached in memory. If this command succeeds, kernel memory of specified value * 1 kilobyte is allocated to the file system that is mounted by this command.

EXAMPLES

```
mount -F sfxfs /dev/sfdsk/gfs/dsk/vol0301 /mount_point
mount -F sfxfs -o quota /dev/sfdsk/gfs/dsk/vol0301 /mount_point
mount -F sfxfs -o quota,tdir=5120,tinode=4096 /dev/sfdsk/gfs/dsk/vol0301 /mount_point
```

SEE ALSO

[fsck_sfxfs\(1M\)](#), [sfxnewfs\(1M\)](#)

[mount\(1M\)](#), [mountall\(1M\)](#), [mount\(2\)](#), [mnttab\(4\)](#), [vfstab\(4\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

NOTES

- If specified values for **textent**, **tdir**, **tinode** and **tquota** are smaller than the minimum or larger than the maximum, kernel message 0112 or 0113 is indicated. In this case, specified values are modified to the minimum or the maximum, and a file system is mounted, and specified values are set in /etc/mnttab by this command. The values of these are modified automatically, so this file system can be continuously used. But if this message is indicated, after **umount(1M)**, review specified values and retry mounting.
- When **remount** is specified, if **textent**, **tdir**, **tinode** and **tquota** are specified, these values are ignored.

B.1.8 ncheck_sxfs(1M) - Lists file names and i-node numbers

SYNOPSIS

```
ncheck -F sxfs [ generic_options ] [ -o specific_options ] [ special ... ]
```

DESCRIPTION

The ncheck command outputs the GFS Local File System file name that was specified with the parameter special, and its i-node number.

When a file system is configured with two or more partitions, specify the special file name under the path name for the representative partition.

The option *generic_options* is supported by the generic ncheck command. See the description of **ncheck(1M)** options.

OPTIONS

- V** This option displays the command line, but does not execute the command.
- i *i-list*** This option outputs the information on the i-nodes specified with i-list.
- a** This option outputs "." and ".." directory entries.
- s** This option outputs special files and files with set user ID mode.
- o** This option specifies sxfs local options. Delimit these options with a comma ",".
- m** This option displays mode information. Specify this option with the **-i** option.

EXAMPLES

```
# ncheck -F sxfs /dev/sfdsk/gfs/dsk/vol0301
```

SEE ALSO

ff(1M), ncheck(1M), attributes(5)

B.2 Management Commands Specific to the GFS Local File System

B.2.1 sfxadd(1M) - Extends the GFS Local File System

SYNOPSIS

```
sfxadd -D raw_device_file[,raw_device_file,...] raw_device_file
sfxadd -pa -D raw_device_file[,raw_device_file,...] raw_device_file
```

DESCRIPTION

The **sfxadd** command can extend the current GFS Local File System without saving or restoring the data.

The file system can be extended by adding a new partition to the current file system.

The file system can be extended while the file system is mounted (online extension) or unmounted (offline extension). If the file system is extended while the file system is unmounted, the status of the file system must be normal. If an attempt is made to reflect extension of the file system while the status of the file system is not normal, the **sfxadd** command will output the message "file system (partition-name) is corrupted, run fsck manually." In this case, use the **fsck**(1M) command to restore the status of the file system, then execute again.

When adding data partitions, preregister using the **-pa** and **-D** options to recover normally even if an unexpected system failure occurs while a data partition is being newly added. After preregistering, specify the **-D** option to instruct actual reflection.

If preregistration for adding partitions has been performed or if actual extension is not completed because of an unexpected system failure, the file system will be managed as processing midway. Use the **sfxinfo**(1M) command to check whether the file system is being managed as processing midway.

When the **sfxadd** command is used to add a GFS Local File System partition, an entry will be added to and managed in `/etc/sfxfstab`.

OPTIONS

-D *raw_device_file*

This option specifies the character special device for a data partition that is to be added. When adding two or more data partitions, delimit the special file name with a comma ",".

-pa

Pre-register the partition addition instruction. Pre-registering enables processing to be continued normally even if an unexpected system failure occurs. If the file system is used as an alternate file system, pre-register the partition for all systems using the file system.

OPERANDS

raw_device_file

This parameter specifies the character special device name for the representative partition.

EXAMPLES

Case of adding `/dev/sfdsk/gfs/rdsk/vol0601` and `/dev/sfdsk/gfs/rdsk/vol0701` to a GFS Local File System where `/dev/sfdsk/gfs/rdsk/vol0301` is used as a representative partition

Preregister.

```
# sfxadd -pa -D /dev/sfdsk/gfs/rdsk/vol0601,/dev/sfdsk/gfs/rdsk/vol0701 /dev/sfdsk/gfs/rdsk/vol0301
```

Reflect the setting data to the file system.

```
# sfxadd -D /dev/sfdsk/gfs/rdsk/vol0601,/dev/sfdsk/gfs/rdsk/vol0701 /dev/sfdsk/gfs/rdsk/vol0301
```

B.2 Management Commands Specific to the GFS Local File System

FILE

/etc/sxfstab

This file contains the names of each of the file system component partitions.

SEE ALSO

[df_sxfs\(1M\)](#), [fsck_sxfs\(1M\)](#), [mkfs_sxfs\(1M\)](#), [sfxadm\(1M\)](#), [sfxinfo\(1M\)](#), [sfxnewfs\(1M\)](#), [sfxnode\(1M\)](#)

NOTES

- If the file system is to be extended, estimate the maximum size of extension. Specify the data size of the file system when the file system is created (when the **mkfs** or **sfxnewfs** command is executed). The GFS Local File System records a free data area as an extent. For this free data management area, the area size is calculated automatically so that the data of the data area specified at file system creation can be managed. When the file system is extended, however, this management area may be rendered insufficient. (Because the free data area is recorded as an extent, the free data area can be managed if not fragmented. If fragmentation is extensive, an unusable free area will be created because of insufficient management area.)
If there appears to be a shortage of management area for a current file system, the file system can be defragmented to render the management area normal. See [Chapter 8, "Defragmenting GFS Local File System."](#) for details on defragmentation. To determine whether the management area is insufficient, specify the **-ov** option in the **df(1M)** command to display the total number of leafs and the number of free leafs.
- When the **-pa** option of the **sfxadd** command is specified to preregister partition addition, it will be recognized as part of existing configuration elements for management of the GFS Local File System, and so the original file system can be deleted. This occurs when creating a GFS Local File System where a partition for which addition has been preregistered is specified as a separate file system to delete the addition specification or when the **sfxadm -D** command is used to delete a preregistered partition from the file system.
Once a partition has been registered, including preregistered partitions, only the addition data can be deleted.
- For operation as an alternate file system from multiple systems, preregister using the **-pa** option for all nodes. If this preregistration is not performed, the recovery data will be insufficient if a system failure occurs during operation of an active node and normal recovery will not be possible. When preregistration has been performed, file system recovery using the **fsck(1M)** command will extend the file system automatically and delete the preregistration data.
- If a system failure occurs during file system extension, the **sfxinfo** command can be used to check whether processing has been completed. Refer to the [sfxinfo\(1M\)](#) manual for details on the **sfxinfo** command options.
- The maximum number of partitions that can be used to configure a file system is set when the file system is created (when the **mkfs** or **sfxnewfs** command is executed). Partitions exceeding the number of partitions specified at file system creation cannot be configured. The default value is 16.

B.2.2 sfxadm(1M) - Changes GFS Local File System partition information settings

SYNOPSIS

```
sfxadm [-r [-a]] raw_device_file,...  
sfxadm -L size raw_device_file  
sfxadm -A raw_device_file,...  
sfxadm -D raw_device_file,...  
sfxadm -U raw_device_file
```

DESCRIPTION

The sfxadm command changes the settings for GFS Local File System partition information.

GFS Local File System contains information on the file system component partitions in each of the partitions.

The partition information settings must be added or changed in the following cases:

- When the special file name of file system has changed
- When a file system is moved to another system
- When the size of a log area in a log partition is changed
- When the version of the file system is upgraded

When the sfxadm is used to change partition configuration information for the GFS Local File System, the entry in /etc/sfxfstab will be updated and the information on the each partition will be changed.

OPTIONS

-r

This option is superblock and partition information restore.

Partition in the restoration origin become first partition specified by special,...

-a

When the -r option is specified, performs the recovery using a copy of the super block and the partition configuration data in the partition of the recovery source. Use this option if the super block or the partition configuration data is broken.

raw_device_file, ...

This option specifies the character special device name for each partition in the GFS Local File System whose partition information is changed.

-L *size raw_device_file*

This option changes the size of log area in a log partition. In the size field, specify the size of the log area to be changed. This parameter is to be specified in megabyte. A value from 5 to 100 can be used here.

In the raw-device-file field, specify the character special device for the representative partition for the GFS Local File System in which log area size is to be changed.

-A *raw_device_file, ...*

This option adds partition configuration information for the GFS Local File System to /etc/sfxfstab. In the raw_device_file field, specify all character special devices for partitions configuring the file system. At the beginning of the raw_device_file field, specify a representative partition.

At the beginning of the raw_device_file field, specify a representative partition. This option is used if a device cannot be opened on the standby node of the alternate file system. Before specifying this option, specify the -a option with the sfxnode command on an active node to set the node information.

-D *raw_device_file,...*

This option deletes partition configuration information for the GFS Local File System from /etc/sfxfstab. In the raw_device_file field, specify character special devices for partitions configuring the file system.

B.2 Management Commands Specific to the GFS Local File System

-U *raw_device_file*

This option upgrades the version of the GFS Local File System. In the *raw_device_file* field, specify a character special file for the representative partition of the GFS Local File System.

EXAMPLES

```
# sfxadm /dev/sfdsk/gfs/rdisk/vol0301,/dev/sfdsk/gfs/rdisk/vol0400,/dev/sfdsk/gfs/rdisk/vol0500,/dev/sfdsk/gfs/rdisk/vol0600
```

When a file system is configured with two or more partitions, specify the special file name for each partition.

FILE

/etc/sfxfstab

This file contains the names of each of the file system component partitions.

SEE ALSO

[mkfs_sfxfs\(1M\)](#), [sfxnews\(1M\)](#), [sfxnode\(1M\)](#), [sfxadd\(1M\)](#)

B.2.3 sfxddr(1M) - Restores backup data and checks device restoring

SYNOPSIS

```
sfxddr [ -f ] [ if=file ] of=special [ ibs=n ] [ obs=n ]
sfxddr -i [ if=file ] [ ibs=n ]
```

DESCRIPTION

sfxddr restores backup to a disk data for the GFS Local File System created by **dd(1M)** **sfxddr** checks for the following conditions before writing backup data to a disk:

- The specified input file is the GFS Local File System.
- The specified disk is the same device as the backup source.
- The disk is not being mounted.
- The disk partition sizes match.

If any of these conditions is not met, **sfxddr** terminates without restoring the data. When if is not specified, the standard input is used for input. **of** operand is necessary. As with **dd(1M)**, the input-output block size can be specified. Where sizes are specified, a number of bytes is expected. A number may end with **k**, **b**, or **w** to specify multiplication by 1024, 512, or 2 respectively; a pair of numbers may be separated by **x** to indicate multiplication.

After restoration, **sfxddr** divides the number of input-output blocks into the number of transferred whole blocks plus any partial blocks transferred, and reports the respective result.

OPTIONS

-f

This option does not check whether the output device is the same as the backup source. This option is available only for restoring data to a different disk from the backup source.

-i

This option displays the partition configuration, creation time, and the path of the last mounted directory, concerning to file system of the backup data. This option does not restore data.

OPERANDS

The following operands are supported:

if=file

Specify the input path; standard input is the default.

of=special

Specify the output device name.

ibs=n

Specify the input block size in n bytes (default is 512).

obs=n

Specify the output block size in n bytes (default is 1024k).

EXIT STATUS

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

DIAGNOSTICS

f+p records in(out)

numbers of full and partial blocks read(written)

NOTES

- When data is restored to a different disk from the backup source with **-f**, use **sfxadm(1M)** to change the partition information.

B.2 Management Commands Specific to the GFS Local File System

- To use **sfxddr** for backup data created with the **-c** option of **sfxmkbkup(1M)**, specify **uncompress(1)** to uncompress the backup data.

```
# dd if=/dev/rmt/0 ibs=256k | uncompress -c | sfxddr -i
```

B.2.4 sfxdefrag(1M) - Defragments the GFS Local File System

SYNOPSIS

```

sfxdefrag [ -f ] [ -V | -S ] { special | mount-point } { simple | adjacent | evacuate | dircomp }
sfxdefrag [ -f ] [ -V | -S ] [ -h ] { special | mount-point } { simple | adjacent | evacuate | dircomp } file
[ file [...] ]
sfxdefrag [ -f ] [ -V | -S ] { special | mount-point } free

```

DESCRIPTION

The **sfxdefrag** command uses the specified defragmentation policy to defragment the GFS Local File System specified using the block special file named *special* or the mount point *mount-point* of a representative partition.

The **sfxdefrag** command can defragment the specified file system while the file system is mounted (online) or unmounted (offline).

The defragmentation policies are as follows:

simple policy:

The extent allocation efficiency of a file is improved.

The area is allocated from large free extents and fragmentation is removed.

If sufficient large free extents are existed, defragmentation can be performed faster than the other policies (adjacent and evacuate).

adjacent policy:

The extent allocation efficiency of a file is improved.

The free extents following the extent to be defragmented are allocated and fragmentation is removed.

evacuate policy:

The extent allocation efficiency of a file is improved.

If the extents following the extent to be defragmented are used by other files, the extents are allocated after being forcibly moved and fragmentation is removed.

Although time is required to forcibly move the extents used, this policy offers ideal defragmentation.

free policy:

Large free extents are made out to the file system.

dircomp policy:

Unnecessary extents in the directory are deleted and the directory is compressed.

Only a super user can execute the **sfxdefrag** command.

OPTIONS

The following options are supported:

-f

This option specifies offline defragmentation. An error will occur if this option is specified for an online file system. When this option is specified, the mount information must be registered to */etc/vfstab* file beforehand.

-V

This option displays progress in detail.

-S

This option does not display progress.

-h

This option specifies that only the specified directory hierarchy is to be defragmented when a directory to be defragmented is specified.

If this option is not specified, all hierarchies under the specified directory will be defragmented.

OPERANDS

The following operands can be specified:

special

Specify the block special file name of the representative partition of the target file system.

When the *special* operand and **-f** option are specified, specify the block special file name registered in the */etc/vfstab* file.

mount-point

Specify the mount point of the target file system.

When the *mount-point* operand and **-f** option are specified, specify the mount point registered in the */etc/vfstab* file.

file [file [...]]

Specify the file or directory to be defragmented when **simple**, **adjacent**, or **evacuate** is specified as the defragmentation policy.

Use a relative path name for the root directory of the target file system for the file or directory.

EXIT STATUS

The following exit states are returned:

0 Fragmentation terminated normally.

>0 An error occurred.

EXAMPLES

- Using the **simple** policy to defragment the file system specified by mount point */sfxfs* in unmount state

```
# sfxdefrag -f /sfxfs simple
```

- Using the **adjacent** policy to defragment directory */sfxfs/target_dir* (file system mounted at mount point */sfxfs*) of the file system of special block file */dev/sfdsk/gfs/dsk/vol1100*

```
# sfxdefrag /dev/sfdsk/gfs/dsk/vol1100 adjacent target_dir
```

- Using the **evacuate** policy to defragment files *target_file1* and *target_file2* under the mount point */sfxfs*

```
# sfxdefrag /sfxfs evacuate target_file1 target_file2
```

- Using the **free** policy to defragment the file system of mount point */sfxfs*

```
# sfxdefrag /sfxfs free
```

- Using the **dircomp** policy to defragment only the hierarchy of directory */sfxfs/target_dir* (file system mounted at mount point */sfxfs*) of the file system of special block file */dev/sfdsk/gfs/dsk/vol1100*

```
# sfxdefrag -h /dev/sfdsk/gfs/dsk/vol1100 dircomp target_dir
```

FILES

/etc/mnttab

Specifies the block special file name or mount point in the file when defragmenting online.

/etc/vfstab

Specifies the block special file name or mount point in the file when defragmenting offline.

NOTES

- A file system mounted using read-only cannot be defragmented.
- A file system cannot be defragmented in offline state when read-only is specified in */etc/vfstab*.
- The following operations cannot be executed when defragmenting online:

B.2.4 sfxdefrag(1M) - Defragments the GFS Local File System

- The file system cannot be unmounted (**umount(1M)**).
- The file system size cannot be extended (**sfxadd(1M)**).
- The following operations cannot be executed when defragmenting offline:
 - The file system cannot be mounted (**mount(1M)**).
 - The file system size cannot be extended (**sfxadd(1M)**).
 - The file system integrity cannot be recovered (**fsck(1M)**).
 - Files cannot be manipulated.
- If offline and an option enabling the quota function in /etc/vfstab have been specified, defragmentation will be performed. However, the quota function will not be enabled, and as a result, if the quota function is enabled after offline defragmentation has been performed, the quota data will be updated.

B.2.5 sfxdump(1M) - Dumps a GFS local file system

SYNOPSIS

sfxdump [*options*] [*arguments*] *files_to_dump*

DESCRIPTION

sfxdump backs up all files specified by *files_to_dump* (normally either a whole file system or files within a file system changed after a certain date) to magnetic tape, diskette, or disk file. When running **sfxdump**, the file system must be inactive; otherwise, the output of **sfxdump** may be inconsistent and restoring files correctly may be impossible. A file system is inactive when it is unmounted or the system is in single user mode. A file system is not considered inactive if one tree of the file system is quiescent while another tree has files or directories being modified.

Options are a single string of one-letter **sfxdump** options.

Arguments may be multiple strings whose association with the options is determined by order. That is, the first argument goes with the first option that takes an argument; the second argument goes with the second option that takes an argument, and so on.

files_to_dump is required and must be the last argument on the command line. See OPERAND for more information.

With most devices **sfxdump** can automatically detect the end-of-media. Consequently, the *d*, *s*, and *t* options are not necessary for multi-volume dumps, unless **sfxdump** does not understand the way the device detects the end-of-media, or the files are to be restored on a system with an older version of the restore command.

The **sfxdump** and **ufsdump** commands are functionally compatible, though the backup formats are different. As a result, the **ufsrestore** command cannot use the backup data created using the **sfxdump** command for restoration. In the same way, the **sfxrestore** command cannot use the backup data created using the **ufsdump** command for restoration.

OPTIONS

The following options are supported:

0-9

The "dump level." All files specified by *files_to_dump* that have been modified since the last **sfxdump** at a lower dump level are copied to the *dump_file* destination (normally a magnetic tape device). For instance, if a "level 2" dump was done on Monday, followed by a "level 4" dump on Tuesday, a subsequent "level 3" dump on Wednesday would contain all files modified or added since the "level 2" (Monday) backup. A "level 0" dump copies the entire file system to the *dump_file*.

a archive_file

Archive file. Archive a dump table-of-contents in the specified *archive_file* to be used by **sfxrestore**(1M) to determine whether a file is in the dump file that is being restored.

b factor

Blocking factor. Specify the blocking factor for tape writes. The default is 20 blocks per write for tapes of density less than 6250BPI (bytes-per-inch). The default blocking factor for tapes of density 6250BPI and greater is 64. The default blocking factor for cartridge tapes (*c* option) is 126. The highest blocking factor available with most tape drives is 126. Note: the blocking factor is specified in terms of 512-byte blocks, for compatibility with **tar**(1).

c

Cartridge. Set the defaults for cartridge instead of the standard half-inch reel. This sets the density to 1000BPI and the blocking factor to 126. Since **sfxdump** can automatically detect the end-of-media, only the blocking parameter normally has an effect. When cartridge tapes are used, and this option is not specified, **sfxdump** will slightly miscalculate the size of the tape. If the **b**, **d**, **s** or **t** options are specified with this option, their values will override the defaults set by this option.

d bpi

Tape density. Not normally required, as **sfxdump** can detect end-of-media. This parameter can be used to keep a running tab on the amount of tape used per reel. The default density is 6250BPI except when the *c* option is used for cartridge tape, in which case it is assumed to be 1000BPI per

track. Typical values for tape devices are:

1/2" tape

6250 BPI

1/4" cartridge

1000 BPI(The tape densities and other options are documented in the **st(7D)** man page.)

D

Diskette. Dump to diskette.

f dump_file

Dump file. Use *dump_file* as the file to dump to, instead of */dev/rmt/0*. If *dump_file* is specified as *-*, dump to standard output.

If the name of the file is of the form *machine:device*, the dump is done from the specified machine over the network using **rmt(1M)**. Since **sfxdump** is normally run by root, the name of the local machine must appear in the */.rhosts* file of the remote machine. If the file is specified as *user@machine:device*, **sfxdump** will attempt to execute as the specified user on the remote machine. The specified user must have a *.rhosts* file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.

I (lower-case i)

Autoload. When the end-of-tape is reached before the dump is complete, take the drive offline and wait up to two minutes for the tape drive to be ready again. This gives autoloading (stackloader) tape drives a chance to load a new tape. If the drive is ready within two minutes, continue. If it is not, prompt for another tape and wait.

L string

Sets the tape label to *string*, instead of the default none. *string* may be no more than fifteen characters long. If it is longer, it is truncated and a warning printed; the dump will still be done.

n

Notify all operators in the sys group that **sfxdump** requires attention by sending messages to their terminals, in a manner similar to that used by the **wall(1M)** command. Otherwise, such messages are sent only to the terminals (such as the console) on which the user running **sfxdump** is logged in.

o (lower-case o)

Offline. Take the drive offline when the dump is complete or the end-of-media is reached and rewind the tape, or eject the diskette. In the case of some autoloading 8mm drives, the tape is removed from the drive automatically. This prevents another process which rushes in to use the drive, from inadvertently overwriting the media.

s size

Specify the size of the volume being dumped to. Not normally required, as **sfxdump** can detect end-of-media. When the specified *size* is reached, **sfxdump** waits for you to change the volume. **sfxdump** interprets the specified size as the length in feet for tapes and cartridges, and as the number of 1024-byte blocks for diskettes. The values should be a little smaller than the actual physical size of the media (for example, 425 for a 450-foot cartridge). Typical values for tape devices depend on the **c** option, for cartridge devices, and the **D** option for diskettes:

1/2" tape

2300 feet

60-Mbyte 1/4" cartridge

425 feet

150-Mbyte 1/4" cartridge

700 feet

diskette

1422 blocks (Corresponds to a 1.44-Mbyte diskette, with one cylinder reserved for bad block information.)

S

Size estimate. Determine the amount of space that is needed to perform the dump without actually doing it, and display the estimated number of bytes it will take. This is useful with incremental dumps to determine how many volumes of media will be needed.

t tracks

Specify the number of tracks for a cartridge tape. Not normally required, as **sfxdump** can detect end-of-media. The default is 9 tracks. The **t** option is not compatible with the **D** option. Values for

B.2 Management Commands Specific to the GFS Local File System

Sun-supported tape devices are:

60-Mbyte 1/4" cartridge

9 tracks

150-Mbyte 1/4" cartridge

18 tracks

T *timeout* [hms]

Sets the amount of time to wait for an auto load command to complete. This function modifier is ignored unless the **l** function modifier has also been specified. The default time period to wait is two minutes. Specify time units with a trailing **h** (for hours), **m** (for minutes), or **s** (for seconds). The default unit is minutes.

u

Update the dump record. Add an entry to the file `/etc/sfxfs_dumpdates`, for each file system successfully dumped that includes the file system name, date, and dump level.

v

Verify. After each tape or diskette is written, verify the contents of the media against the source file system. If any discrepancies occur, prompt for new media, then repeat the dump/verification process. The file system must be unmounted. This option cannot be used to verify a dump to standard output.

w

Warning. List the file systems that have not been backed up within a day. This information is gleaned from the files `/etc/sfxfs_dumpdates` and `/etc/vfstab`. When the **w** option is used, all other options are ignored. After reporting, **sfxdump** exits immediately.

W

Warning with highlight. Similar to the **w** option, except that the **W** option includes all file systems that appear in `/etc/sfxfs_dumpdates`, along with information about their most recent dump dates and levels. File systems that have not been backed up within a day are highlighted.

OPERAND

The following operand is supported:

files_to_dump

Specifies the files to dump. Usually it identifies a whole GFS Local File System by its raw device name of the representative partition (for example, `/dev/sfdsk/gfs/rdisk/vol0306`). Incremental dumps (levels 1 to 9) of files changed after a certain date only apply to a whole file system. Alternatively, *files_to_dump* can identify individual files or directories. All files or directories are dumped, which is equivalent to a level 0 dump; however, `/etc/sfxfs_dumpdates` is not updated, even with the **u** option specified. In all cases, the files must be contained in the same file system, and the file system must be local to the system where **sfxdump** is being run.

files_to_dump is required and must be the last argument on the command line.

If no options are given, the default is `9uf /dev/rmt/0 files_to_dump`.

EXAMPLES

Example 1: A sample display of the **sfxdump** command.

To make a full dump of the GFS Local File System configured on `/dev/sfdsk/gfs/rdisk/vol0300` onto a 150 megabyte tape unit 0, use:

```
# sfxdump 0cfu /dev/rmt/0 /dev/sfdsk/gfs/rdisk/vol0300
```

To make and verify an incremental dump at level 5 of the GFS Local File System configured on `/dev/sfdsk/gfs/rdisk/vol0306`, on a 1/2" reel tape unit 1, use:

```
# sfxdump 5fuv /dev/rmt/1 /dev/sfdsk/gfs/rdisk/vol0306
```

EXIT STATUS

While running, **sfxdump** emits many verbose messages. **sfxdump** returns the following exit values:

0 Normal exit.

- 1 Startup errors encountered.
- 3 Abort - no checkpoint attempted.

FILES

/dev/rmt/0
 default unit to dump to

/etc/sfxfs_dumpdates
 dump date record

/etc/group
 to find group sys

/etc/hosts
 to gain access to remote system with drive

/etc/vfstab
 list of file systems

SEE ALSO

[sfxcpio\(1\)](#), [sfxrestore\(1M\)](#)
 cpio(1), tar(1), dd(1M), devnm(1M), prtvtoc(1M), rmt(1M), shutdown(1M), volcopy(1M), wall(1M), st(7D)

NOTES

Read errors

Up to 32 read errors on the file system are ignored.

Process Per Reel

Because each reel requires a new process, parent processes for reels that are already written hang around until the entire tape is written.

Operator intervention

sfxdump requires operator intervention on these conditions: end of volume, end of dump, volume write error, volume open error or disk read error (if there are more than a threshold of 32). In addition to alerting all operators implied by the n option, sfxdump interacts with the operator on **sfxdump**'s control terminal at times when **sfxdump** can no longer proceed, or if something is grossly wrong. All questions **sfxdump** poses must be answered by typing yes or no, as appropriate.

Since backing up a disk can involve a lot of time and effort, sfxdump checkpoints at the start of each volume. If writing that volume fails for some reason, **sfxdump** will, with operator permission, restart itself from the checkpoint after a defective volume has been replaced.

Suggested Dump Schedule

It is vital to perform full, "level 0", dumps at regular intervals. When performing a full dump, bring the machine down to single-user mode using **shutdown(1M)**. While preparing for a full dump, it is a good idea to clean the tape drive and heads. Incremental dumps should be performed with the system running in single-user mode.

Incremental dumps allow for convenient backup and recovery of active files on a more frequent basis, with a minimum of media and time. However, there are some tradeoffs. First, the interval between backups should be kept to a minimum (once a day at least). To guard against data loss as a result of a media failure (a rare, but possible occurrence), capture active files on (at least) two sets of dump volumes. Another consideration is the desire to keep unnecessary duplication of files to a minimum to save both operator time and media storage. A third consideration is the ease with which a particular backed-up version of a file can be located and restored. The following four-week schedule offers a reasonable tradeoff between these goals.

	Sun	Mon	Tue	Wed	Thu	Fri
Week 1: 0 (Full)	5		5	5	5	3
Week 2:		5	5	5	5	3
Week 3:		5	5	5	5	3
Week 4:		5	5	5	5	3

Although the Tuesday through Friday incremental contains "extra copies" of files from Monday, this scheme assures that any file modified during the week can be recovered from the previous day's incremental dump.

B.2 Management Commands Specific to the GFS Local File System

Process priority of the sfxdump command

sfxdump uses multiple processes to allow it to read from the disk and write to the media concurrently. Due to the way it synchronizes between these processes, any attempt to run dump with a nice (process priority) of `-5` or better will likely make **sfxdump** run slower instead of faster.

Partition overlap

The partition which covers the entire disk contains other partitions on the disk. But even if multiple GFS Local File Systems are configured on one disk, **sfxdump** must not be executed on the entire disk partition. **sfxdump** can handle only one GFS Local File System at a time. To dump the multiple GFS Local File Systems configured on one disk, **sfxdump** must be executed on each file systems.

Warning when the w and W options are specified

The `/etc/vfstab` file does not allow the desired frequency of backup for file systems to be specified. Consequently, the `w` and `W` options assume file systems should be backed up daily, which limits the usefulness of these options.

B.2.6 sfxedquota(1M) - Edits user quotas for the GFS Local File System

SYNOPSIS

```
sfxedquota [ -p proto_user ] username ...
sfxedquota -t
```

DESCRIPTION

sfxedquota is an editor for quota of a GFS Local File System. One or more users may be specified on the command line. For each user a temporary file is created with an ASCII representation of the current disk quotas for that user for each mounted GFS Local File System that has a quotas file, and an editor is then invoked on the file. The quotas may then be modified, new quotas added, etc. Upon leaving the editor, sfxedquota reads the temporary file and modifies the binary quota files to reflect the changes made.

The editor invoked is **vi(1)** unless the **EDITOR** environment variable specifies otherwise.

Only the super-user may edit quotas. In order for quotas to be established on a file system, the root directory of the file system must contain a file, owned by root, called quotas. (See [sfxquotaon\(1M\)](#).)

proto_user and *username* can be numeric, corresponding to the UID of a user. Unassigned UIDs may be specified; unassigned names may not. In this way, default quotas can be established for users who are later assigned a UID.

If no options are specified, the temporary file created will have one or more lines of the form

```
fs mount_point blocks (soft = number, hard = number) inodes (soft = number, hard = number)
```

Where a block is considered to be a 1024 byte (1K) block.

The number fields may be modified to reflect desired values.

OPTIONS

-p

Duplicate the quotas of the *proto_user* specified for each username specified. This is the normal mechanism used to initialize quotas for groups of users.

-t

Edit the soft time limits for each file system. If the time limits are zero, the default value used is 1 week. The temporary file created will have one or more lines of the form

```
fs mount_point blocks time limit = number tmunit, files time limit = number tmunit
```

tmunit may be one of **month**, **week**, **day**, **hour**, **min** or **sec**; characters appended to these keywords are ignored, so you may write **months** or **minutes** if you prefer. The number and tmunit fields may be modified to set desired values. Time limits are printed in the greatest possible time unit such that the value is greater than or equal to one. If **default** is printed after the tmunit, this indicates that the value shown is zero (the default).

EXAMPLES

```
# sfxedquota user1
# sfxedquota user1 user2
# sfxedquota -p protuser user1 user2
# sfxedquota -t
```

SEE ALSO

[sfxquota\(1M\)](#), [sfxquotadm\(1M\)](#), [sfxquotaon\(1M\)](#), [sfxrepquota\(1M\)](#)
vi(1), passwd(4)

B.2.7 sfxinfo(1M) - Displays GFS Local File System partition information

SYNOPSIS

```
sfxinfo [-V] [-n] directory | special
sfxinfo { -V } [-n] -d dev
```

DESCRIPTION

The **sfxinfo** command outputs the partition information for each GFS Local File System that was specified with the parameter *directory* or *special*.

Specify *special* with the special file name for one of the partitions that constitute the file system. The representative partition of the file system is displayed on top of the output result.

Specify the *directory* with the path name for the file system mounting point.

When specifying the path name for the mounting point, specify the file system in */etc/vfstab*.

GFS Local File System contains in each partition information on the file system component partitions. The **sfxinfo** command outputs the following detailed information:

When the file system in extension processing of a file system was specified with **sfxadd(1M)**, information to show what is handled in is indicated by a column of use of a partition. When the file system of state was specified during partition addition, it is indicated with TRANSIT(add). And there are "-----" and an indicated case about TRANSIT(add) and output size of line.

[Partition information]

- Partition ID
- Special file name (device number)
- Area size (kirobyte unit)
- Purpose of a partition (meta-data/data/log)

[Node information]

- Host ID
- Host name

OPTIONS

-d dev

This option displays partition information for a specific device number.
The device number is specified by hexadecimal number.

-n

This option displays node information.

-V

This option displays the version information of the GFS Local File System.

EXAMPLES

```
# sfxinfo /dev/sfdsk/gfs/dsk/vol0301
ID  special                size  mount  Type
0   /dev/sfdsk/gfs/dsk/vol0301(3f40021)  25986 /mnt   META
0   /dev/sfdsk/gfs/dsk/vol0301(3f40021)  105751 /mnt   DATA
1   /dev/sfdsk/gfs/dsk/vol0302(3f40022)   5120 /mnt   LOG
2   /dev/sfdsk/gfs/dsk/vol0303(3f40023)  131736 /mnt   DATA
# sfxinfo -V /dev/sfdsk/gfs/dsk/vol0301
sfxfs v1.1 file system
ID  special                size  mount  Type
0   /dev/sfdsk/gfs/dsk/vol0301(3f40021)  25986 /mnt   META
0   /dev/sfdsk/gfs/dsk/vol0301(3f40021)  105751 /mnt   DATA
1   /dev/sfdsk/gfs/dsk/vol0302(3f40022)   5120 /mnt   LOG
2   /dev/sfdsk/gfs/dsk/vol0303(3f40023)  131736 /mnt   DATA
```

B.2 Management Commands Specific to the GFS Local File System

```
# sfxinfo /mnt
ID    special                                size  mount  Type
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)    12626 /mnt   META
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)    113992 /mnt   DATA
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)    5120 /mnt   LOG
# sfxinfo -n /dev/sfdsk/gfs/dsk/vol0301
/dev/sfdsk/gfs/dsk/vol0301:
ID    hostID      hostname
0     807b1f69    node1
1     807b1f70    node2
```

Indication example when file system extension by partition addition.

```
# sfxinfo /dev/sfdsk/gfs/dsk/vol0301
ID    special                                size  mount  Type
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)    12626 /mnt   META
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)    113992 /mnt   DATA
0     /dev/sfdsk/gfs/dsk/vol0301(3f40021)    5120 /mnt   LOG
1     /dev/sfdsk/gfs/dsk/vol0303(3f40023)    ----- /mnt   DATA TRANSIT(add)
```

SEE ALSO

[sfxadd\(1M\)](#)

B.2.8 sfxmkbkup(1M), sfxmkrstr(1M) – Creates a shell script for backing up

SYNOPSIS

```
sfxmkbkup [ -a ] [ -c ] [ -b n ] [ -o filename ] special | mount_point backup_dev
sfxmkbkup -e [ -c ] [ -b n ] special | mount_point
sfxmkrstr [ -a ] [ -c ] [ -b n ] [ -o filename ] special | mount_point backup_dev
```

DESCRIPTION

The **sfxmkbkup** command creates a shell script to obtain backup data of the GFS Local File System specified by the representative device name *special* or mount point *mount_point*.

sfxmkbkup creates two types of scripts: a backup script that backs up data for each volume with **dd**(1M) and a restore script that restores data for each volume with **sfxddr**(1M).

sfxmkrstr creates only a restore script.

Super-user authority (root) is required to execute the **sfxmkbkup** command and run shell scripts created by the **sfxmkbkup** command.

OPTIONS

-a

This option associates backup data for one medium when the data is output. In this case, a device indicated by no-rewind must be specified for *backup_dev*.

-c

This option creates a script that compresses data during a backup with **compress**(1) and uncompresses data when the data is restored.

-b n

Specify the block size of the backup medium in n bytes. The character string specified in this option is specified as obs during backup with **dd**(1M) or as ibs during restoration with **sfxddr**(1M). As with ibs for **dd**(1M), each numeric value can be suffixed by **k** or **b**, and multiplication can be indicated with **x**. The default is 256k.

-o filename

Specify a file name for a shell script that will be created. The backup and restore scripts are created with path names suffixed by ".bkup" and ".rstr," respectively. The default is "sfxfs."

-e

This option displays indicates the backup size. No shell script is created.

special

Specify a special file name for the representative partition of the GFS Local File System.

mount_point

This option specifies the mount point for the GFS Local File System.

backup_dev

This option specifies a backup destination device.

EXAMPLES

To back up the GFS Local File System to a tape unit using special file /dev/sfdsk/gfs/dsk/vol1301 as a representative partition, execute the following command:

```
# sfxmkbkup /dev/sfdsk/gfs/dsk/vol1301 /dev/rmt/0
```

The backup script below, sfxfs.bkup, is created in the current work directory.

B.2 Management Commands Specific to the GFS Local File System

```
#!/bin/sh
IBS=1024k
OBS=256k
TAPE=/dev/rmt/0
SPECS="/dev/sfdsk/gfs/rdisk/vol1301 /dev/sfdsk/gfs/rdisk/vol1401 /dev/sfdsk/gfs/rdisk/vol1501"

n=`echo $SPECS | /usr/bin/wc -w`
echo "It will need $n volume(s) to backup."

i=1
for s in $SPECS
do
    echo "Backup $s"
    echo "Prepare volume #i/$n for $TAPE and press RETURN:¥c"
    read yes
    until /usr/bin/dd if=$s of=$TAPE ibs=$IBS obs=$OBS
    do
        echo "Press RETURN and retry:¥c"
        read yes
    done
    i=`/usr/bin/expr $i + 1`
done

echo "Backup done."
```

The restore script sfxfs.rstr is as follows:

```
#!/bin/sh
IBS=256k
OBS=1024k
TAPE=/dev/rmt/0
SPECS="/dev/sfdsk/gfs/rdisk/vol1301 /dev/sfdsk/gfs/rdisk/vol1401 /dev/sfdsk/gfs/rdisk/vol1501"
n=`echo $SPECS | /usr/bin/wc -w`
echo "It will need $n volume(s) to restore."

i=1
for s in $SPECS
do
    if echo $s | /usr/bin/grep "^!"; then
        s=`echo $s | /usr/bin/cut -c2-`
        opt="-f"
    else
        opt=""
    fi
    echo "Restore $s"
    echo "Prepare volume #i/$n for $TAPE and press RETURN:¥c"
    read yes
    until /usr/lib/fs/sfxfs/sfxddr $opt if=$TAPE of=$s ibs=$IBS obs=$OBS
    do
        echo "Change to CORRECT volume #i/$n and press RETURN:¥c"
        read yes
    done
    i=`/usr/bin/expr $i + 1`
done

echo "Restore done."
```

B.2.8 sfxmkbkup(1M), sfxmkrstr(1M) – Creates a shell script for backing up

To back up and restore data using multiple tape drives, execute **sfxmkbkup** for the number of tape drives shown below, and edit the value of the SPECS variable in the shell script. Then allocate partitions to the sfxfs. bkup and sfxfs.rstr shell scripts.

```
# sfxmkbkup -o ex1 /dev/sfdsk/gfs/dsk/vol1301 /dev/rmt/0
# sfxmkbkup -o ex2 /dev/sfdsk/gfs/dsk/vol1301 /dev/rmt/1
```

To restore some partitions to a different disk from that from which the data was backed up, edit the value of the SPECS variable for the restore script. In this case, be sure to specify an exclamation point, which indicates that volume information is not checked, just before the edited special file name.

```
SPECS="/dev/sfdsk/gfs/rdisk/vol1301 !/dev/sfdsk/gfs/rdisk/vol2101 /dev/sfdsk/gfs/rdisk/vol1501"
```

SEE ALSO

[sfxddr\(1M\)](#)

NOTES

- If the backed-up data is restored to a disk other than the one from which the backup was made, the structure of the disk on which the data is restored must be the same as that of the original disk.
- Do not specify a partition that includes the first cylinder of the disk as the partition to which another partition data is restored. If this is done by mistake, the disk's configuration information will be destroyed and the disk will not be able to be used in the current configuration.
- Once the backed-up data has been restored into to disk other than the original one, the partition information changed with **sfxadm(1M)**. Specify all partitions in the GFS Local File System and execute **sfxadm(1M)**.

B.2.9 sfxnewfs(1M) – Builds the GFS Local File System

SYNOPSIS

sfxnewfs [**-Nv**] [*mkfs-options*] [**-L** *raw_device_file*] [**-D** *raw_device_file,...*] *raw_device_file*

DESCRIPTION

The **sfxnewfs**(1M) is the **mkfs_sfxfs**(1M) front end command for creating the GFS Local File System. It performs calculation with the appropriate parameters and calls **mkfs_sfxfs**(1M).

If **sfxnewfs**(1M) is executed interactively (via standard input from the terminal), the required checks are performed before creating the file system.

OPTIONS

-N

This option displays the parameter that was used to create the file system.
The file system is not actually created by this command.

-v

Verbose. **sfxnewfs** prints out its actions, including the parameters passed to **mkfs**.

mkfs-options

The default parameters that are to be passed to **mkfs**(1M) can be changed with the following options:

-i nbpi

This option specifies the number of bytes per i-node.
The default value is 8,192.
Increase this value when saving the meta-data area.
Values from 512 to 1,048,576 can be used here.

-f free

This option specifies the minimum percentage for the free area of the file system.
The default value is 10%.
Values from 0 to 99 can be used here.

-m mfree

This option specifies the minimum percentage for the free area of V data.
The default value is 10%.
Values from 0 to 99 can be used here

-o opt

This option specifies whether to optimize area or time.
The parameter space indicates the area, time indicates the time.
The default value is time.

-n maxnode

This option specifies the maximum number of nodes sharing the file system, including also the case that the file system is in the future expanded with **sfxnode**(1M) command.
The default value is 16.
Values from 1 to 16 can be used here.

-l maxvol

This option species the maximum number of file system component partitions, including also the case that the file system is in the future expanded with the **sfxadd**(1M) command.
The default value is 16.
Values from 1 to 32 can be used here.

-d maxdsz

This option specifies the maximum size of the data areas for each file system, including also the case that a data partition is to be added.
This parameter is to be specified in megabyte.
A value larger than 1 terabyte (1,048,576 megabytes) cannot be used here.

-a metasz

This option specifies the size of each meta-data area.

This parameter is to be specified in megabyte.

A value that exceeds the size of the representative partition cannot be used here.

-g *logsz*

This option specifies the size of each update log area.

A value from 5 to 100 can be used here.

-e *nblkpext*

This option specifies the average number of blocks that have to be managed in each extent.

This value can be used to calculate data area fragmentation.

The default value is 1.

A value from 1 to 100 can be used here.

-A

This option specifies a configuration in which a data area is not included in a representative partition if a data partition is specified.

-L *raw_device_file*

This option specifies a character special device for update log partition. Specify a partition that accommodates the size of the update log area, because, except the area used as an update log area, the area of update log partition remains unused.

-D *raw_device_file,...*

This option specifies a character special device for data partition. When specifying two or more files, delimit them by a comma ",".

OPERANDS

raw_device_file

This parameter specifies the character special device name for the representative partition.

EXAMPLES

```
# sfxnewfs /dev/sfdsk/gfs/rdsk/vol0301

# sfxnewfs -L /dev/sfdsk/gfs/rdsk/vol0601 /dev/sfdsk/gfs/rdsk/vol0301

# sfxnewfs -D /dev/sfdsk/gfs/rdsk/vol0400,/dev/sfdsk/gfs/rdsk/vol0500 /dev/sfdsk/gfs/rdsk/vol0301
```

SEE ALSO

[fscck_sfxfs\(1M\)](#), [mkfs_sfxfs\(1M\)](#), [sfxadd\(1M\)](#), [sfxnode\(1M\)](#), [sfx tune\(1M\)](#), [fs_sfxfs\(4\)](#)

B.2.10 sfxnode(1M) - Adds or changes GFS Local File System node configuration information

SYNOPSIS

```
sfxnode raw_device_file,...  
sfxnode -a -n node -i hostid raw_device_file  
sfxnode -d [ -n node | -i hostid ] raw_device_file
```

DESCRIPTION

The **sfxnode** command adds or changes the GFS Local File System node information.

When adding nodes that share a GFS Local File System, execute the **sfxnode** command on these nodes; this will modify the node configuration information (host ID or name). The number of each special device as seen from the node is also set in the partition configuration information.

To delete node information, specify the **-d** option. Using the **-n** option, specify the host name of a node to be deleted. Alternatively, using the **-i** option, specify the host ID to be deleted. If neither the **-n** option nor the **-i** option is specified, node information of the node for which the **sfxnode** command is executed is deleted. In the *raw_device_file* field, specify a character special device for the representative partition.

When adding nodes that share a GFS Local File System, if a device cannot be opened on the switched node of the switched file system, specify the **-a** option to add node information. In this case, use the **-n** option to specify a host name of the switched node, or use the **-i** option to specify the host ID of the switched node. This option is specified on an active node. After specifying this option, specify the **-A** option with the **sfxadm** command to add partition configuration information of the GFS Local File System corresponding to the switched node.

When the **sfxnode** is used to change partition configuration information for the GFS Local File System, the entry in */etc/sfxfstab* will be updated and the information on the each partition will be changed.

OPTIONS

The following options can be specified:

- a**
Adds node information.
- d**
Deletes node information.
- n** *node*
This option specifies a host name.
- i** *hostid*
This option specifies a host ID.

OPERANDS

The following operands can be specified:

***raw_device_file*,...**

When add own node information to GFS Local File System (no option case), this operand specifies all character special devices of all partition; that specified the representative partition at the first; that constitute the GFS Local File System on the nodes.

raw_device_file

When option is specified, this operand specifies the character special device for the representative partition.

EXAMPLES

Add own node information for GFS Local File System (3 partitions configuration)

```
# sfxnode /dev/sfdsk/gfs/rdisk/vol0301,/dev/sfdsk/gfs/rdisk/vol0401,/dev/sfdsk/gfs/rdisk/vol0501
```

Add specified node information for GFS Local File System

```
# sfxnode -a -n moony -i 0x80fca402 /dev/sfdsk/gfs/rdisk/vol0301
```

Delete specified node information for GFS Local File System

```
# sfxnode -d -n moony /dev/sfdsk/gfs/rdisk/vol0301
```

NOTES

Although the GFS Local File System can be used on all nodes where the **sfxnode** command is executed, access it only from one of the two nodes. In other words, mount and use the GFS Local File System that is to be shared by the nodes only on one of these nodes.

If the **sfxnode** command failed and displayed the message "node full can't sfxnode", the number of registered nodes has already amounted to the maximum number of sharing nodes that was specified to create the file system. After checking the number of the registered nodes by using the **sfxinfo** command and deleting unnecessary node information by using the **sfxnode** command with **-d** option, add the node information.

FILE

/etc/sfxfstab

This file contains the names of each of the file system component partitions.

SEE ALSO

[mkfs_sfxfs\(1M\)](#), [sfxadd\(1M\)](#), [sfxadm\(1M\)](#), [sfxnewfs\(1M\)](#)

B.2.11 sfxquota(1M) - Displays a user's GFS Local File System disk quota and usage

SYNOPSIS

```
sfxquota [-v] [ username ]
```

DESCRIPTION

sfxquota displays users' GFS Local File System disk usage and limits. Only the super-user may use the optional *username* argument to view the limits of other users.

sfxquota without options displays only warnings if the usage is above quota limits for mounted GFS Local File Systems.

A login name or user ID can be specified as a *username*. If the *username* is omitted, the information of the user executing the **sfxquota** command is displayed.

OPTION

-v

Display user's quotas on all mounted GFS Local File Systems where quotas exist.

EXAMPLES

```
# sfxquota user1
Over file quota on /sfxfs, remove 2 files within 7.0 days
# sfxquota -v user1
Disk quotas for user1 (uid 1000):
Filesystem      usage  quota  limit  timeleft  files  quota  limit  timeleft
/sfxfs2         2258   3072   5120             71     80    100
% sfxquota -v
```

SEE ALSO

[sfxedquota\(1M\)](#), [sfxquotadm\(1M\)](#), [sfxquotaon\(1M\)](#), [sfxrepquota\(1M\)](#)
[passwd\(4\)](#)

NOTE

sfxquota does not report quotas for NFS mounted GFS Local File Systems.

B.2.12 sfxquotadm(1M) - Manages quotas file for the GFS Local File System

SYNOPSIS

```
sfxquotadm -i filsys
sfxquotadm -e filsys
sfxquotadm -c filsys
sfxquotadm -d filsys
```

DESCRIPTION

The **sfxquotadm** command manages the quotas file of the specified GFS Local File System.

The **sfxquotadm** command can copy the limits of all users between external quotas and internal quotas file, compare the limits, and delete internal quotas file.

The **sfxquotadm** command is used when external quotas file are deleted or when restoration has overwritten external quotas file.

The *filsys* must be mounted. In addition, the quotas file (external quotas) must be existed in the root directory. Amount point or block special file can be specified for *filsys*.

Only a super user can execute the **sfxquotadm** command.

OPTIONS

-i

This option reflects the limit values of all users of an external quotas file to an internal quotas file.

-e

This option reflects the limit values of all users of an internal quotas file to an external quotas file.

-c

This option compares the limit values of all users between external quotas and internal quotas file.

For a super user (user ID 0), the restricted times of the block and file counts when the software limit is exceeded are compared. If the user limit values between files do not match, the user ID and contents are displayed.

Display examples are given below. The contents of an external quotas file are indicated by [**<**]. The contents of an internal quotas file are indicated by [**>**].

```
--- uid = root(0) ---
< blocks time limit = 0 (default), files time limit = 0 (default)
> blocks time limit = 1.00 week, files time limit = 12.00 days
--- uid = 1000 ---
< block limits (soft = 0, hard = 0) file limits (soft = 0, hard = 0)
> block limits (soft = 0, hard = 5120) file limits (soft = 0, hard = 100)
--- uid = 2100 ---
< block limits (soft = 0, hard = 5120) file limits (soft = 0, hard = 100)
> block limits (soft = 3072, hard = 5120) file limits (soft = 80, hard = 100)
```

-d

This option delete internal quotas file.

Internal quotas file can be deleted only when the quota function is off.

EXAMPLES

```
# sfxquotadm -i /mnt
# sfxquotadm -e /mnt
# sfxquotadm -c /mnt
# sfxquotadm -d /dev/sfdsk/gfs/dsk/vol2100
```

SEE ALSO

[sfxedquota\(1M\)](#), [sfxquota\(1M\)](#), [sfxquotaon\(1M\)](#), [sfxrepquota\(1M\)](#)

B.2.13 `sfxquotaon(1M)`, `sfxquotaoff(1M)` - Turns GFS Local File System quotas on and off

SYNOPSIS

```
sfxquotaon [ -v ] filesystem ...  
sfxquotaon [ -v ] -a
```

```
sfxquotaoff [ -v ] filesystem ...  
sfxquotaoff [ -v ] -a
```

DESCRIPTION

The `sfxquotaon` turns on disk quotas for one or more GFS Local File Systems.

Before turn on disk quotas for a file system, a file named `quotas`, owned by root, must exist in the root directory of the file system. See [sfxedquota\(1M\)](#) for details on how to modify the contents of this file.

The `sfxquotaoff` turns off disk quotas for one or more GFS Local File Systems.

The file systems specified must already be mounted.

These commands update the `mntopts` field of the appropriate entries in `/etc/mnttab` to indicate when the quota function are on or off for each file system. If the quota function is on, "**quota**" will be added to `mntopts`. if the quota function is off, "**noquota**" will be added to `mntopts`.

`filesystem` must be either the mount point of a file system, or the block device that configures the file system.

OPTIONS

`sfxquotaon`

-a

Enable quotas for GFS Local File System which "**rq**" or "**quota**" are marked in the `mntopts` field of `/etc/vfstab`, are currently mounted as read-write, and have a `quotas` file in the root directory.

-v

Display a message for each file system after quotas are turned on.

`sfxquotaoff`

-a

Force all GFS Local File Systems in `/etc/mnttab` to have their quotas disabled.

-v

Display a message for each file system after quotas are turned off.

EXAMPLES

```
# sfxquotaon /mnt  
# sfxquotaon -a  
# sfxquotaoff /mnt  
# sfxquotaoff -a
```

SEE ALSO

[sfxedquota\(1M\)](#), [sfxquota\(1M\)](#), [sfxquotadm\(1M\)](#), [sfxrepquota\(1M\)](#)
`mnttab(4)`, `vfstab(4)`

B.2.14 sfxrepquota(1M) - Summarizes quotas for the GFS Local File System

SYNOPSIS

```
sfxrepquota [ -v ] filesystem ...  
sfxrepquota [ -v ] -a
```

DESCRIPTION

sfxrepquota prints a summary of the disk usage and quotas for the specified GFS Local File Systems. The current number of files and amount of space (in kilobytes) is printed for each user along with any quotas created with **sfxedquota(1M)**.

The filesystem must have the file quotas in its root directory.

Only the super-user may view quotas which are not their own.

OPTIONS

- a**
Report on all mounted GFS Local File Systems that have "rq" or "quota" in the mntopts field of the /etc/vfstab file.
- v**
Report quotas for all users, even those who do not consume resources.

EXAMPLES

```
# sfxrepquota /sfxfs  
/dev/sfdsk/gfs/dsk/vol1400 (/sfxfs):  
          Block limits          File limits  
User      used  soft  hard  timeleft  used  soft  hard  timeleft  
user1    --  2258 3072 5120          71   80  100  
user2    --  1345 3072 5120          38   80  100  
user3    ++  3424 3072 5120  7.0 days    81   80  100  7.0 days  
# sfxrepquota -v /sfxfs  
/dev/sfdsk/gfs/dsk/vol1400 (/sfxfs):  
          Block limits          File limits  
User      used  soft  hard  timeleft  used  soft  hard  timeleft  
user1    --  2258 3072 5120          71   80  100  
user2    --  1345 3072 5120          38   80  100  
user3    ++  3424 3072 5120  7.0 days    81   80  100  7.0 days  
user4    --    0 3072 5120           0   80  100  
# sfxrepquota -v -a
```

SEE ALSO

[sfxedquota\(1M\)](#), [sfxquota\(1M\)](#), [sfxquotadm\(1M\)](#), [sfxquotaon\(1M\)](#)

B.2.15 sfxrestore(1M) - Restores from the dump created with sfxdump

SYNOPSIS

sfxrestore *i|r|t|x* [*abcdefghijklmsvy*] [*archive_file*] [*factor*] [*e_opt*] [*dump_file*] [*n*] [*filename...*]

DESCRIPTION

The **sfxrestore** utility restores files from backup media created with the **sfxdump** command. **sfxrestore**'s actions are controlled by the key argument. The key is exactly one function letter (**i**, **r**, **t**, or **x**) and zero or more function modifiers (letters). The key string contains no SPACE characters. Function modifier arguments are listed on the command line in the same order as their corresponding function modifiers appear in the key string.

The **sfxrestore** also operates on a system other than the GFS Local File System. For information about the operations to be performed when the dump created with **sfxdump** is restored on a file system other than GFS Local File System or the relationships between **sfxrestore** and **ufsrestore**, see NOTES below.

OPTIONS

[Function letters]

One (and only one) of the following function letters is required:

i

Interactive. After reading in the directory information from the media, **sfxrestore** invokes an interactive interface that allows you to browse through the dump file's directory hierarchy and select individual files to be extracted. See Interactive commands, below, for a description of available commands.

r

Recursive. Restore the entire contents of the media into the current directory (which should be the top-level of the file system). To completely restore a file system, use this function letter to restore the level 0 dump, and again for each incremental dump. Although, this function letter is intended for a complete restore onto a clear file system, if the file system contains files not on the media, they are preserved.

t

Table of contents. List each filename that appears on the media. If no filename argument is given, the root directory is listed. This results in a list of all files on the media, unless the **h** function modifier is in effect. The table of contents is taken from the media or from the specified archive file, when the function modifier is used.

x

Extract the named files from the media. If a named file matches a directory whose contents were written onto the media, and the **h** function modifier is not in effect, the directory is recursively extracted. The owner, modification time, and mode are restored (if possible). Existing files are overwritten and a warning is given. If no filename argument is given, the root directory is extracted. This results in the entire tape being extracted unless the **h** function modifier is in effect.

[Function modifiers]

a *archive_file*

Read the table of contents from *archive_file* instead of the media. This function modifier can be used in combination with the **t**, **i** or **x** function letters, making it possible to check whether files are on the media without having to mount the media. When used with the **x** and interactive (**i**) function letters, it prompts for the volume containing the file(s) before extracting them.

b *factor*

Blocking factor. Specify the blocking factor for tape reads. For variable length SCSI tape devices, unless the data was written with the default blocking factor, a blocking factor at least as great as that used to write the tape must be used; otherwise, an error will be generated. Note that a tape block is 512 bytes. Refer to the man page for your specific tape driver for the maximum blocking factor.

d

Turn on debugging output.

B.2 Management Commands Specific to the GFS Local File System

e *e_opt*

Specifies how to handle files that have extent attribute information. It's impossible to set the information onto files if the destination file system does not support extent attribute information or lacks free extents appropriate to satisfy the extent attribute information requirements. This function modifier specifies the action in those cases. Valid values for *e_opt* are:

force

Fails to restore files if extent attribute information could not be set.

ignore

Ignores to extent attribute information entirely.

warn

Issues a warning message if extent attribute information could not be set to a file. The extent attribute information of the file is erased.

The default value for *e_opt* is **warn**.

f *dump_file*

Use *dump_file* instead of `/dev/rmt/0` as the file to restore from. Typically *dump_file* specifies a tape or diskette drive. If *dump_file* is specified as '-', **sfxrestore** reads from the standard input. This allows **sfxdump(1M)** and **sfxrestore** to be used in a pipeline to copy a file system:

```
# sfxdump 0f - /dev/sfdsk/gfs/rdsk/vol0007 | (cd /home;sfxrestore xf -)
```

If the name of the file is of the form *machine:device*, the restore is done from the specified machine over the network using **rmt(1M)**. Since **sfxrestore** is normally run by root, the name of the local machine must appear in the `/.rhosts` file of the remote machine. If the file is specified as *user@machine:device*, **sfxrestore** will attempt to execute as the specified user on the remote machine. The specified user must have a `.rhosts` file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.

h

Extract or list the actual directory, rather than the files that it references. This prevents hierarchical restoration of complete sub trees from the tape.

I (lower-case i)

Autoload. When the end-of-tape is reached before the restore is complete, take the drive off-line and wait up to two minutes (the default, see the **T** function modifier) for the tape drive to be ready again. This gives autoloading (stackloader) tape drives a chance to load a new tape. If the drive is ready within two minutes, continue. If it is not, prompt for another tape and wait.

L *string*

The string as label that should appear in the header of the dump file. If the labels do not match, **sfxrestore** issues a diagnostic and exits.

m

Extract by i-node numbers rather than by filename to avoid regenerating complete pathnames. Regardless of where the files are located in the dump hierarchy, they are restored into the current directory and renamed with their i-node number. This is useful if only a few files are being extracted.

o (lower-case o)

Offline. Take the drive off-line when the restore is complete or the end-of-media is reached and rewind the tape, or eject the diskette. In the case of some autoloading 8mm drives, the tape is removed from the drive automatically.

s *n*

Skip to the *n*'th file when there are multiple dump files on the same tape. For example, the command:

```
# sfxrestore xfs /dev/rmt/0hn 5
```

would position you to the fifth file on the tape when reading volume 1 of the dump. If a dump extends over more than one volume, all volumes except the first are assumed to start at position 0, no matter what "**s n**" value is specified. If "**s n**" is specified, the backup media must be at BOT (beginning of tape). Otherwise, the initial positioning to read the table of contents will fail, as it is performed by skipping the tape forward *n*-1 files rather than by using absolute positioning. This is because on some devices absolute positioning is very time consuming.

T *timeout [hms]*

Sets the amount of time to wait for an autoload command to complete. This function modifier is ignored unless the **I** function modifier has also been specified. The default timeout period is two

minutes. The time units may be specified as a trailing **h** (hours), **m** (minutes), or **s** (seconds). The default unit is minutes.

v

Verbose. **sfxrestore** displays the name of each file it restores, preceded by its file type.

y

Do not ask whether to abort the restore in the event of tape errors. **sfxrestore** tries to skip over the bad tape block(s) and continue as best it can.

[Interactive Commands]

sfxrestore enters interactive mode when invoked with the **i** function letter. Interactive commands are reminiscent of the shell. For those commands that accept an argument, the default is the current directory. The interactive options are:

add [filename]

Add the named file or directory to the list of files to extract. If a directory is specified, add that directory and its files (recursively) to the extraction list (unless the **h** function modifier is in effect).

cd directory

Change to directory (within the dump file).

delete [filename]

Delete the current directory, or the named file or directory from the list of files to extract. If a directory is specified, delete that directory and all its descendents from the extraction list (unless the **h** function modifier is in effect). The most expedient way to extract a majority of files from a directory is to add that directory to the extraction list, and then delete specific files to omit.

extract

Extract all files on the extraction list from the dump media. **sfxrestore** asks which volume the user wishes to mount. The fastest way to extract a small number of files is to start with the last volume and work toward the first.

help

Display a summary of the available commands.

ls [directory]

List files in directory or the current directory, represented by a **`.`** (period). Directories are appended with a **`.`** (slash). Entries marked for extraction are prefixed with a **`.`** (asterisk). If the verbose option is in effect, i-node numbers are also listed.

marked [directory]

Like **ls**, except only files marked for extraction are listed.

paginate

Toggle the pagination of the output from the **ls** and **marked** commands. The **paginate** used is that defined by the **PAGER** environment variable, or **more(1)** if that envvar is not defined. The **PAGER** envvar may include white-space-separated arguments for the pagination program.

pwd

Print the full pathname of the current working directory.

quit

sfxrestore exits immediately, even if the extraction list is not empty.

setmodes

Prompts: set owner/mode for **`.`** (period). Type **y** for yes to set the mode (permissions, owner, times) of the current directory **`.`** (period) into which files are being restored equal to the mode of the root directory of the file system from which they were dumped. Normally, this is what you want when restoring a whole file system, or restoring individual files into the same locations from which they were dumped. Type **n** for no, to leave the mode of the current directory unchanged. Normally, this is what you want when restoring part of a dump to a directory other than the one from which the files were dumped.

setpager command

Sets the command to use for paginating output instead of the default or that inherited from the environment. The command string may include arguments in addition to the command itself.

verbose

Toggle the status of the **v** function modifier. While **v** is in effect, the **ls** command lists the i-node numbers of all entries, and **sfxrestore** displays information about each file as it is extracted.

B.2 Management Commands Specific to the GFS Local File System

what

Display the dump header on the media.

OPERAND

filename

Specifies the pathname of files (or directories) to be restored to disk. Unless the **h** function modifier is also used, a directory name refers to the files it contains, and (recursively) its subdirectories and the files they contain. *filename* is associated with either the **x** or **t** function letters, and must come last.

EXIT STATUS

The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred. Verbose messages are displayed.

FILES

/dev/rmt/0

The default tape drive.

*/tmp/rstdir**

File containing directories on the tape. The directory where this file is created can be specified by setting the environment variable **TMPDIR**.

*/tmp/rstmode**

Owner, mode, and timestamps for directories. The directory where this file is created can be specified by setting the environment variable **TMPDIR**.

./restoresymtable

Information passed between incremental restores.

ENVIRONMENT

PAGER

The command to use as a filter for paginating output. This can also be used to specify the options to be used. Default is **more(1)**.

TMPDIR

sfxrestore normally creates temporary files in the directory */tmp*. You may specify another directory by setting the environment variable **TMPDIR** to your chosen directory. (If **TMPDIR** isn't a valid directory, then **sfxrestore** will use */tmp*).

SEE ALSO

[sfxdump\(1M\)](#)

[mkfs\(1M\)](#), [mount\(1M\)](#), [rmt\(1M\)](#)

DIAGNOSTICS

sfxrestore complains about bad option characters.

Read errors result in complaints. If **y** has been specified, or the user responds **y**, **sfxrestore** will attempt to continue.

If the dump extends over more than one tape, **sfxrestore** asks the user to change tapes. If the **x** or **i** function letter has been specified, **sfxrestore** also asks which volume the user wishes to mount. If the **s** function modifier has been specified, and volume 1 is mounted, it is automatically positioned to the indicated file.

There are numerous consistency checks that can be listed by **sfxrestore**. Most checks are self-explanatory or can "never happen." Common errors are given below.

filename: not found on tape

The specified file name was listed in the tape directory, but was not found on the tape. This is caused by tape read errors while looking for the file, or from using a dump tape created on an active file system.

expected next file *inumber*, got *inumber*

A file that was not listed in the directory showed up. This can occur when using a dump tape created on an active file system.

Incremental tape too low

When doing an incremental restore, a tape that was written before the previous incremental tape, or that has too low an incremental level has been loaded.

Incremental tape too high

When doing incremental restore, a tape that does not begin its coverage where the previous incremental tape left off, or one that has too high an incremental level has been loaded.

media read error: invalid argument

Blocking factor specified for read is smaller than the blocking factor used to write data.

Tape read error while restoring *filename*

Tape read error while skipping over *inode inumber*

Tape read error while trying to resynchronize

A tape read error has occurred

If a file name is specified, then its contents are probably partially wrong. If an i-node is being skipped or the tape is trying to resynchronize, then no extracted files have been corrupted, though files may not be found on the tape.

resync restore, skipped *num*

After a tape read error, **sfxrestore** may have to resynchronize itself. This message lists the number of blocks that were skipped over.

Incorrect tape label. Expected *'foo'* got *'bar'*

The **L** function modifier was specified, and its value did not match what was recorded in the header of the dump file.

NOTES

sfxrestore can get confused when doing incremental restores from dump tapes that were made on active file systems.

A "level 0" dump must be done after a full restore. Because **sfxrestore** runs in user mode, it has no control over i-node allocation. This means that **sfxrestore** repositions the files, although it does not change their contents. Thus, a full dump must be done to get a new set of directories reflecting the new file positions, so that later incremental dumps will be correct.

Internal quotas file is not restored by **sfxrestore**. If the file system on which quota function is used, it's necessary to reflect the contents of external quotas file to internal quotas file by executing **sfxquotadm** with **-i** option. For details of the [sfxquotadm\(1M\)](#), see the reference manual.

sfxrestore fails if the file larger than 2 gigabytes is restored on the file system which does not support files larger than 2 gigabytes .

sfxrestore and **ufsrestore** are functionally compatible, though the backup formats are different. As a result, the **sfxrestore** cannot use the backup data created using the **ufsdump** for restoration. In the same way, **ufsrestore** cannot use the backup data created using **sfxdump** for restoration.

If files are restored on the file system that does not support extent attribute information such as UFS, extent attribute information is erased. If files are restored on the file system that does not support ACLs, ACLs are erased.

B.2.16 sfxstat(1M) - Reports statistical information on the GFS Local File System

SYNOPSIS

```
sfxstat [ -abcdfilmA ] [ interval [ count ] ] [ mount_point ... ]
```

DESCRIPTION

The **sfxstat** command displays the statistical information on the GFS Local File System.

sfxstat first reports statistical information recorded after the file system is mounted. The next time, it reports the information recorded since the last report.

When the parameter *interval* is specified, **sfxstat** reports statistical information for the interval that was specified in this parameter (in seconds). When no *interval* specified, it reports on statistical information only once.

When the parameter *count* is specified, **sfxstat** reports statistical information for the interval it takes to count to count. When no *count* specified, it keeps displaying statistical information.

When the parameter *mount_point* is specified, **sfxstat** reports statistical information on the specified file system. When no *mount_point* specified, **sfxstat** reports statistical information on all mounted GFS Local File System.

OPTIONS

When the option is omitted, reports all available information.

-a

This option reports the following information with respect to the manipulation of each allocated extent:

- Search request count
- Acquisition count
- Number of acquired blocks
- Number of acquisition failures
- Release count
- Number of released blocks
- Number of times that it was possible to concatenate released blocks
- Number of times where a fragmentation occurred through free extent management
- Number of times where a concatenation occurred through free extent management

-b

This option reports the following information:

- Number of times of buf acquisition by meta-management
- Number of times of dynamic buf acquisition
- Number of times of sleep during buf acquisition request

-c

This option reports the following information:

- Access count for each type of meta-cache
- Cache hit count for each type of meta-cache
- Number of updates for each type of meta-cache
- Read count for each meta-type
- Write count for each meta-type
- Writing time for each meta-type
- Average time for each meta-type

-d

This option reports the following information on direct I/O:

- Read count
- Write count
- Number of reading blocks
- Number of writing blocks

- Number of times that reading was handled as ordinary I/O
- Number of times that writing was handled as ordinary I/O
- Number of times where page flush occurred after reading
- Number of times where page flush occurred after writing

-f

This option reports the following information on the manipulation of each of the free extents:

- Number of search requests
- Acquisition count
- Number of acquired blocks
- Number of acquisition failures
- Release count
- Number of released blocks
- Number of times that released blocks were concatenated
- Number of times of fragmentation by free extent management
- Number of times of connection by free extent management

-i

This option reports the following information:

- Actual read I/O count
- Actual write I/O count
- Actual read I/O block count
- Actual write I/O block count

-l

This option reports the following information on the log daemon:

- Number of times the log daemon was activated
- sync request count
- fsync request count
- Number of times when the log buffer was full
- Number of requests from the meta-write daemon
- Number of times that sleep occurred because of an excessive number of transactions
- Number of times that sleep occurred because the log disk was full
- Number of times that sleep occurred because the preceding transaction was too large
- Number of times that sleep occurred because of expansions of the preceding transaction
- Number of times that I/O instructions were issued from the log daemon
- Time required for writing
- Number of written blocks
- Number of valid blocks
- Number of log target processing requests
- Number of parallel operations for log target processing
- Average number of parallel operations
- Number of using extremely large buffer
- Number of generated sub-transactions

-m

This option reports the following information on the meta-daemon:

- Number of times the meta-daemon was activated
- Number of buf that were waiting for I/O by the meta-daemon
- Number of times the meta-daemon was activated because the disk was full
- Number of times the meta-daemon was activated because the meta-cache was full

-A

This option reports all available information.

B.2 Management Commands Specific to the GFS Local File System

EXAMPLES

```
# sfxstat 5 /mnt
Filesystem:/mnt Tue Jan 14 17:47:23 JST 2003
metavolume[0]
metatype access cachehit      mod    R-I/O    W-I/O    W-MS    I/Oave
DBMAP      972      972      972      0        22       730     33
IBMAP      978      978      978      0        21       570     27
NBMAP      0         0         0         0         0         0         0
LBMAP      0         0         0         0         0         0         0
SBMAP      0         0         0         0         0         0         0
IEX        0         0         0         0         0         0         0
DIR        4343     3361     971      982      32       1500    46
SYM        0         0         0         0         0         0         0
NODE       0         0         0         0         0         0         0
LEAF       0         0         0         0         0         0         0
SLIST      0         0         0         0         0         0         0
DINODE     2935     2813     2935     122      181      10360   57
SBLOCK     0         0         0         0         4         90      22
ACL        0         0         0         0         0         0         0
QUOTA      0         0         0         0         0         0         0

          readcnts  readblks  writecnts  writeblks
data volume[00]:      0         0         0         0
data volume[01]:      0         0         0         0
data volume[02]:      0         0         0         0
data volume[03]:      0         0         0         0
lwakeup  lsync  lfsync  lbfull  lmeta  ltfull  ldfull  logbig  logbigx
    68      1      0      26      0      0      0      0      42
Inwrite   lms  lnblks  loccupy  Intrans  lpara  lave  lbigbuf  lsubtrn
    69  1530  6750  88700   1962    21    0    0      0
fsearch  falloc  fnalloc  fealloc  ffree  fnfree  fcomb  fsplit  fcomb
    0      0      0      0      0      0      0      0      0
asearch  aalloc  analloc  aealloc  afree  anfree  acomb  atsplit  atcomb
  4343    0      0      0    978   978    0      0      0
mwakeup  mnbuf  mlogful
    55   298      6
METAFULL DBMAP  IBMAP  NBMAP  LBMAP  SBMAP  IEX  DIR
          0      0      0      0      0      0      0
          SYM  NODE  LEAF  SLIST  DINODE  SBLOCK  ACL
          0      0      0      0      0      0      0
          QUOTA
          0
          dirr  dirw  dirrblk  dirwblk  dirr-n  dirw-n  dirrflsh  dirwflsh
          0      0      0      0      0      0      0      0
          bufalloc  dbufalloc  dbufsleap
          2446      0      0
```

SEE ALSO

sar(1)

B.2.17 sfxxtune(1M) - Tunes up the GFS Local File System

SYNOPSIS

sfxxtune [*specific_options*] *special* | *filesystem*

DESCRIPTION

The **sfxxtune** command changes the dynamic parameters of GFS Local File System, which affect the layout policies.

Before using **sfxxtune**, unmount the file system.

When using **sfxxtune**, specify the file system in */etc/vfstab*.

Specify the parameters that are to be changed with the options below.

When a file system is configured with two or more partitions, specify the special file name under the path name for the representative partition.

OPTIONS

specific_options

The following options can be specified for **sfxxtune**.

-m minfree

This option specifies the percentage of area prevented from accessing of normal users.

-o[space|time]

This option specifies a file system optimization strategy.

space: Space

time: Access time

-a maxcontig

This option specifies the maximum number of contiguous blocks that can be allocated before waiting for forcible rotation.

-M mminfree

This option specifies the percentage of V data area prevented from accessing of normal users.

EXAMPLES

```
# sfxxtune -m 5 /dev/sfdsk/gfs/dsk/vol0301
```

SEE ALSO

mkfs(1M), fork(2), terminfo(4), attributes(5), largefile(5)

B.3 General Commands Specific to the GFS Local File System

B.3.1 sfxcp(1) - Copies files with extent attribute information

SYNOPSIS

```
sfxcp [-fip] [ -e ext_action ] source_file target_file  
sfxcp [-fip] [ -e ext_action ] source_file... target  
sfxcp -r|-R [-fip] [ -e ext_action ] source_dir... target
```

DESCRIPTION

The **sfxcp** command copies both files with extent attribute information and files without extent attribute information on the GFS Local File System. If necessary, the **sfxcp** command will operate on a file system other than the GFS Local File System, but in this case **cp**(1) should be used as much as possible.

In format 1 shown in SYNOPSIS, neither *source_file* nor *target_file* must be a directory file, and they must not be specified with the same name. The **sfxcp** utility copies the contents of the file indicated by *source_file* and the extent attribute information of the GFS Local File System to the destination path indicated by *target_file*. If *target_file* already exists, **sfxcp** overwrites the contents of the file. The mode (ACL if it is valid), owner, and group of the file, however, remain unchanged. For information about extent attribute information, see the explanation of the **-e** option. The latest update time in *target_file* and the latest access time in *source_file* are set as the copy time. If *target_file* file is not found, **sfxcp** creates a file named *target_file*. The mode of this file is the same as that of *source_file*. However, a sticky bit is not specified unless the user that issued this command is a super user. For the owner and group of *target_file*, the owner name and the group name of the owner are used. If *target_file* indicates a link to another file having a link, the other link remains and *target_file* assumed to be a new file.

In format 2, **sfxcp** copies one or more files indicated by *source_file* to a directory indicated by *target*. For each *source_file*, a new file is created in *target* in the same mode (the same ACL ACL if it is valid). In this case, this system uses the owner and group of the user that issues this command. When the type specified in *source_file* is directory, *target* is not found, and *target* is not directory, processing fails.

In format 3, **sfxcp** copies one or more directories indicated by *source_dir* to the directory indicated by *target*. Either **-r** or **-R** must be specified. **sfxcp** copies all files and sub-directories in each directory indicated by *source_dir*.

OPTIONS

-e *ext_option*

Specifies an operation for extent attribute information. Explanation of *ext_option*

AWE

Append Write Extent (default). When *target* and *target_file* already exist with extent attribute information, this option does not copy the extent attribute information. Otherwise, this option copies extent attribute information.

IWE

Interactive Write Extent. When *target* and *target_file* already exist with extent attribute information, this option displays a confirmation message asking whether the extent attribute information should be overwritten. If the response is **y**, the extent attribute information is copied. If the response is another character, the information is not overwritten.

NWE

Not Write Extent. This option does not copy extent attribute information in the same way as **cp**(1).

OWE

OverWrite Extent. This option forcibly copies the extent attribute information of source to *target* and *target_file*. Even though the source does not contain extent attribute information, it is forcibly copied to *target* and *target_file*.

-f

Releases a link. If a file descriptor is not obtained for the destination file, this option releases the link of the destination file and attempts to continue processing.

-i

Indicates interactive mode. When *target* and *target_file* specified, as destination files already exist, this option displays a confirmation message asking whether the data should be overwritten. If the response is y, copy processing is continued. If the response is another character, the data is not overwritten.

When both the **-i** option and **-e IWE** option are specified together, they are processed independently in interactive mode.

-r

Recursive. **sfxcp** copies to *target* the entire directory (that is, all files in the directory), and also copies all subdirectories in the directory and all files in the subdirectories.

-R

The same as **-r**, except that a named pipe file is copied, not read.

-p

Holds data. **sfxcp** copies the contents of the file indicated by *source_file*, and also specifies an owner and group ID, access right mode, update time, access time of the output file. If ACL is valid, the input file value is set as the ACL value. This command does not fail when it is not possible to specify an update time, access time, or access right mode. When the owner and group ID cannot be specified, **sfxcp** does not fail, and clears S_ISUID and S_ISGID of the output file. If **sfxcp** outputs a diagnostic message to the standard error output and returns a termination status other than 0.

To store the owner, group ID, access right mode, update time, and access time, the user must have access permission to the appropriate file. In other words, the user must have the same owner ID as that of the target file.

OPERANDS

The following operands can be specified:

source_file

Path name of an ordinary file to be copied

source_dir

Path name of a directory to be copied

target_file

Path name of the file used as the output when one file is copied regardless of whether the file already exists.

target

Path name of a directory to which a copied file is to be written

EXAMPLES

Examples in which one file is copied

```
% sfxcp goodies goodies.old
% ls goodies*
goodies goodies.old
```

Examples in which one file group is copied to a directory

```
% sfxcp ~/src/* /dst
```

Examples in which a specific directory is copied first to a new directory, and then to an existing directory

```
% ls ~/bkup
```

B.3 General Commands Specific to the GFS Local File System

```
/usr/example/fred/bkup not found
% sfxcp -r ~/src ~/bkup
% ls -R ~/bkup
x.c y.c z.sh
% sfxcp -r ~/src ~/bkup
% ls -R ~/bkup
src x.c y.c z.sh

src:
x.c y.c z.sh
```

Example in which data is forcibly copied on the basis of source information when target has extent attribute information

```
% sfxcp -e OWE source target
```

ENVIRONMENT

For information about environment variables **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**, which affect the execution of **sfxcp**, see **environ(5)**.

EXIT STATUS

The following termination status information is returned:

- 0 All files have been copied successfully.
- >0 An error has occurred.

SEE ALSO

[sfxcpio\(1\)](#), [sfxmv\(1\)](#), [sfxsetext\(1\)](#), [sfxgettext\(1\)](#)
[cp\(1\)](#), [chmod\(1\)](#), [chown\(1\)](#), [setfacl\(1\)](#), [utime\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#)

NOTES

The access right mode of *source_file* is stored when copy is executed.

Users can clearly specify the end of a command line option with two consecutive hyphens (--). When this method is used and a file name beginning with a single hyphen (-) is specified as the filename argument, **sfxcp** recognizes that the character string indicates a file name. If both -- and - are written on the same line, **sfxcp** interprets the second hyphen as specifying a file name.

sfxcp can copy data between different file systems. If a file on the GFS Local File System is copied to another file system (for example, UFS), however, extent attribute information is deleted.

sfxcp will operate on a file system other than the GFS Local File System, but its operation will be equivalent to the conventional cp.

B.3.2 sfxcpio(1) - Extracts or restores files from an archive with extent attribute information

SYNOPSIS

```
sfxcpio -i [ bBcdfkmPrsStuvV6 ] [ -C bufsize ] [ -E file ] [ -I file [ -M message ] ] [ -R id ] [ pattern ... ]
sfxcpio -o [ aABcLPvV ] [ -C bufsize ] [ -O file [ -M message ] ]
sfxcpio -p [ adLLmPuvV ] [ -R id ] directory
```

DESCRIPTION

The **sfxcpio** command copies both files containing extent attribute information and files not containing extent attribute information from the GFS Local File System to the sfxcpio archive and vice versa.

The **sfxcpio** command also operates on a system other than the GFS Local File System. In this case, using **cpio**(1) is strongly recommended. For information about the relationships between the **sfxcpio** and **cpio** commands or the operations to be performed when the sfxcpio archive is restored to a system other than the GFS Local File System, see NOTES below.

The **-i**, **-o**, and **-p** options are used to select the processing to be executed. The type of processing for each option is explained in detail. (These types of processing are mutually exclusive.)

[Restore mode]

sfxcpio -i (copy in) extracts a file using the output previously created by **sfxcpio -o** as the standard input. Only files having the name specified in pattern are selected. For information about pattern, see **OPERANDS** and **sh**(1) below. The extracted file is created as defined by the conditions, and copied to the current directory on the basis of the options described below. The file access right is the same as that of the previous output of **sfxcpio -o**.

The owner and group are the same as the owner and group of the current user unless the current user is super-user. If the current user is a super-user, the owner and group are the same as the owner and group obtained from the output file previously processed with **sfxcpio -o**. If a file to be created by **sfxcpio -i** already exists and the version of the file is the same or a newer version, a message is displayed and the file is not replaced. (The **-u** option is used to replace an existing file regardless of the conditions.)

[Storage mode]

sfxcpio -o (copy out) reads a path name list from the standard input and copies their files to the standard output together with path names, status information, and extent attribute information. When data is output, the output data is blocked on 512-byte boundaries by default. If the **-B** or **-C** option is used, output data can be blocked using the user-specified block size or with the block size, depending on a device.

[Pass Mode]

sfxcpio -p (pass) reads a path name list for file from the standard input, and creates a file according to the conditions, and copies it to a destination directory on the basis of the options below.

When data is output to a character-type device (**-o**) or input from a character-type device (**-i**), if the end of the medium (for example, end of a tape) is reached without the **-O** and **-I** option, **sfxcpio** displays the following message:

To continue, type device/file name when ready.

To continue the processing, replace the medium, enter a character-type special device name (for example, `/dev/rmt/0m`), and press **RETURN** key. This function is used to instruct the **sfxcpio** command to use different devices and to continue the processing. For example, when there are two tape units, this function is convenient for switching the current device to another device so that the **sfxcpio** command can continue processing during replacement of a tape. (The **sfxcpio** command terminates processing when the **RETURN** key is pressed.)

OPTIONS

This system supports the following options:

-i

(copy in) **sfxcpio -i** extracts a specified file from the standard input.

B.3 General Commands Specific to the GFS Local File System

-o (copy out) **sfxcpio -o** reads a path name list from the standard input and copies the file to the standard output.

-p (pass) **sfxcpio -p** reads a path name list for a specified file from the standard input.

After the **-o**, **-i**, or **-p** option is specified, the following options can be specified in an any order.

-a After copying a file, this option resets the input file access time. If **sfxcpio -pla** (cannot be specified together with **-m**) is specified, the access time of a linked file is not reset.

-A Adds a file to an archive. This option must be paired with the **-O** option. This option is valid only for files, floppy disks, or hard disk partition archives.

-b Reverses the byte order of each word. (can only be used together with the **-i** option.)

-B Block data as 5120 bytes for data input/output. When the **-B** and the **-C** options are not used, the default buffer size is 512 bytes. The **-B** option is not used for the **-p** option. This is because the **-B** option is valid for input from or output to a `/dev/rmt/0m` a character-type special device.

-c Reads and writes header information in ASCII mode, taking into account portability. There are no restrictions on user ID or group ID for this header format. The **-c** option means using extension device numbers supported only in the SVR4-based systems.

-C bufsize Block input/output is in *bufsize* units. *bufsize* is replaced with a positive integer. When the **-B** and the **-C** options are not used, the default buffer size is 512 bytes. (The **-C** option is not used for the **-p** option; This is because the **-C** option is valid only for input to or output from a character-type special device such as `/dev/rmt/0m`.)

-d Create a directory as required.

-E file Specifies an (one filename per line) input file (file), including a file name list to be extracted from archive.

-f Extracts all files excluding those specified in pattern. (For information about pattern, See **OPERANDS** below.)

-l file Read the contents as an input archive. If the file is a character-type special device and it finishes reading all of the current medium, replace the medium and press the **RETURN** key to continue processing. This option can be used only with the **-i** option.

-k Skips a damaged file header or input-output error. To copy a file from a damaged medium or media that are out of sequence, this option allows only files having a normal header to be read. (When an **sfxcpio** archive contains another **sfxcpio** archive and an error occurs, the **sfxcpio** command terminates. The **sfxcpio** command searches for the next normal header. When a smaller archive is found, this command reads data until it encounters its trailer and then terminates processing.) The **-k** option can be used only with the **-i** option.

-l As much as possible, links files instead of copying them. (This option can be used only with the **-p** option.)

-L Traces a symbolic link. The default does not trace a symbolic link.

-m Fetches the previous file update time. This option is invalid if a directory is being copied. (This option cannot be used with the **-a** option.)

-M message

B.3.2 sfxcpio(1) - Extracts or restores files from an archive with extent attribute information

Defines a message for replacement of a medium. When a character-type special device is specified with the **-O** or **-I** option, this option defines a message that is displayed at the end of a medium. One %d can be used to display the sequence number of the next medium.

-O file

Changes an **sfxcpio** output destination to file. When file is a character-type special device and the current medium is full, replace the medium and press the **RETURN** key to continue processing. This option can be used only with the **-o** option.

-P

Holds ACL. When this option is used for output and ACL exists, ACL is written to the standard output together with other attributes. ACL is created as a special file having a special file type. When this option is used for input, ACL is restored from the standard input together with other attributes. This option can identify a special file type. This option cannot be used with the **-c** option because ACL is not necessarily supported on all systems. Also, portability is not assured. To take into account portability, use an ASCII header.

-r

Changes a file name in interactive mode. When only the **RETURN** key is pressed, the file is skipped. If "." is entered, the original path name is used. (This option is invalid for **sfxcpio -p**.)

-R id

Changes the owner and group information for each file to a user ID (*id* must be a valid login ID in the /etc/passwd file). Only a super user can use this option.

-s

Substitutes bytes for each half word.

-S

Substitutes a half word for each word.

-t

Displays the entered character strings. No file is then created. (This option cannot be used with **-V** option.)

-u

Executes copying regardless of the condition. (An old file is not ordinarily replaced with a new one.)

-v

Displays detail information and lists file names. When the **-v** option is used together with the **-t** option, data is output the same way as for the **ls -l** command (see **ls(1)**).

-V

Displays detail information: that is, completely displays the contents for each input or output file. This option is convenient when the user has confirmed that **sfxcpio** operates without displaying all file names.

OPERANDS

The following operands can be specified.

directory

Path name of an existing directory to be processed by **sfxcpio -p**.

pattern

Formula using a representation method for pattern matching. This formula is similar to the one (see **sh(1)**) the shell uses for file name pattern matching or regular representation. The following meta-characters are defined:

*

Matches all character strings, including an empty character string.

?

Matches any 1 character.

[...]

Matches one of the characters enclosed with braces. When two characters are delimited by a hyphen '-', they match all characters (including those two characters) between two characters according to the default collation sequence of the system. If the character following the first bracket '[' is a '!', the results is not assured.

!

B.3 General Commands Specific to the GFS Local File System

Indicates a negation. For example, the `!abc*` is specified, the pattern matches a file name that does not begin with character string `abc`.

In *patterns*, meta-characters `?`, `*`, and `[...]` match a slash (`/`) and a backslash (`\`) is an escape character. More than one pattern can be specified. If no character is specified in *pattern*, the default is `*` (that is, all files are selected).

Each pattern must be enclosed in double quotation marks (`"`). If it is not, a file in the current directory may be used.

USAGE

A file of two gigabytes or more cannot be stored. Also, an archive of eight gigabytes or more cannot be created.

EXAMPLES

The following are three **sfxcpio** examples:

When the standard input is transferred to **sfxcpio -o**, via a pipe, files are grouped and collected as one archive file (`../newfile`). The **-c** option ensures compatibility with a different machine from the machine with the archive file. To transfer a file name list to **sfxcpio -o**, via a pipe, **find(1)**, **echo(1)**, and **cat(1)**, can be used instead of **ls(1)**. If necessary, the output destination can be changed from a file to a device.

```
% ls | sfxcpio -oc > ../newfile
```

sfxcpio -i extracts a file that matches a specified pattern (`memo/a1`, `memo/b*`), using the output of **sfxcpio -o** (in the example below, transferred to a pipe using `cat`). This command also creates a directory under the current directory (**-d** option) as required, and stores the file in the appropriate directory. The **-c** option is available when an input file is created with a compatible header. If no pattern is specified, all files in `newfile` are stored in the directory.

```
% cat newfile | sfxcpio -icd "memo/a1" "memo/b*"
```

sfxcpio -p reads file names via a pipe and copies or links (**-l** option) those files to another directory (in the example below, `newdir`). The **-d** option means that a directory is created as required. The **-m** option fetches a modification time. To create a path name required to transfer data to **sfxcpio**, use the **-depth** option of **find(1)** command. With this function, the user can eliminate a problem that occurs when an attempt is made to create files under a read-only directory. The copy destination directory `newdir` is necessary.

```
% find . -depth -print | sfxcpio -pdlmv newdir
```

When **sfxcpio** is used together with `find` and **-L** option is specified in the **sfxcpio** option, use **-follow** in the `find` option and vice versa. If **-follow** is not used, an unexpected result is displayed.

To perform archive processing for multiple tapes, remove the old tape, mount a new one, and enter the next unit name (ordinarily, the same name as the first tape unit) to continue with the next tape. If the **RETURN** key is pressed, **sfxcpio** terminates.

ENVIRONMENT

For information about environment variables **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, **LC_TIME**, **TZ**, and **NLSPATH**, see **environ(5)**.

TERMINATION STATUS

The following termination status information is returned:

0	Terminated normally.
>0	An error occurred.

SEE ALSO

[sfxcp\(1\)](#), [sfxmv\(1\)](#), [sfxgettext\(1\)](#), [sfxsettext\(1\)](#)
cpio(1), ar(1), cat(1), echo(1), find(1), ls(1), setfacl(1), sh(1), tar(1), vold(1M), archives(4), attributes(5), environ(5), largefile(5), standards(5)

NOTES

The path name is limited to 256 bytes (default) in binary notation; otherwise, limited to 1,024 bytes.

If a file has too large a user ID or group ID for a selected header format, an error message is displayed. To create an archive that can support all the user ID or group ID values, use the `-c` option.

Only a super user can copy a special file.

One block is 512 bytes.

When the access right of a specific file is 000, the file contains character data; and the user is not root, the file is not stored.

While the volume management daemon is operating, in some cases a floppy disk unit cannot be accessed using a normal device name such as `/dev/rdiskette`. For details, see **vold(1M)**.

To fetch and copy a file from a tape to hard disk, use the same blocking factor as that used to copy the file from hard disk to the tape. Therefore, specify the **-B** option.

sfxcpio can read only an archive in the **sfxcpio** format. An archive in the **cpio** format cannot be read.

sfxcpio will operate on a file system other than the GFS Local File System, but its operation will be equivalent to the conventional **cpio**.

sfxcpio can operate between different file systems. If a file on the GFS Local File System is copied to another file system (for example, UFS), extent attribute information is erased.

B.3.3 sfxgettext(1) - Displays extent attribute information

SYNOPSIS

sfxgettext [-iH] *file1 file2 ...*

DESCRIPTION

The **sfxgettext** command displays extent attribute information about a specified file on the GFS Local File System.

The details of the extent attribute information are listed below. Note, however, that this information is not displayed when the specified file is not on the GFS Local File System or its file attribute is not ordinary file.

- File system block size
- Initial number of blocks
- Incremental number of blocks
- Allocation flag
- File name

For information about the initial number of blocks, increment number of blocks, and allocation flag, see [sfxsettext\(1\)](#).

OPTIONS

-i

This option disables display when no extent attribute information is defined for a specified file and lists only files that have extent attribute information.

-H

Displays normal data after a column header.

EXAMPLES

```
% sfxgettext /mnt/file1 /mnt/file2 /mnt/file3 /mnt/file4 /mnt/file5
      8          1024      512    contig      /mnt/file1
      8          2048       0    noextend    /mnt/file2
      8           512       0    contig,noextend /mnt/file3
      8           512       0      -          /mnt/file4
      8           -        -        -          /mnt/file5
% sfxgettext -i /mnt/file1 /mnt/file2 /mnt/file3 /mnt/file4 /mnt/file5
      8          1024      512    contig      /mnt/file1
      8          2048       0    noextend    /mnt/file2
      8           512       0    contig,noextend /mnt/file3
      8           512       0      -          /mnt/file4
% sfxgettext -iH /mnt/file1 /mnt/file2 /mnt/file3 /mnt/file4 /mnt/file5
Blocksize(KB)  init-blocks  inc-blocks   flags      filename
      8          1024      512    contig      /mnt/file1
      8          2048       0    noextend    /mnt/file2
      8           512       0    contig,noextend /mnt/file3
      8           512       0      -          /mnt/file4
```

SEE ALSO

[sfxsettext\(1\)](#), [sfxcp\(1\)](#), [sfxcpio\(1\)](#), [sfxmv\(1\)](#)
[statvfs\(2\)](#)

B.3.4 sfxmv(1) - Moves files with extent attribute information

SYNOPSIS

```
sfxmv [-fi] source target_file
sfxmv [-fi] source... target_dir
```

DESCRIPTION

The **sfxmv** command moves a file with or without extent attribute information on the GFS Local File System. The **sfxmv** command also operates on the other file system, but in this case, **mv(1)** should be used as much as possible.

In format 1 above, the **sfxmv** utility moves the file indicated by the *source* operand to the destination indicated by the *target_file*. The same name cannot be specified for *source* and *target_file*. If *target_file* is not found, **sfxmv** creates a new file with the same name. If *target_file* is found, its contents are overwritten. When the last operand does not indicate an existing directory, **sfxmv** is assumed to be in format 1.

In format 2, the **sfxmv** utility moves each file specified in the *source* operand to each file in an existing directory indicated in the *target_dir* operand. In the destination path name for a *source*, a slash (/) and the last path name of *source* are added after the destination directory name. When the last operand indicates an existing directory, **sfxmv** is assumed to be in format 2.

When *target_file* is identified in write-disabled mode, **sfxmv** displays a mode (see **chmod(2)**), requests a response, and reads one line from the standard input. If the response is positive and **sfxmv** has the appropriate permission, it moves the file. Otherwise, this command terminates. When *target_file* makes ACL valid, the access right is not completely displayed in mode. If the parent directory of *source* is in write-enabled mode with a sticky bit set, one or more of the following conditions must be true:

- o The user must own the file.
- o The user must own directories.
- o The file must be in user-write-enabled mode.
- o The user must be a privileged user.

When *source* indicates a file and *target_file* indicates a link to another file having a link, the other link remains, and *target_file* changes to a new file.

OPTIONS

-f

sfxmv moves a file without displaying a prompt even when it writes data to an existing target. Note that this option is the default if the standard input is not a terminal.

-i

sfxmv always displays a confirmation prompt when an existing target is overwritten in a move operation. If the response is **y**, the move processing continues. If the response is another character, **sfxmv** does not overwrite target.

When both the **-f** and **-i** options are specified, they are not assumed to be illegal. Instead, the **-f** option is valid.

OPERANDS

The following operands can be specified:

source

Path name of a file or directory to be moved

target_file

A new path name of a file or directory to be moved

target_dir

Path name of an existing directory used as the destination of the move file operation

B.3 General Commands Specific to the GFS Local File System

ENVIRONMENT

For information about environment variables **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**, which affect the execution of **sfxmv** see **environ(5)**.

EXIT STATUS

The following termination status information is returned:

0	All input files have been moved correctly.
>0	An error has occurred.

SEE ALSO

[sfxcp\(1\)](#), [sfxcpio\(1\)](#), [sfxgettext\(1\)](#), [sfxsettext\(1\)](#)

[mv\(1\)](#), [cp\(1\)](#), [cpio\(1\)](#), [ln\(1\)](#), [rm\(1\)](#), [setfacl\(1\)](#), [chmod\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#)

NOTES

When *source* and *target_dir* are on different file systems, **sfxmv** copies the file and deletes the originals; In this case, all links to other files are deleted. Since users can clearly specify the end of a command line option with two consecutive hyphens (--), **sfxmv** can identify a file name argument that begins with a single hyphen (-). For a move operation in BSD, **sfxmv** accepts a single hyphen (-) as a synonym of two hyphens (--). Note, however, that the BSD move function may be deleted in future releases. If both two consecutive hyphens (--) and a single hyphen (-) are specified on the same line, the latter is interpreted as specifying a file name.

When target is specified, use **sfxcp(1)** to store extent attribute information for target (with **-e NWE** specified).

sfxmv can move a file between different file systems. If a file on the GFS Local File System is moved to another file system (for example UFS), however, extent attribute information is deleted.

sfxmv will operate on a file system other than the GFS Local File System, but its operation will be equivalent to the conventional **mv**.

B.3.5 sfxsetext(1) - Specifies extent attribute information

SYNOPSIS

```
sfxsetext -a init-blocks[,inc-blocks] [ -f flags ] file
```

DESCRIPTION

The **sfxsetext** command specifies extent attribute information for a specified file on the GFS Local File System.

After creating the specified file, this command specifies extent attribute information according to the specified options. In this case, the initial area allocated by the number of blocks specified in *init-blocks* is allocated.

However, extent attribute information cannot be specified when the specified file already exists or does not belong to the GFS Local File System.

OPTIONS

-a *init-blocks*[,*inc-blocks*]

init-blocks specifies the number of blocks allocated to the file as the initial area. Zero (0) cannot be specified.

inc-blocks specifies the increment (in blocks) when write data exceeds the initial area. If 0 is specified for *inc-blocks* or *inc-blocks* is not specified, the normal allocation is used.

When **noextend** is specified using the **-f** option, either specify 0 for *inc-blocks* or do not specify the option. *init-blocks* and *inc-blocks* must be specified in the unit of blocks (eight kilobytes) used for the file system.

-f *flags*

This option specifies the data area allocation method. The combination of the flags below can be specified for *flags*. To specify more than one flag, use a comma (,) as the delimiter.

contig

Allocates a contiguous area for the number of blocks specified in *init-blocks* and *inc-blocks*. Depending on the number of blocks specified in *inc-blocks*, the *init-blocks* area is allocated as shown below.

If 0 is specified:

Allocates a contiguous area for the number of blocks specified in *init-blocks*.

If less than *init-blocks*:

Searches for a contiguous space that is equivalent to the number of blocks specified in *init-blocks*. If it cannot be found, the system allocates a contiguous area for *init-blocks* using the blocks specified in *inc-blocks*. If the specified value in *inc-blocks* is not a multiple of *init-blocks*, the system allocates a contiguous area for the number of blocks specified in *inc-blocks*, and then allocates a contiguous area for the remaining (including fractional) blocks.

If greater than *init-blocks*:

Allocates a contiguous area for the number of blocks specified in *init-blocks*.

noextend

If the write data exceeds the number of blocks specified in *init-blocks* or the file size is changed, additional area is not allocated, and error information is returned.

If the **-f** option or **contig** is not specified, the system does not necessarily allocate a contiguous area for the number of blocks specified in *init-blocks* or *inc-blocks*.

EXAMPLES

```
% sfxsetext -a 1024,512 -f contig /mnt/file1
% sfxsetext -a 2048 -f noextend /mnt/file2
% sfxsetext -a 512 -f contig,noextend /mnt/file3
% sfxsetext -a 512 /mnt/file4
```

SEE ALSO

[sfxadm\(1M\)](#), [sfxcp\(1\)](#), [sfxcpio\(1\)](#), [sfxgettext\(1\)](#), [sfxmv\(1\)](#)
[statvfs\(2\)](#)

NOTES

In some cases, even though the file system has sufficient space, area allocation fails because of the specified option values.

For *init-blocks* and *inc-blocks*, think about the value to be specified as the required number of blocks. If the number of blocks is unnecessarily large, used area increases, causing disk resources to be used inefficiently.

B.4 Miscellaneous

B.4.1 fs_sfxfs(4) – The partition management format of the GFS Local File System

SYNOPSIS

```
#include <sys/param.h>
#include <sys/types.h>
#include <sys/fs/sfxfs_fs.h>
#include <sys/fs/sfxfs_inode.h>
```

DESCRIPTION

The GFS Local File System resides after the first 16 sectors of each partition.

The superblock for each GFS Local File System is placed 16 sectors after the beginning of the partition and at the end of the partition.

The superblock contains the basic information on the structure of the file system.

Placing the original superblock at the first part of the partition and a copy at the end of the partition helps to avoid a fatal error that would be caused by damage to the superblock.

The superblock information must be consistent among the partitions that constitute the same file system.

The following information is stored in the superblock:

fs_clean

This parameter contains information on the file system status. FSCLEAN indicates a normally unmounted file system. FSACTIVE indicates that the mounted file system is currently being updated. FSSTABLE indicates that the mounted file system is idle. FSFIX indicates that the file system is being repaired with **fsck_sfxfs(1M)**.

FSBAD indicates that the file system contains an inconsistency.

File systems in FSCLEAN or FSSTABLE status do not have to be repaired with fsck. If an attempt is made to mount a file system that is FSACTIVE **mount(2)** in read-write enabled mode, ENOSPC is returned.

fs_minfree

fs_mminfree

The **fs_minfree** parameter indicates the allowed minimum ratio of free space in the data area that is available for requests from common users. The **fs_mminfree** parameter indicates the allowed minimum ratio of free space in the V-data area that is available for requests from common users. If the free space of the is lower than the ratio defined by these limits, only the super user can allocate additional area.

By default, a value of 10% is specified.

fs_time

This parameter indicates the last time the superblock was written to disk.

fs_dsize

This parameter indicates the number of data blocks in each file system.

fs_ysize

This parameter indicates the number of i-nodes in the file system.

fs_msize

This parameter indicates the number of V-data blocks in the file system.

fs_fdsize

This parameter indicates the number of free data blocks in the file system.

fs_fysize

This parameter indicates the number of free i-nodes in the file system.

fs_fmsize

This parameter indicates the number of free V-data blocks in the file system.

B.4 Miscellaneous

fs_optim

This parameter specifies which operation should have highest priority during area acquisition in the file system in order to reduce allocation time or fragmentation.

FS_OPTTIME places highest priority on the reduction of allocation time.

FS_OPTSPACE places highest priority on the suppression of fragmentation.

fs_fsmnt

This parameter is for recording the path name used at mounting.

SEE ALSO

[fck_sfxfs\(1M\)](#), [mkfs_sfxfs\(1M\)](#), [sfx tune\(1M\)](#)

[mount\(2\)](#)

Appendix C Corrective Measures against I/O Errors in the GFS Local File System

This appendix explains corrective measures that are to be taken if an I/O error occurred in the GFS Local File System.

C.1 If the message "sfxfs...: read error" or "sfxfs...:write error" is output

The message "sfxfs...: read error" indicates that an I/O error occurred while file system data was read from the disk. The Message "sfxfs...:write error" indicates that an I/O error occurred while file system data was written to the disk.

If one of these error messages is output, an I/O path or device is faulty and needs to be recovered.

Perform a recovery as follows:

1) Determine the file system and partition where the error occurred.

To do so, enter the GFS Local File System management command `sfxinfo(1M)` using the device number in the error message as key.

```
# sfxinfo -d device-number
```

2) If "sfxfs...:write error" is output

If this error occurs, file system information that was updated in memory has not been saved to the disk. In this case, it is recommended to back up as many files as possible in the file system before the file system is unmounted: After the file system is unmounted, updated information that has not been registered properly is lost.

3) Unmount the file system where an error occurred.

Use the `umount` command to unmount the faulty file system. For this command, specify the mount point that was determined in step 1).

4) Make a copy of the file system data in the partition where the error occurred, replace the partition, then use `fsck (1M)` to restore the consistency of the file system. Issue `fsck(1M)` while specifying to perform a full check.

```
# fsck -F sfxfs -o nolog special-device-name
```

5) If the file system cannot be recovered by step 4), replace the faulty location, recreate the file system, and restore the data from the last backup.

Appendix D Command correspondence with GFS Local File System and UFS file system

This appendix explains the command correspondence with GFS Local File System and UFS file system (the Solaris OE standard).

D.1 Correspondence with GFS Local File System commands and UFS commands

Correspondence with GFS Local File System commands and UFS commands is shown in the following with a list.

The Solaris OE standard command that is not mentioned in the following lists can be used on a GFS Local File System.

A correspondence list with GFS Local File System commands and UFS commands (1/2)

GFS Local File System Command	UFS Command	Comment
clri(1M)	clri(1M)	Clrea i-node. Must be appointing a file system type.
sfxcp(1)*	cp(1)	Copy a file. (*: Accompanied with extent attribute information) When not need to maintain extent attribute information, can substitute it by a Solaris OE standard command.
sfxcpio(1)*	cpio(1)	Extract or restore a file from an archive. (*: Accompanied with extent attribute information) When not need to maintain extent attribute information, can substitute it by a Solaris OE standard command.
df_sfxfs(1M)	df_ufs(1M)	Report the free disk area of the file system. Must be appointing a file system type.
sfxedquota(1M)	edquota(1M)	Edit user quotas for file system.
ff_sfxfs(1M)	ff_ufs(1M)	List file names and statistical information for the file system. Must be appointing a file system type.
fsck_sfxfs(1M)	fsck_ufs(1M)	Check and repair for file systems. Must be appointing a file system type.
fstyp(1M)	fstyp(1M)	Determine the type for each file system.
-	fsdb(1M)	File system debugger.
-	labelit(1M)	Provide and print labels for file systems.
mkfs_sfxfs(1M)	mkfs_ufs(1M)	Construct a file system. Must be appointing a file system type.
mount_sfxfs(1M)	mount_ufs(1M)	Mount the file system.
sfxmv(1)*	mv(1)	Move files. (*: Accompanied with extent attribute information) When not need to maintain extent attribute information, can substitute it by a Solaris OE standard command.
ncheck_sfxfs(1M)	ncheck_ufs(1M)	Create a file system file name and i-node number. Must be appointing a file system type.
sfxnewfs(1M)	newfs(1M)	Construct a new file system.
-	quot(1M)	Summarize file system ownership.

D.1 Correspondence with GFS Local File System commands and UFS commands

sfxquota(1M)	quota(1M)	Display a user's file system disk quota and usage. Cannot display the information of GFS Local File System done NFS mount.
-	quotacheck(1M))	File system quota consistency checker. In a GFS Local File System, adjustment is unnecessary automatically in order to handle it.
sfxquotaon(1M)	quotaon(1M)	Turn file system quotas on.
sfxquotaoff(1M)	quotaoff(1M)	Turn file system quotas off.
sfxrepquota(1M)	repquota(1M)	Summarize quotas for a file system.

-: no correspondence command.

**A correspondence list with GFS Local File System commands and UFS commands
(2/2)**

GFS Local File System Command	UFS Command	Comment
sfxadd(1M)	-	Extend the file system.
sfxadm(1M)	-	Change file system partition information settings.
sfxddr(1M)	-	Restore backup data and check restore device.
sfxdefrag(1M)	-	Defragment the file system.
sfxgetext(1)	-	Display extent attribute information.
sfxinfo(1M)	-	Display file system partition information.
sfxmkbkup(1M)	-	Create shell script for file system backup file.
sfxmkrstr(1M)	-	Create shell script for file system restore file.
sfxnode(1M)	-	Add or change file system node configuration information.
sfxquotadm(1M)	-	Manage quotas file for file system.
sfxsetext(1)	-	Specify extent attribute information.
sfxstat(1M)	-	Report statistical information on the file system.
tar(1)*	tar(1)	Create tape archives and add or extract files. (*: To create the archives with extent attribute information, using sfxcpio(1).)
sfxtone(1M)	tunefs(1M)	Tune up the file system.
sfxdump(1M)	ufsdump(1M)	Dump the file system.
sfxrestore(1M)	ufsrestore(1M)	Restore from dump.
-	volcopy(1M)	Make an image copy of a file system.

-: no correspondence command.

Part 3 Global File Services Shared File System

Chapter 12 File System Functions

This chapter describes the functions and features of the GFS Shared File System.

12.1 File System Overview

The GFS Shared File System is a shared file system that allows simultaneous access from multiple Solaris OE to which a **shared disk device** is connected. (A file system of this type is referred to as a shared file system. A file system such as UFS or **GFS Local File System** that will be used only within a **node** is referred to as a local file system.)

The GFS Shared File System is an optimum shared file system for business uses that are API-compatible with UFS or the GFS Local File System, and provides high reliability and high performance.

The GFS Shared File System maintains consistency of **file data** even with updating from multiple nodes, enabling data transfer by a distributed application with a conventional API when the application is executed on multiple nodes.

Also, continuous file operation on other nodes is assured even though one node fails, making the GFS Shared File System especially suitable for an environment that requires high availability of the file system.

The GFS Shared File System is available on the following system:

- 32-bit/64-bit of Solaris 8 OE
- 32-bit/64-bit of Solaris 9 OE

The GFS Shared File System has the following functions:

- Simultaneous shared access from multiple nodes to files or file systems
- Maintaining consistency for **file data** reference and updating from multiple nodes
- File access using a file cache on each node
- Continuous file operation on other nodes if one node fails while maintaining file system consistency
- High-speed file system recovery function
- High-speed I/O processing by contiguous block assignment to areas in a file
- Support of multi-volumes to implement I/O processing load distribution and large-capacity files
- Extension of file system size without rebuilding file system
- GUI-based file system operation using a Web browser

Like the UFS file system, the GFS Shared File System has the following major application interfaces:

- **64-bit file system interface**
- **direct I/O**
- **file lock**
- **quota function**
- **ACL**



See

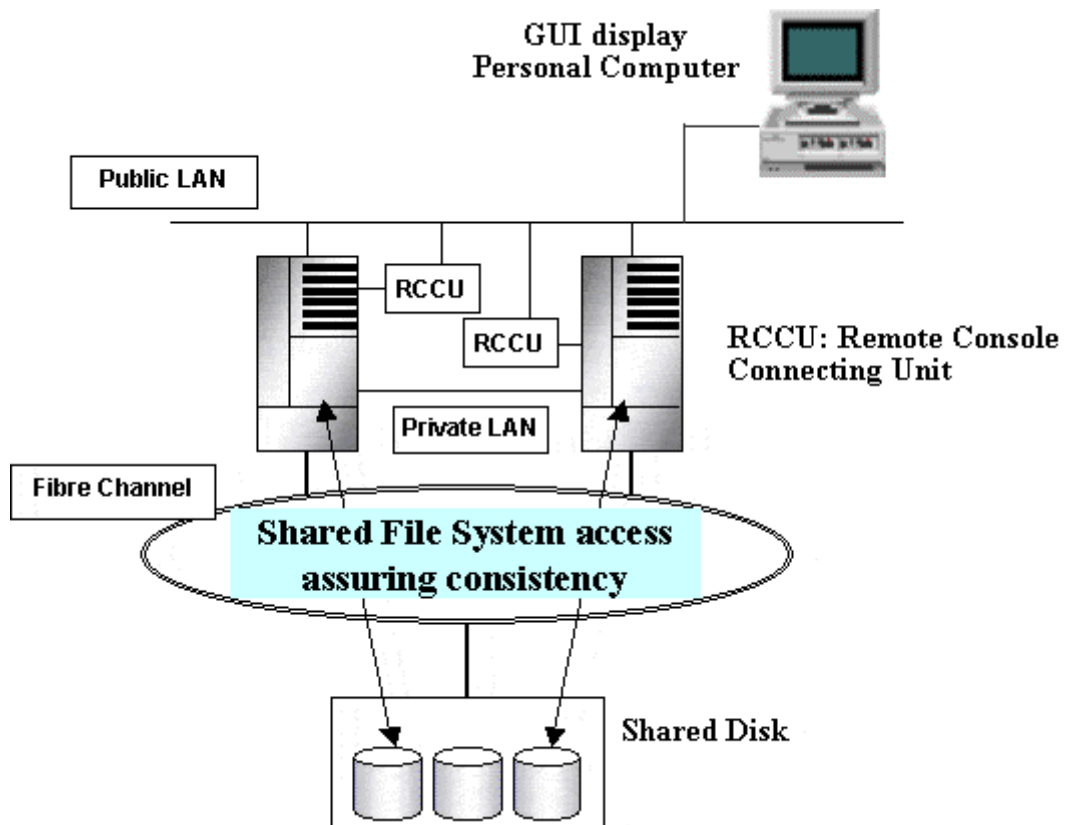
For the functions that are unavailable with the GFS Shared File System, see [Section 12.10.2, "Unavailable functions."](#)

12.1.1 Basic hardware configuration for use of file system

The following basic configuration is required to build the GFS Shared File System:

- Shared disk device which provides simultaneous access between **nodes** sharing the file system
- One or more independent private LANs for inter-node survival monitoring which is used for **PRIMECLUSTER**.
- Remote Console Connecting Unit
- GUI display personal computer or Solaris OE computer with bit map
- Public LAN

Figure Basic hardware configuration



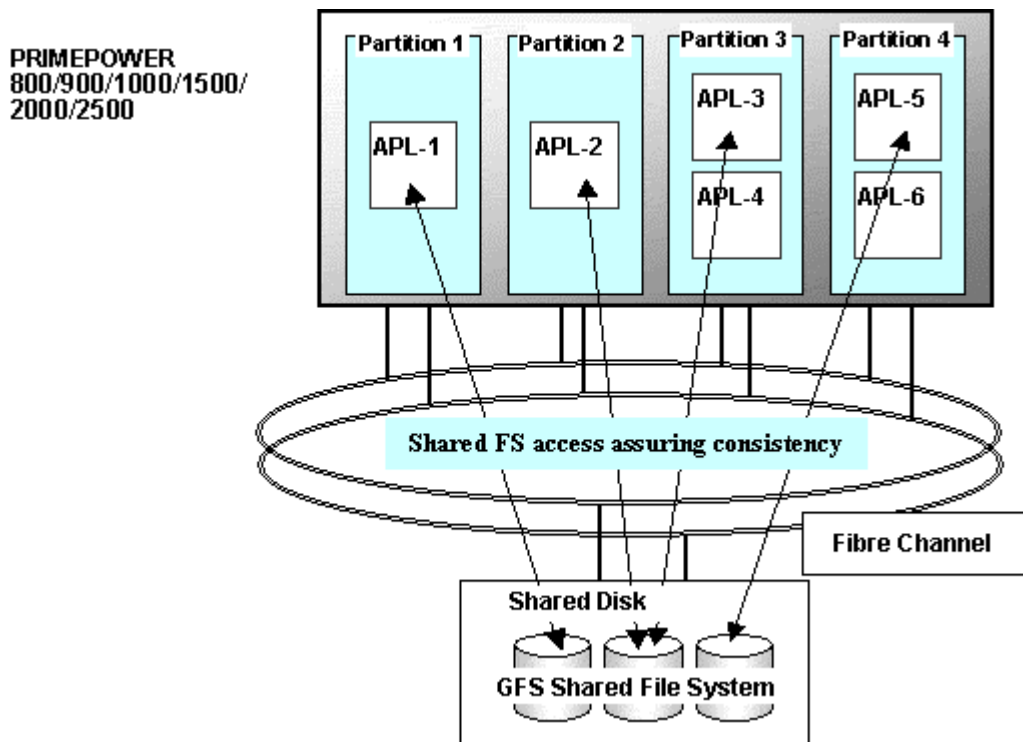
12.1.2 File system sharing with PRIMEPOWER

800/900/1000/1500/2000/2500

The PRIMEPOWER 800/900/1000/1500/2000/2500 can be used for multiple nodes by splitting one frame into multiple partitions^(*). Using the GFS Shared File System in this configuration allows a file system to be shared between nodes.

(*1: Partitions as used here refer to nodes split into multiple nodes through logical partitioning of the system.)

Figure Sharing file system with PRIMEPOWER 800/900/1000/1500/2000/2500

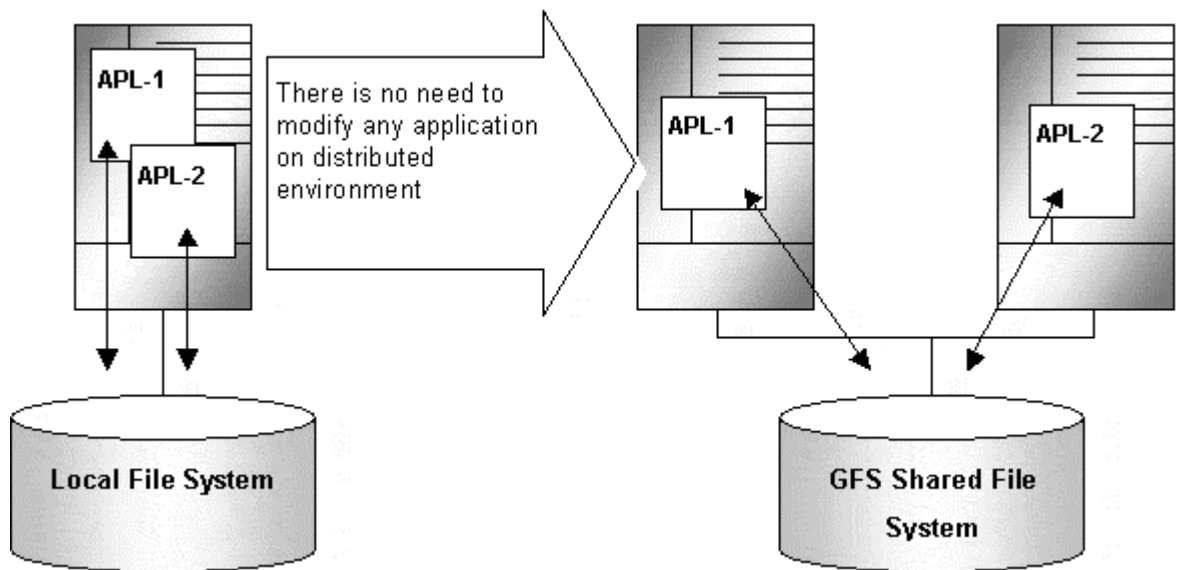


12.2 Simultaneous Shared Access Maintaining Coherency

The GFS Shared File System implements simultaneous referencing and updating from multiple **nodes** to a file system on a **shared disk device**. The GFS Shared File System maintains consistency for updating from multiple nodes to files or file systems. The **file lock** function spanning nodes can be used with a conventional API.

In this way, applications at multiple nodes can perform exclusive updating of mutual **file data** and reading of the latest data. A conventional UNIX file system API, such as file lock, can do these operations.

Figure Distributed execution of file related application



12.3 High Availability

The GFS Shared File System allows continuous access to file systems if a node failure or disk block failure occurs.

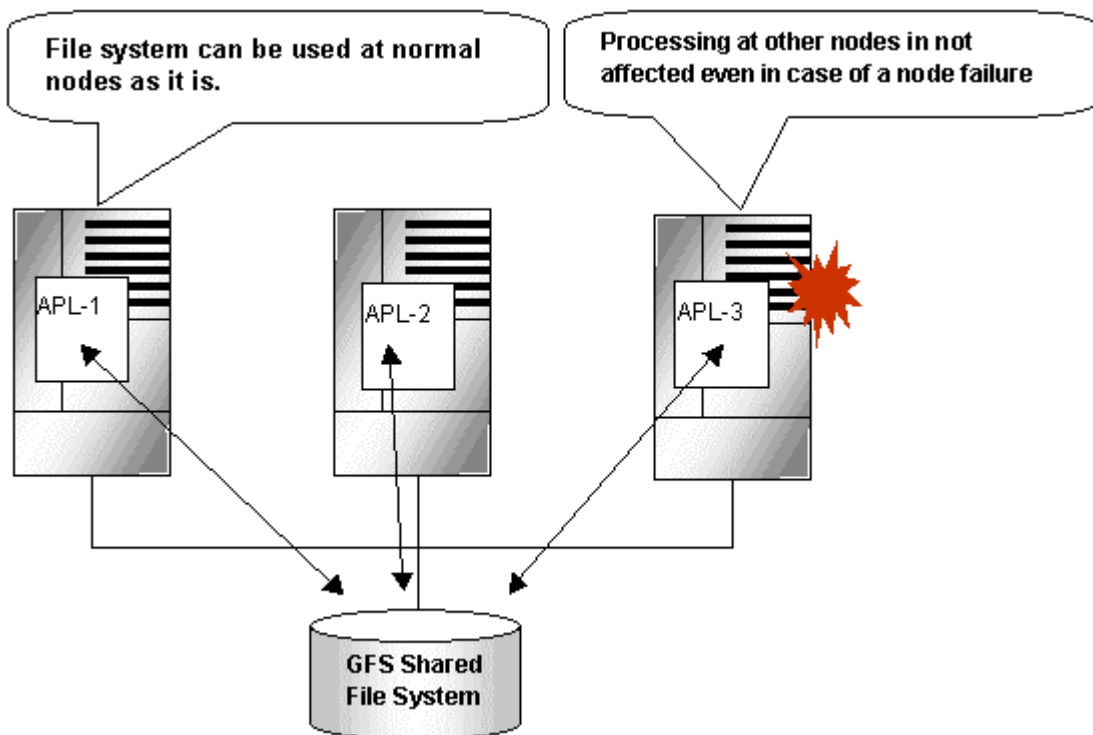
12.3.1 Uninterrupted operation in case of node failure

If one **node** fails when the GFS Shared File System is being used from multiple nodes, file access from the other nodes can be continued. The GFS Shared File System automatically recovers the file system from the other nodes. In other words, the processing of the application programs operating on the other nodes can be continued without causing a file system operation error.



For more information on the uninterrupted operation function, see [Chapter 14, "Failure Recovery."](#)

Figure Uninterrupted operation in case of node failure



12.3.2 High-speed file system recovery

In case of a **node** failure, `fsck_sfcfs (1M)` must be executed for ordinary file systems to recover consistency. In most file systems, it is necessary to inspect all of the **meta-data** for the systems in order to recover consistency. If a node failure occurs, a large amount of time may be required until the file systems can actually be used.

The GFS Shared File System records operations that have changed the file system structure in an area called update log. These operations include file creation and deletion. Using the log in the **update log area** allows the file system to recover from a system failure in less than a minute.

When recovering from a system failure, the GFS Shared File System retrieves the **update log** during the recovery process. The file system then decides whether to invalidate or complete the file system operation that was in process, and reflects the result. The file system structure can then be mounted and used without fully checking it.

As mentioned in the previous section, the GFS Shared File System operating on multiple **nodes** does not require execution of `fsck_sfcfs(1M)` because consistency recovery processing is performed in case of a node failure.



Note

The `fsck_sfcfs(1M)` full check mode is also provided. To recover the file system from a hardware failure on the disk, execution of full check `fsck_sfcfs(1M)` may be required.

A function for generating an **update log area** in another **partition** is provided in order not to lower file system performance during the update log I/O process, see [Appendix “F.2.14 sfcnewfs \(1M\).”](#)

12.3.3 Area reassignment in case of disk block failure

The GFS Shared File System automatically assigns another disk block to a new **meta-data area** if a disk block hardware failure occurs.

Reassignment of the disk block allows continuous file system processing if a disk failure occurs only on a specific block.



Note

This function only suppresses use of a block where an I/O error has occurred. If a request to use the same block is issued, however, an I/O error may occur again. In the case of an I/O error, the error must be corrected because the response time for that request increases. If a block reassignment occurs as a result of an I/O error, backup the file system, correct the hardware failure by replacing the failed disk with a new one, and restore the data to recover the file system.

12.3.4 Operation Continuity If the LAN Fault Occurs

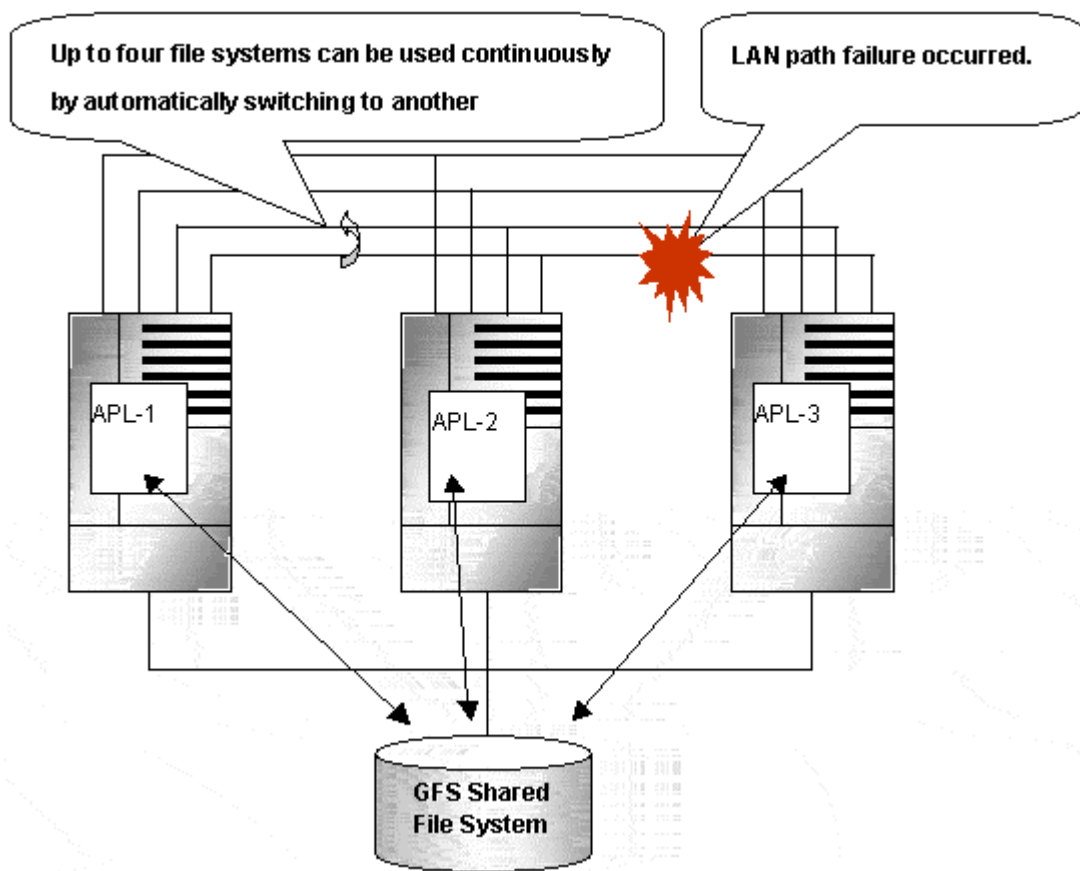
Two or more communication paths can be set up from AC to MDS. Since an alternate communication path is automatically discovered when a LAN failure occurs, file system operation can continue.

A maximum of four communication paths can be set up. For example, if four paths are set up and three network interface cards fail, file system operation can continue. Private LANs may be specified for any or all of the four communication paths.



For details on AC and MDS, see [Section 13.2, "Component composition."](#)

Figure Operation Continuity If the LAN Fault Occurs



12.4 Performance

The GFS Shared File System has the following functions for implemented for high speed file system access.

12.4.1 Data access performance

The GFS Shared File System allows access to the file system on the **shared disk device** from multiple **nodes**. Conventional distributed file systems transfer **file data** from management servers to clients that have actually issued an access request via a LAN in the network. However, the GFS Shared File System accesses the disk directly from a request node, reducing network load and speeding up response time for a read or write request in comparison with NFS.

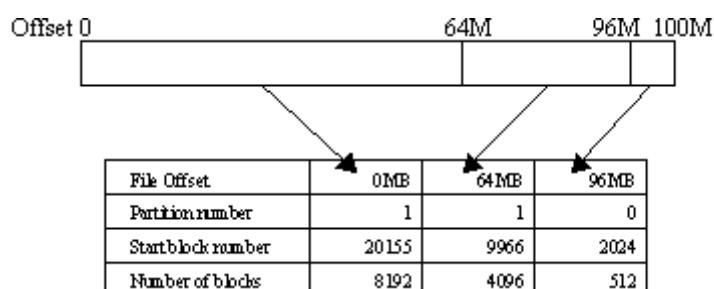
12.4.2 Contiguous block assignment

The GFS Shared File System assigns contiguous blocks to **file data** to take better advantage of batch I/O processing to improve the file system performance. The GFS Shared File System manages area on the **extent-base** for assigning contiguous blocks. For the range of blocks that can be acquired contiguously, the file system manages area assignment using a file offset, a start block number, and the number of blocks used.

The file system also takes into consideration the possibility of contiguous block assignment when extending the file.

The following picture shows a block assignment example for a file has three extents: 0, 64MB, 96MB file offset and 64MB, 32MB, 4MB extent length respectively.

Figure Contiguous block assignment



Empty file data areas are also managed in the extent base to implement high-speed assignment of optimum empty areas.

12.4.3 Multi-partition configuration

The GFS Shared file system provides a function that unifies **partitions** into one file system.

With the GFS Shared file system, you can easily solve the problem of there being insufficient area by adding partitions.

In multi-partition configuration, a **round robin allocation** system enables the use of the **file data area** in different partitions, thus improving file system performance by sharing the I/O load between different disks.

meta-data partitions can be added when **meta-data** fills up the disk as a result of an increase in the amount of **file data**, meta-data partitions can be added when meta-data fills up the disk as a result of an increase in the amount of **file data**.

12.4.4 Individualized meta-cache management

The GFS Shared File System constructs its cache management of **meta-data** in an individualized manner. Many conventional file systems manage meta-data in a uniform manner. However, the GFS Shared File System provides individualized management of **i-node**, directory block, and indirect block areas on disks that takes into consideration the characteristics of the access, improving the cache-hit ratio and reducing resources used.

12.5 Extendability

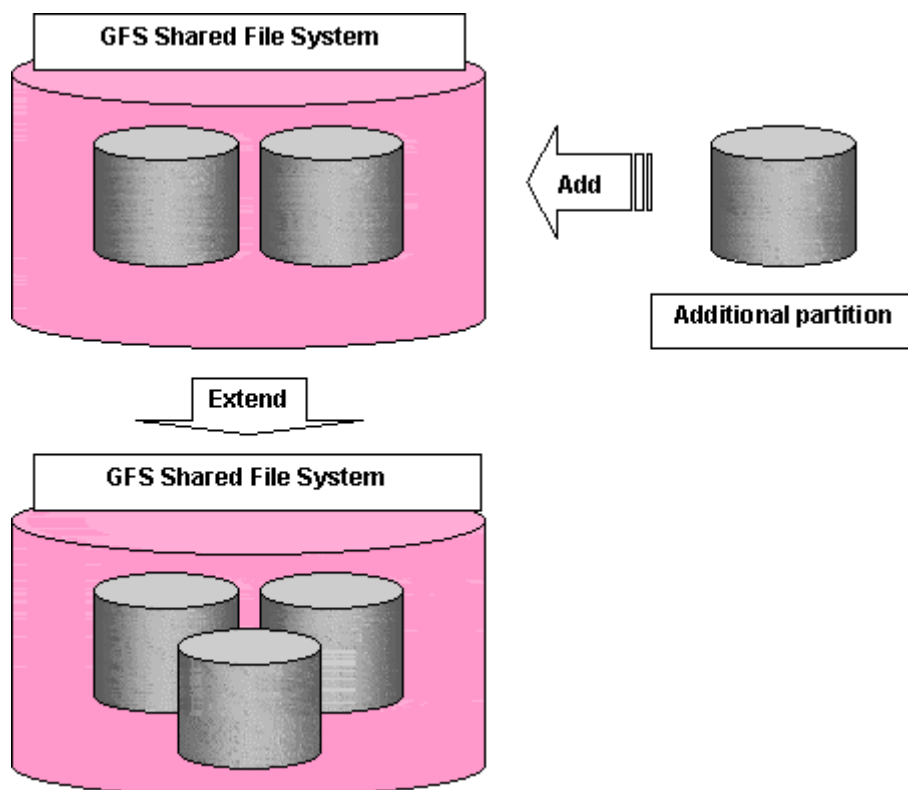
The GFS Shared File System is capable of easily extending its data partitions by specifying free disk partitions, providing a quick solution if the file system has too few free areas.

12.5.1 File system extension

The GFS Shared File System adds **partitions** to a conventional system file, thus solving the problems of there being an area shortfall without taking additional time for the backup and/or re-creation of the file system.

file data partitions and **meta-data partitions** can be added even when the file system is mounted, allowing users to overcome area shortfalls without stopping regular operation.

Figure File system extension

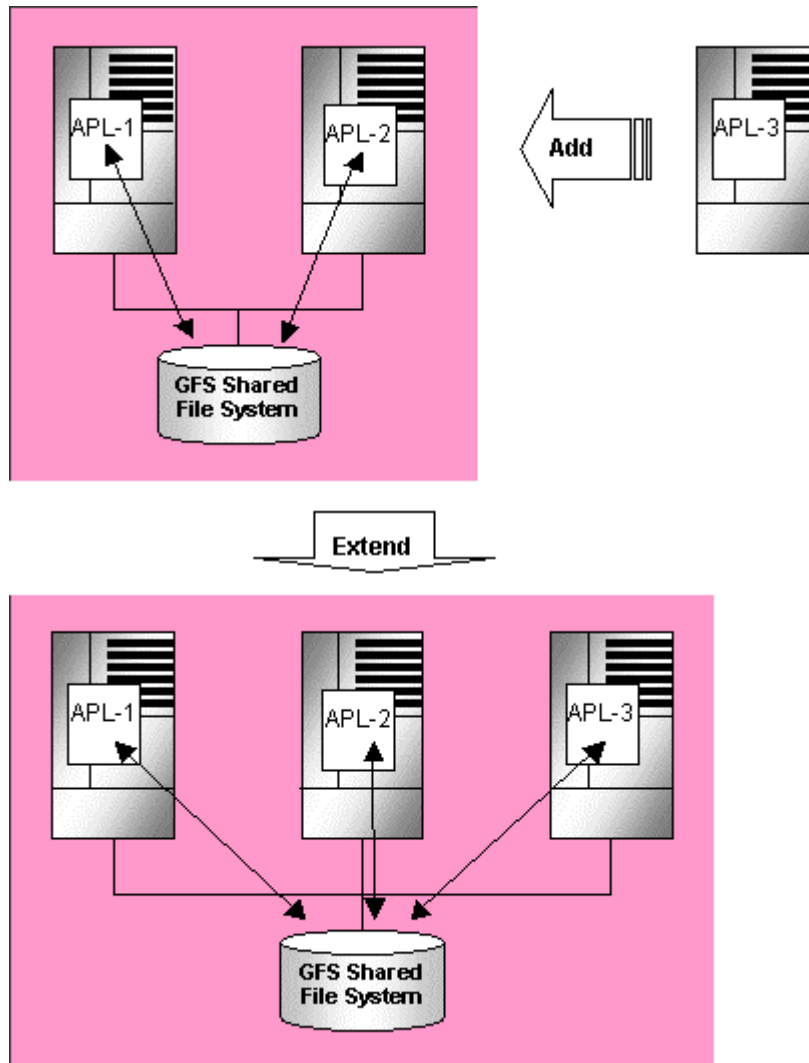


12.5.2 Adding shared nodes

The GFS Shared File System has the capability of setting or adding up to 4 **nodes** that can share the system.

If, at installation, two nodes had been set up to share a file system, it is possible to later increase the number of shared nodes to three or four.

Figure File system extension



12.6 High-speed backup and restoration of a file system

The GFS Shared File System, cooperating with **GDS** provides a snapshot function that allows users to create a replica (snapshot) of a GFS shared file system. Snapshots allow operations such as non-disruptive backups or data analysis to run concurrently with core business applications.

Furthermore, when GDS is combined with the advanced copy feature of our company's ETERNUS3000/GR720/GR730/GR740 disk array, a high-speed snapshot copy can be created instantly without imposing a load on the server or SAN.

Snapshot copy is instantly logically complete. File system access need only be stopped very briefly while the copy logically completes, minimizing the impact on core business applications.

Even if the GFS shared file system is mounted, a snapshot copy can be created. If a snapshot is created while an application is accessing the file system, the volume data may not be complete. To create a snapshot while running an application, the application or data base must ensure data integrity.

The restored file system can be immediately used after the logical completion of restoration.



See

For detail, see [Section 22.7, " High speed backup and restoration of a GFS shared file system"](#) in this manual and Section "Snapshot by Detaching Slice" in Global Disk Services Configuration and Administration Manual.

12.7 File Extension Attribute Information

The GFS Shared File System supports the functions of the file extension attribute information.

The file extension attribute information can be specified for each file otherwise the default value for the directory is used.

The default value of the file extension attribute information is set up for directories. Once set for a directory, the attribute takeover function automatically sets the attribute information for newly created files and directories within that directory. Setting the extent attribute information for a directory does not change the extent attribute information of existing files or directories within the directory.

Each of the file system extension attribute supported by the GFS Shared File System is described below.



For details, see [Section 21.3, "file extension attribute information."](#)

12.7.1 Extent Attribute Information

The functions of the GFS Shared File System include pre allocation, area extension in specific units and guarantee of a continuous area. The **extent attribute information** enables these functions. These functions enhance data access performance by ensuring the file data block allocation and the file data block continuity.

12.7.2 Ignore Update Last Access Time Attribute Information

This function prevents updating of the last access time for the specified files and directories. The access load to the file system and the meta-data for a frequently accessed file is reduced. To enable this function, use **sfctime(1M)** to set the update last access time attribute information for the desired file.

12.7.3 Ignore Update Last Modify Time Attribute Information

This function prevents updating of the last modified time for the specified files and directories. The access load to the file system and the meta-data for a frequently accessed file is reduced. To enable this function, use **sfcmtime(1M)** to set the update last modify time attribute information for the desired file.

12.7.4 File Data Allocation Attribute Information

New data areas for the specific file are allocated from a specific file data partition. This function restricts allocation to a specific allocation device within a file system that is composed of more than one file data partition. To enable this function, use **sfcdv(1M)** to set the file data allocation attribute information for the desired file.

12.8 Operability

12.8.1 GUI

The GFS Shared File System uses a GUI that allows a Web browser to be used to create, delete, operate, and modify its file systems and monitor its status.

12.9 Upper Limits of the File System

The following table lists the upper limits of the GFS Shared File System on a file system basis:

Table Upper limits of file system

Item	Upper limit
Maximum file system capacity	32 terabyte (1 terabyte per one partition)
Maximum file size	1 terabyte
Maximum number of sharable nodes	4 nodes
Maximum directory size	2 gigabytes
Maximum number of partitions consisting of a file system	32 partitions
Maximum number of i-nodes per file system	128 mega

The maximum number of the GFS Shared File System in a cluster system is 30.

12.10 Notes

Below are notes on using the GFS Shared File System.

12.10.1 Relationships with Other Components of PRIMECLUSTER

CIP must have been set up, as it is used for the file system monitoring facility (FsRM) of the GFS Shared File System.



For details to set up CIP, see the “*PRIMECLUSTER Cluster Foundation Configuration and Administration Guide*.”

GDS is used to create logical volumes for use by the GFS Shared File System.



For notes, see [Section 12.10.5, "Notes on use of logical volumes of GDS"](#).

For details to set up GDS, see the “*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*.”

The GFS Shared File System must work with the other nodes in the same cluster and must recognize that each node works correctly. When a node downs, it turns to be the state called LEFTCLUSTER. If there is a node with this state, it is impossible to change the management state of the GFS Shared File System. To keep the GFS Shared File System operational whenever a node downs, set up the Shutdown Facility so that the LEFTCLUSTER state will turn into the DOWN state automatically.



For details of the Shutdown Facility and the Monitoring Agents, refer to the following manuals.

- Section 5.5.2, “Setting up the Shutdown Facility” of the “*PRIMECLUSTER Configuration and Administration Guide*”
- Section 3.3.1, “PRIMECLUSTER SF” of the “*PRIMECLUSTER Concepts Guide*”
- Section 8, “Shutdown Facility (SF)” of the “*PRIMECLUSTER Cluster Foundation Configuration and Administration*”

12.10.2 Unavailable functions

The GFS Shared File System provides none of the following functions:

- Use as a root file system, /usr, /var, /opt
- lockfs(1M)
- Setting of hard link for a directory
- open(2) from another node to a mmap(2)
- mmap(2) from another node to a writeable
- open(2) open(2) from another node to a mmap(2) file for which **MAP_SHARED** and **PROT_WRITE** are specified
- mmap(2) which specifies **MAP_SHARED** and **PROT_WRITE** from another node to a writable open(2) file
- The strict file terminal management in the case of write(2) from another node to an open(2) file for which **O_APPEND** was specified
- Use of IPv6

- Setting extended file attributes, which is added in Solaris 9 OE
- Mount specifying non-blocking mandatory locking option, which is added in Solaris 9 OE

12.10.3 Service ports used by the GFS Shared File System

The GFS Shared File System reserves TCP service ports 9100 to 9200.

If these port numbers conflict with other applications, change the entry port number beginning with `sfcs-` in the `/etc/inet/services` file.

12.10.4 Notes on use as loopback virtual file system(lofs)

When using a GFS shared file system as a loopback virtual file system (lofs), the file system may not be stopped while any related lofs is mounted. The node will panic if the file system is stopped and a related lofs is mounted. When using a GFS Shared File System as a loopback virtual file system, unmount all related lofs before stopping the file system.

12.10.5 Notes on use of logical volumes of GDS

GDS is volume management software designed to improve the availability and operation manageability of information stored on disk units of a Storage Area Network (SAN). Various kinds of access control can be performed on the logical volumes of GDS to protect data against damage due to unauthorized access.

To use a logical volume of GDS in the GFS Shared File System, make the following settings:

- The type of disk class to which the logical volume belongs is shared.
- All nodes sharing the GFS Shared File System are specified in the scope of the disk cluster to which the logical volume belongs.
- When the attribute lock mode of the logical volume is `lock=off`, the volume is started automatically.
- The attribute access mode of the logical volume is set as read-write.

When the GDS logical volumes are stopped, access is totally prohibited. Thus, GFS Shared File System commands can be used only to active GDS logical volumes.



For the operations of disk classes to which GDS logical volumes belong, refer to the description of class operations under "Operation using Global Disk Services Management View" in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

For the operations of logical volumes of GDS, refer to the description of volume operations under "Operation using Global Disk Services Management View" in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

12.10.6 Note on stopping a node

Use **shutdown(1M)** commands to stop a node.

If **reboot(1M)**, **halt(1M)**, **poweroff(1M)**, or **uadmin(1M)** is used to stop a node, GFS may not work correctly.

Chapter 13 File System Structure

This chapter describes the structure of the GFS Shared File System.

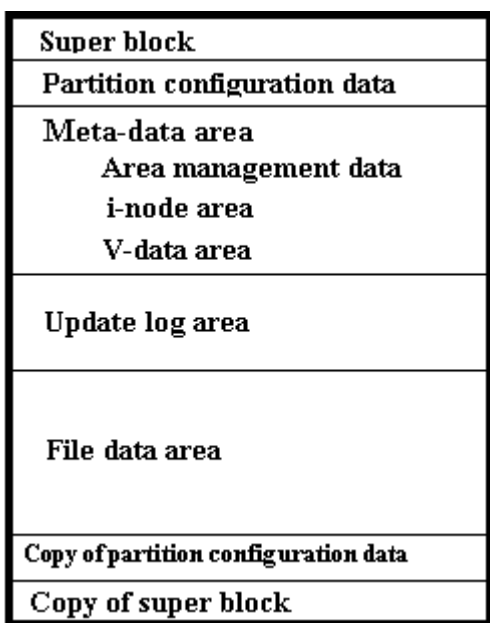
13.1 Disk Structure of File System

This section describes the disk structure of the GFS Shared File System.

As illustrated below, the disk structure of the GFS Shared File System consists of the following elements:

- **super block**
- **partition configuration data**
- **meta-data area (area management data, i-node area, V-data area)**
- **update log area**
- **file data area**

Figure Disk structure of the GFS Shared File System



The GFS Shared File System manages the V-data area in units of 1,024 bytes and the file data area in unit of 8,192 bytes.

Conventional file systems have decentralized meta-data areas on their disks. However, the GFS Shared File System centralizes **meta-data** to improve meta-data access performance. Also, localizing updated portions of the file system decreases the recovery time required to process the update log.

The **GFS Local File System** has also the same structure.

13.1.1 Super block

The super block contains the file system type, the creation and update dates, label data, the size and layout of the file system, and a data history of empty areas.

The **super block copy** at the end of the file system exists to allow continuous processing if a disk device block failure occurs in the primary super block area.

13.1.2 Partition configuration data

The GFS Shared File System is able to combine multiple **partitions** into a file system. The partition configuration data contains information about the partitions comprising a file system and the allocation of data areas in a partition.

This area is used when a file system configuration or a shared device are changed.

Similar to the super block, the partition configuration data is important for operation of the file system and for protection from disk device block failures a backup copy exists at the end of the partition.

13.1.3 Meta-data area

The meta-data area is an area for storing meta-data, which exists only in a **representative partition**.

13.1.3.1 Area management data

The area management data includes the allocation information of **i-node**, **V-data** and **file data**.

File data allocation information is managed in **extent-base** while combining a part to include extent information with a list managed with a degree of continuity.

[sfcnews\(1M\)](#) based on options specified creates the area for i-node, V-data and area management data. However, these areas can be dynamically increased when **meta-data partitions** are added during file system extension.

13.1.3.2 i-node area

The i-node area is a data structure containing file data. Each i-node stores pointers to the extent attribute, which includes a file type, a file length, file owner and group IDs, access permissions, and file data. There is one i-node for each file.

13.1.3.3 V-data area

The V-data area is defined to include a directory block, a symbolic link path name, and an indirect block area. The V-data area is managed in units of 1,024 bytes for efficient use of directory blocks.

13.1.4 Update log area

The **update log area** stores a history of file system structure changes for high-speed file system recovery. This area is maintained as a cyclic log.

The update log area contains data about the following processes:

- Acquiring or releasing an **i-node**
- Updating i-node data
- Allocating or releasing an extent, which is a group of contiguous file system data blocks handled as a single unit
- Acquiring or releasing **V-data**
- Updating V-data

The GFS Shared File System assures that data is written to the **V-data area** before the file system structure is updated. If a system failure occurs, file system consistency is restored by using [fsck_sfcfs\(1M\)](#) to either invalidate or re-execute unprocessed changes to the file system. Also, system failure processing is conducted. Only changes to the file system structure are recorded in the update log; file-data is not recorded.

13.1.5 File data area

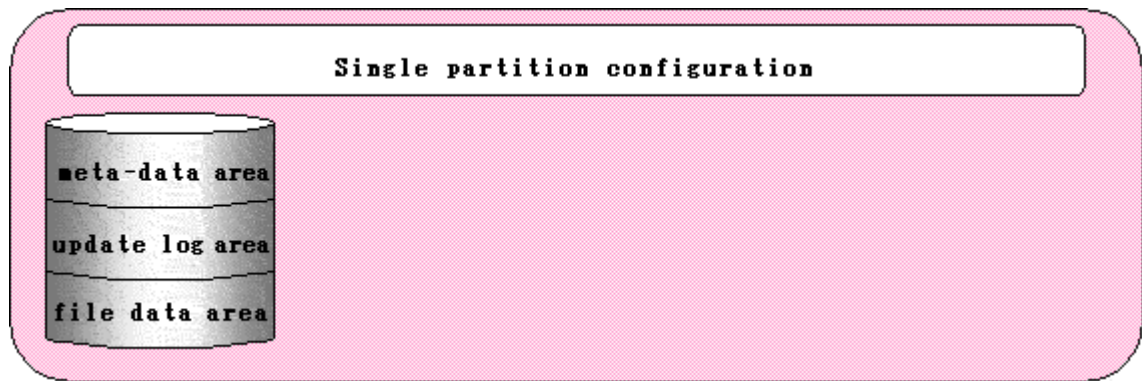
The file data area is an area for storing file data. In these areas, the minimum block size is 8192 bytes and the blocks are managed for contiguous allocation in a file area..

13.1.6 Partition configuration

The GFS Shared File System has a multi-partition configuration. Any of the partitions may consist of one or more partitions which can be allocated to one file system.

In a single partition configuration, an update log area, a meta-data area, and a file data area are allocated to one partition.

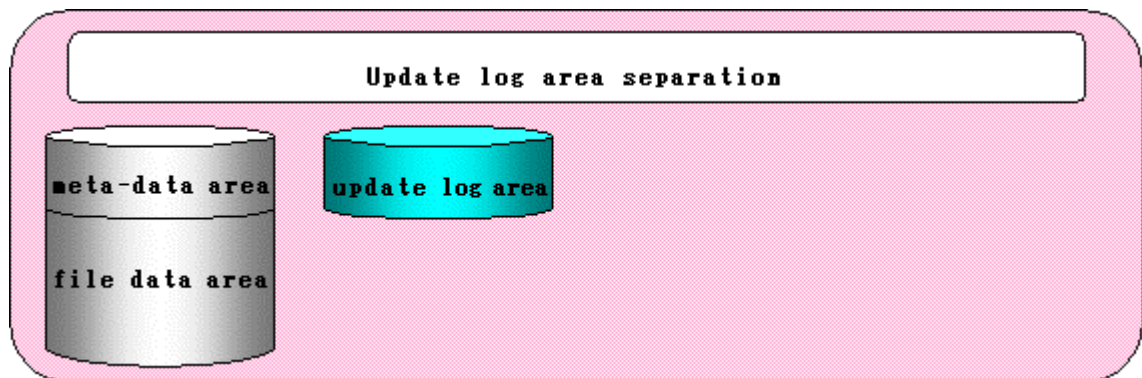
Figure Single partition configuration



Examples of multi-partition configurations are shown on the following pages.

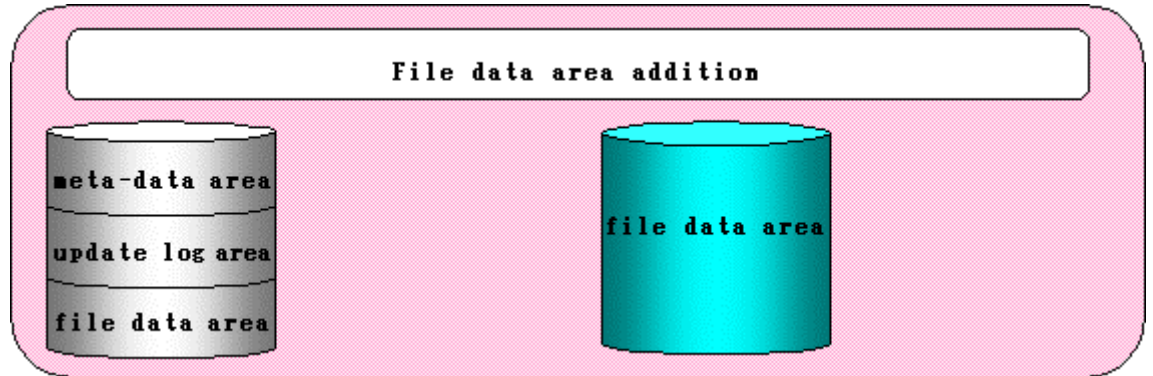
1. Separating the update log area to a separate partition (update log partition separation)

Figure Update log area separation



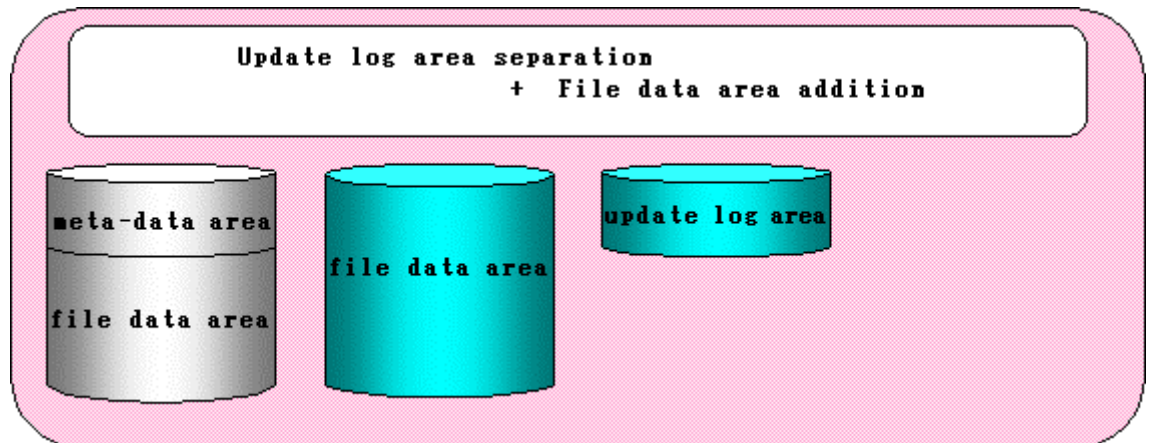
2. Adding a file data area to another partition. (file data area addition)

Figure File data area addition



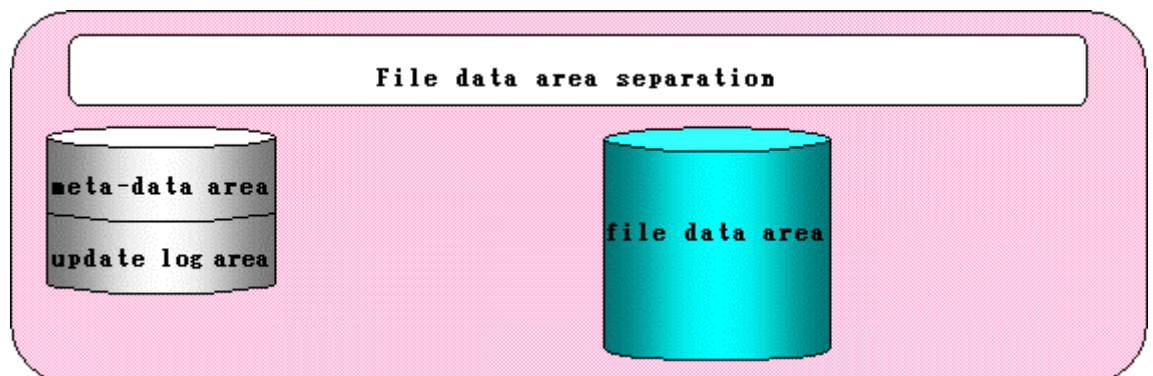
3. Multi-partition configuration with update log area separation and file data area addition

Figure Update log area separation and file data area addition



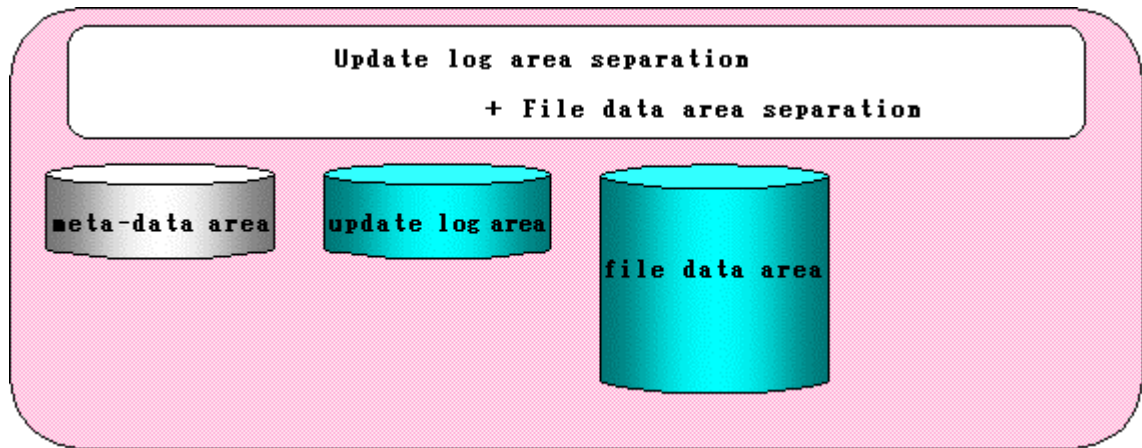
4. Partition configuration in which the file data area is separated from the representative partition.

Figure File data area separation



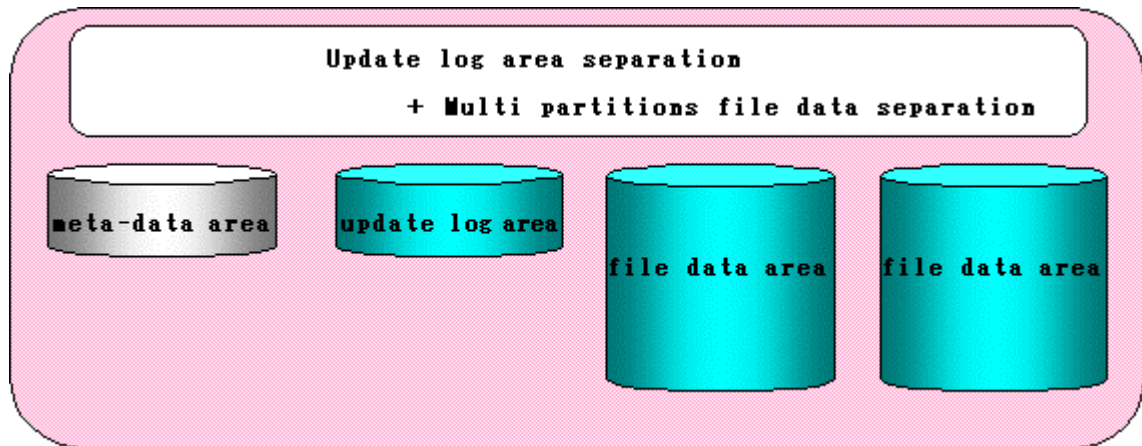
5. Partition configuration in which the meta-data area, update log area, and file data area are independent.

Figure Update log area separation, and file data area separation



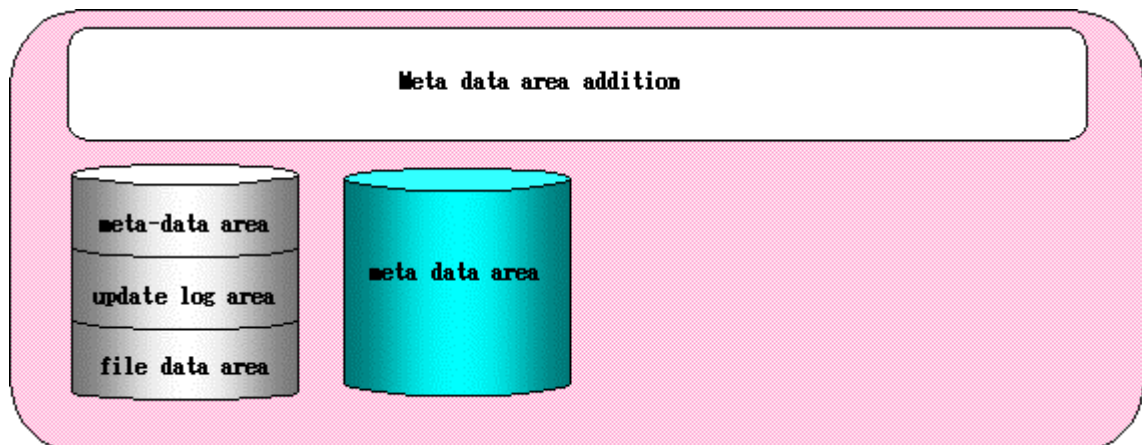
6. Partition configuration in which the meta-data area, update log area, and multiple file data areas are independent.

Figure Update log area separation, and multiple file data area separation



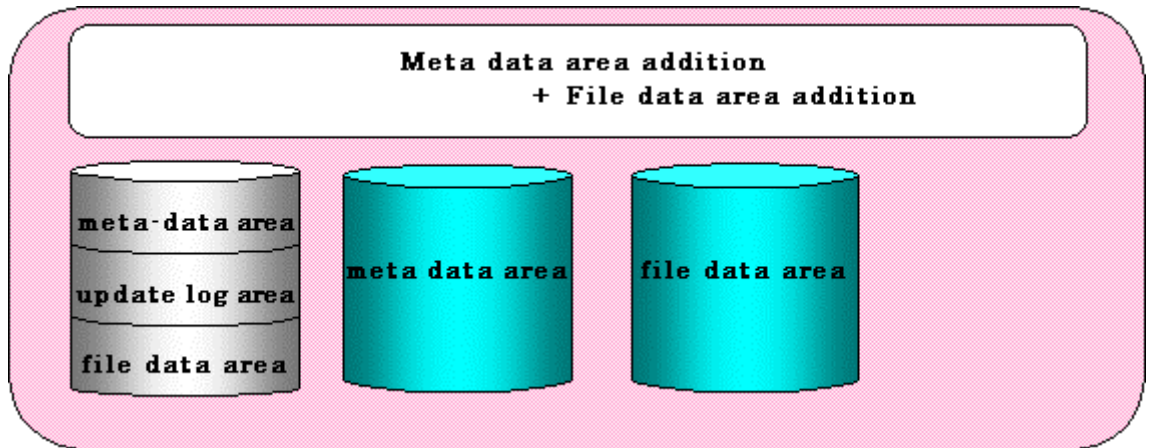
7. Adding the meta-data area to another partition. (meta-data area addition)

Figure Meta-data area addition



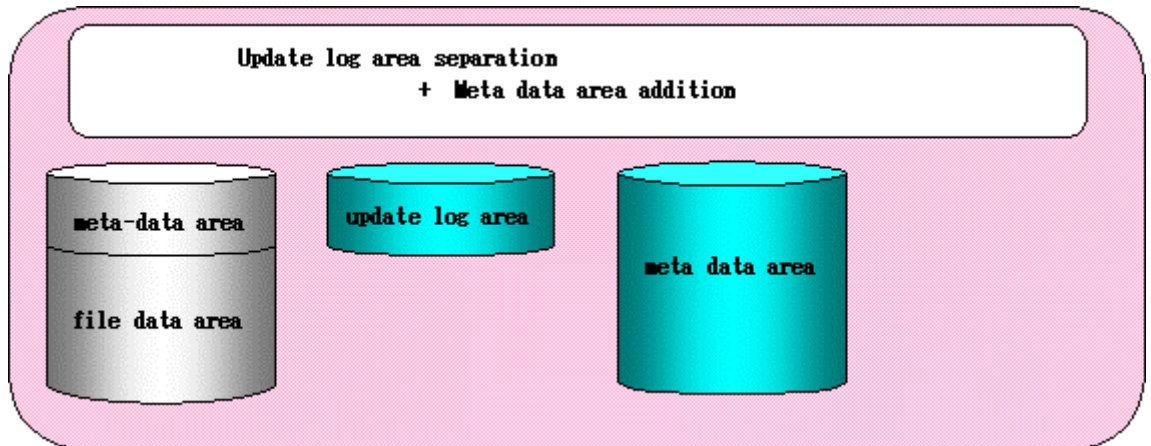
8. Adding a meta-data area and file data area to another partition. (meta-data area and file data area addition)

Figure Meta-data area and File data area addition



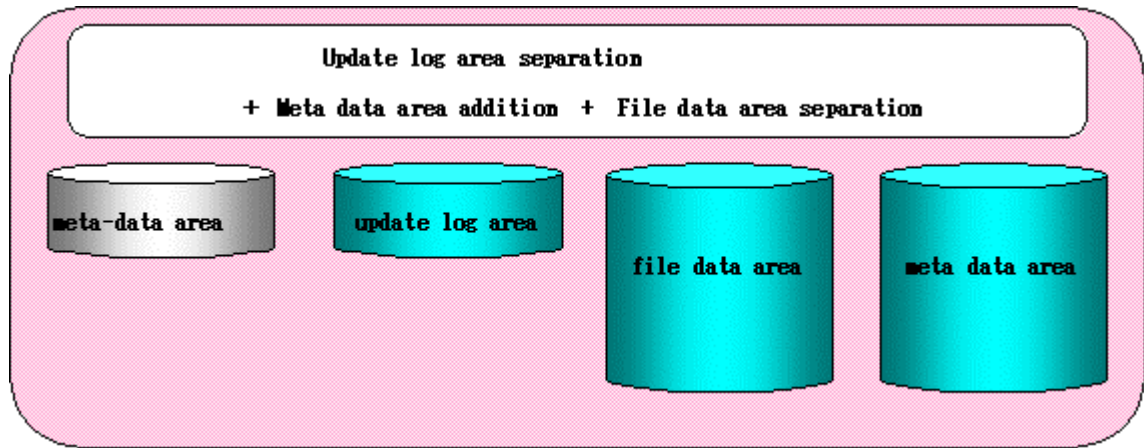
9. Multi-partition configuration with update log area separation and meta-data area addition

Figure Update log area separation and meta-data area addition



10. Multi-partition configuration with update log area separation and meta-data area addition and file data area separation.

Figure Update log area separation and Meta-data area addition and File data area separation



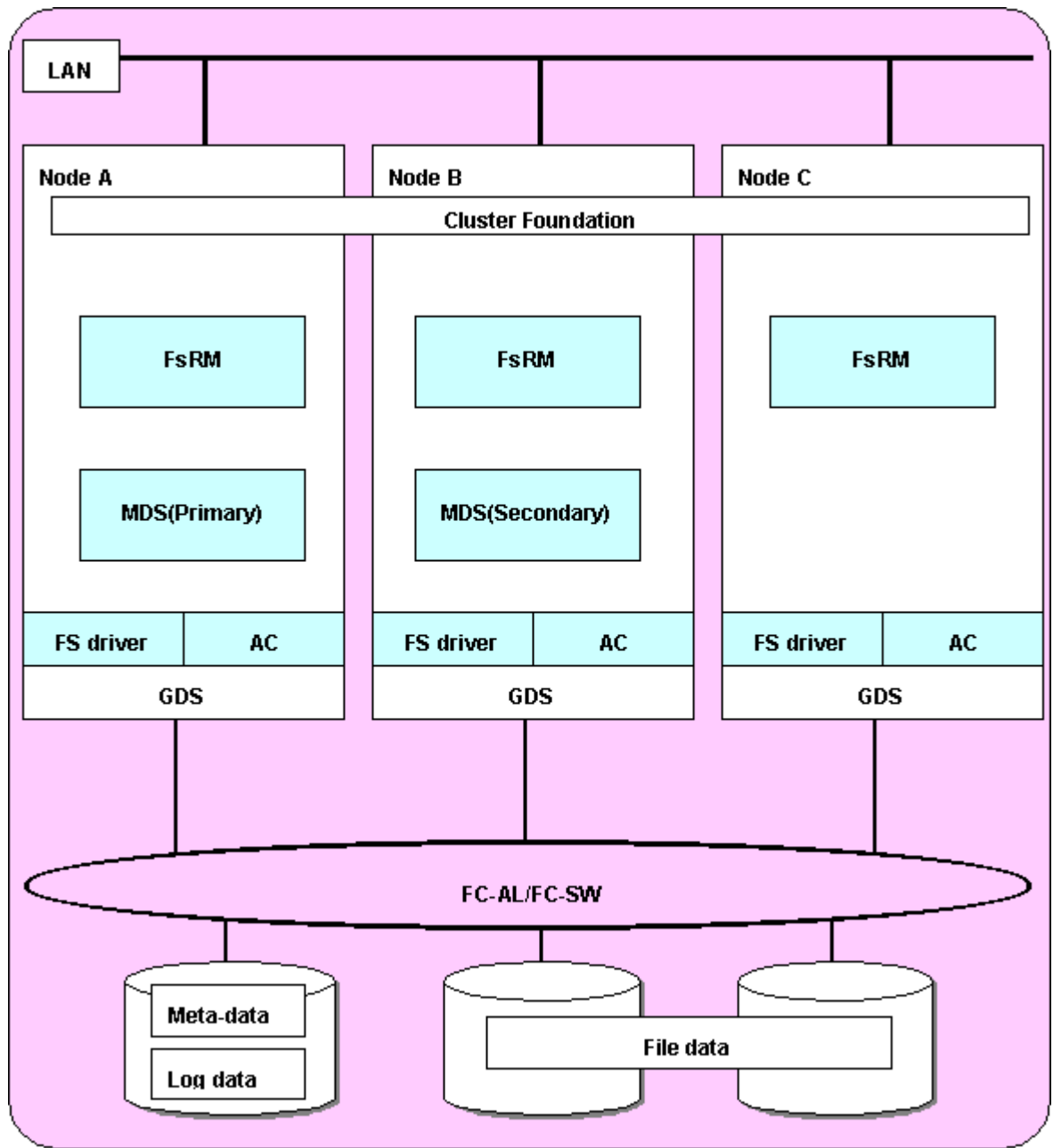
In a multi-partition configuration, **super block** and **partition configuration data** are maintained in all partitions.

13.2 Component configuration

The component configuration of the GFS Shared File System is illustrated in the following figure. The GFS Shared File System consists mainly of three components:

1. **MDS** (Meta-data server)
The file system server function component (sfcsmsg) of the GFS Shared File System
2. **AC** (Access client)
The file system client function component (kernel component AC) of the GFS Shared File System
3. **file system driver**
The I/F functional portion for control of the GFS Shared File System.
4. Monitoring daemons
sfcfrmd, sfcprmd, sfcfsd, sfchnsd, sfcpcnd

Figure Component composition



Each component is described in the following sections:

13.2.1 MDS(Meta-data Server)

The MDS is a server daemon that manages the meta-data of the GFS Shared File System. It operates as a user process. Each file system has two processes: primary MDS and secondary MDS. Two or more nodes of those that constitute a file system are predefined as **MDS nodes** and MDS operates on one of these nodes. If not set specifically, the MDS nodes are only the two nodes of the primary and secondary MDS node.

The secondary MDS is used for standby and runs on a different node from the primary MDS.

The MDS uses multithreading for parallel processing of the **AC** requests.

The MDS has the following major functions:

- Token management
- Meta-data area management
- Update log area management
- File data area management
- File system free area management
- AC node management

13.2.2 AC(Access Client)

This client processes requests from an application that accesses the GFS Shared File System. It operates within the kernel. Access clients exist on all the nodes of the scalable class where the GFS Shared File System is mounted. The AC allows cache management of meta-data and file data within the client.

13.2.3 File System Driver

The **file system driver** is a component that is used to create an interface that a command uses to control a file system. Data exchange with each **AC** can be done by a command.

13.2.4 Monitoring daemon

- **sfcfcmd**
Receives a request from the sfcfsd command and provides the communication paths to enable execution of the sfcfsd and sfcprmd commands and inter-node communications. Configuration database management and the management partition I/O functions are also provided.
- **sfcprmd**
Used to monitor the startup, stop, or failure of the processes that constitute the GFS Shared File System. When the configuration process starts or executes another daemon or command, a request is transmitted to sfcprmd, and then sfcprmd executes the daemon or a command as required.
The process from which a request is transmitted is referred to as the client process, while the process to be started from sfcprmd as required is referred to as the target process.
The target process is monitored by sfcprmd. When the target process terminates normally or abnormally, the status is posted to the client process.
- **sfcfsd**
Provides MDS/AC control functions such as the GFS Shared File System operation start/stop, failover of MDS, online node addition/deletion, and failback (migration) to the definition state of the file system configuration.
- **sfchnsd**
Used to provide functions such as node down event reception, domain-wide lock, membership information acquisition, and node state acquisition.
- **sfcpcnd**
Performs live monitoring of sfcprmd, sfcfcmd, or sfcclnd that is to be monitored. When the process to be monitored abnormally end (process down), this daemon makes system go down in order to maintain consistency of file systems.

13.3 File Access and Token Management

Each **AC** caches file data to improve file system performance. Since it is necessary to manage discard operation of cache to ensure consistency in each file system, the GFS Shared File System uses file-exclusive tokens (referred to hereafter as tokens) for management.

A token guarantees that the AC may access the file. The token contains information that indicates how an AC is allowed to access a particular file. An AC must have a token for a file when the AC accesses the file. Two types of tokens are provided: one that ensures **meta-data** access and another that ensures **file data** access.

- Meta-data access and token management
 - A meta-data token is required to access i-node and V-data.
- File data access and token management

Tokens are used to access file data in ordinary files. The two types of available tokens (write authority and read authority) are managed in units of logical blocks. Such operations as simultaneously reading from the same block and simultaneously reading from/writing to different blocks in the same file can be performed simultaneously by multiple shared nodes without degrading performance.

The file data token is required to access ordinary files.

Chapter 14 Failure Recovery

This chapter describes the following failure recovery functions of the GFS Shared File System:

- **MDS** failure recovery
- **AC** degeneration

14.1 MDS Failure Recovery

If an **MDS** fails, the GFS Shared File System automatically performs failure recovery to continue operation of the file system.

The following provides an overview of the operations performed if either the primary MDS or secondary MDS fails.

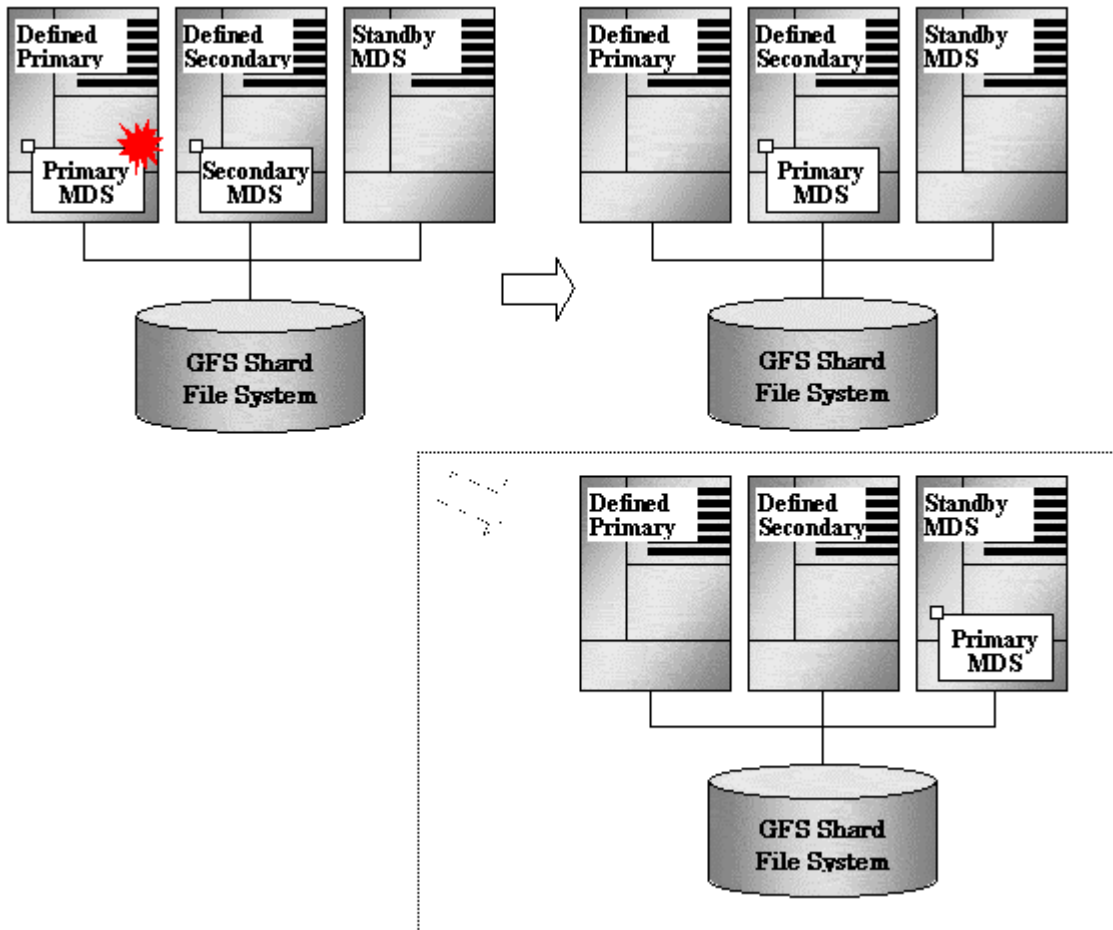
14.1.1 Automatic recovery for primary MDS failures

There is one case in which the standby secondary **MDS** becomes the primary MDS. This case occurs if both the primary and secondary MDSs are set up as operable and the primary MDS process fails. In this case, the following process is performed:

- 1) Switching the secondary MDS to the primary MDS

The secondary MDS operates as the primary MDS in place of a failed primary MDS.

In a file system in which three or more MDS nodes are set, if the secondary MDS does not work as the primary MDS after the specified time, a primary MDS can be activated from another MDS node to continue operations.



- 2) Replaying the **update log**

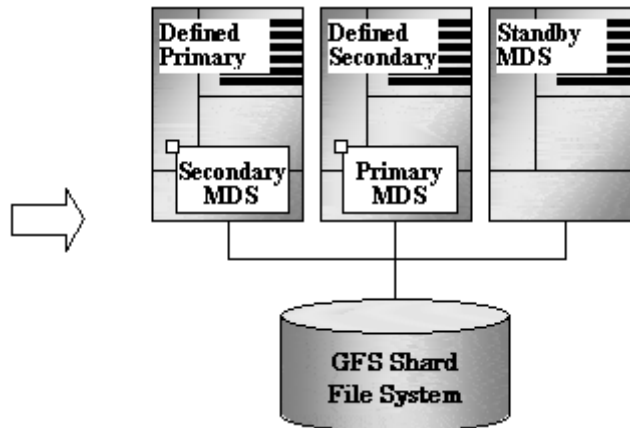
The new primary MDS replays the update log to ensure consistency of the file system.

- 3) Resuming processing after MDS switching

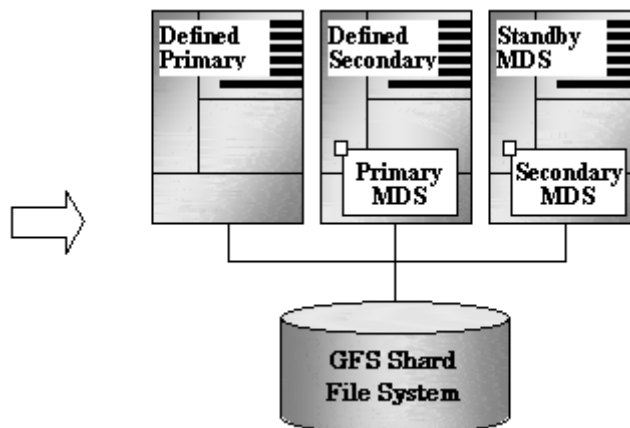
The AC sends a processing request to the new primary MDS. Until that request is accepted, all access to the file system is blocked.

- 4) Restarting the failed MDS

Usually, the failed MDS is restarted as the secondary MDS.



In a file system in which three or more MDS nodes are set, if the secondary MDS does not work after the specified time, a secondary MDS can be activated from another MDS node to continue operations.



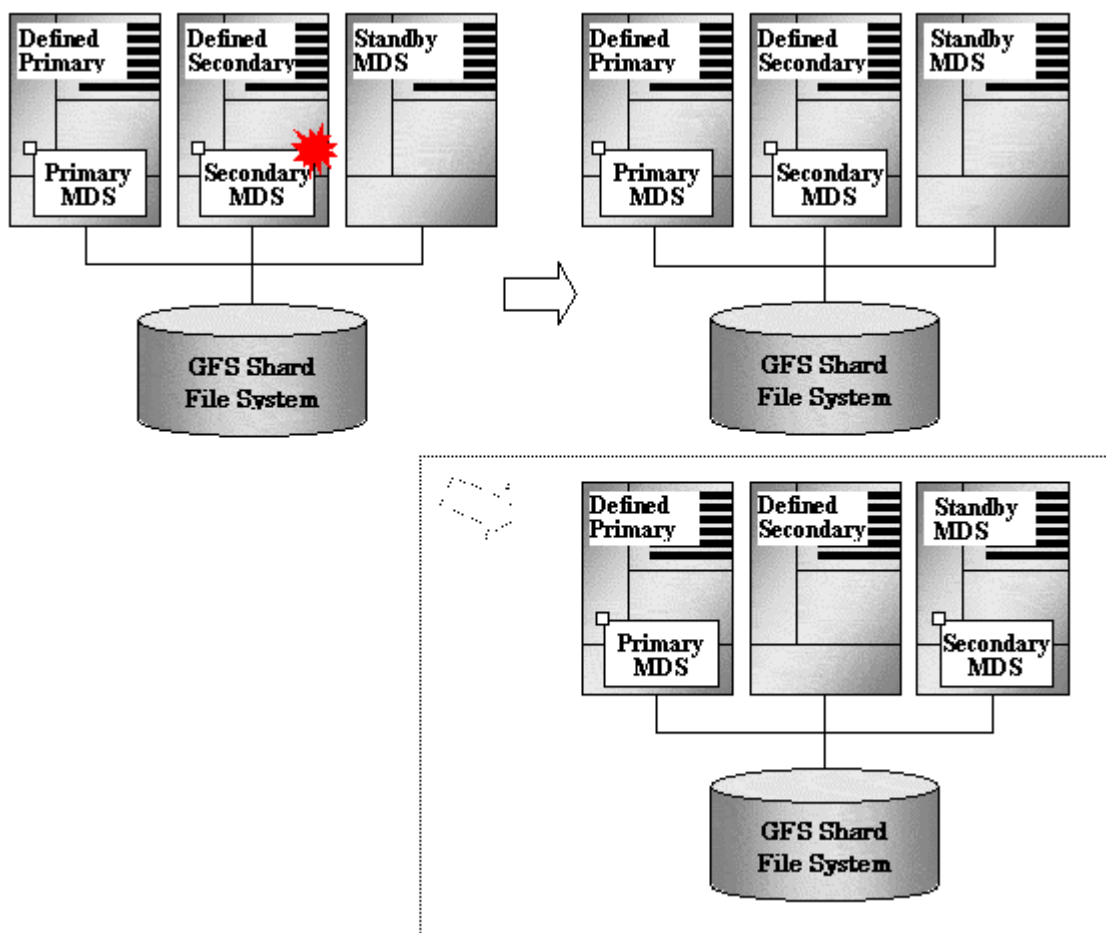
14.1.2 Automatic recovery for secondary MDS failures

If the secondary **MDS** process fails, operation is not affected. The secondary MDS is automatically started again as the secondary MDS in preparation for a failure of the primary MDS process. In this case, the following process is performed:

1) Restarting the failed MDS

The MDS is restarted as a secondary MDS.

In a file system in which three or more **MDS node** are set, if the secondary MDS does not work after the specified time, a secondary MDS can be activated from another MDS node to continue operations.



- 2) Making the restarted secondary MDS effective
The AC performs the processing necessary for recognizing the restarted secondary MDS.

14.1.3 Automatic recovery in case primary MDS goes down during primary MDS operation only

If the primary **MDS** goes down in a state in which the MDS nodes are degraded only during primary MDS operation, the following processing is performed automatically.

- 1) Restarting the failed MDS
The failed MDS is restarted as the primary MDS.
- 2) Replaying the update log
The new primary MDS replays the **update log** to ensure the consistency of the file system.
- 3) Resuming processing after MDS switching
The AC makes a processing request to the new primary MDS.

14.1.4 Automatic recovery by standby MDS node

MDS normally operates on the predefined primary MDS node and secondary MDS node. By setting a standby MDS node (i.e., **MDS node** other than primary or secondary MDS node), however, the number of nodes on which MDS can operate can be increased.

If the primary MDS node or secondary MDS node, or both are not activated after the specified time, the standby MDS node activates MDS instead of the primary MDS node or secondary MDS node.

MDS operation is required for accessing a file system and increasing the number of MDS nodes is advantageous to ensure continued file system operation.



See

For details on the MDS node settings, see [Section 15.2.2 "Node layout."](#)

14.2 AC degeneration

When a GFS Shared File System is mounted, MDS holds information about the AC. When a node to which AC is operating is downed, MDS automatically cancels the AC information for the downed node and management of the file system continues for other nodes.

Chapter 15 File System Design

This chapter describes the main features of the operational design of the GFS Shared File System.

15.1 Mount and unmount opportunity of the file system at the time of system starting and ending

For a GFS shared file system, the setup for automatic mounting and unmounting during system startup and shutdown differs from ufs.

It is required to set no for the mount at boot parameter in /etc/vfstab for all GFS shared file systems. The option noauto controls the automatic mount during system startup.

When noauto option is not set in /etc/vfstab, the file system is automatically mounted during system startup. The automatic mount is performed by the script S81sfcsrm which is executed during level 2 processing.

When noauto is set as a mount option in /etc/vfstab, the file system is not automatically mounted during system startup. If the file system is later mounted, it will be automatically unmounted during system shutdown. The script K09sfcsrm unmounts file systems during system shutdown.



Note

For the GFS Shared File system to mount file systems during system startup, the GDS volumes for the GFS management partition and the file systems must be automatically activated during system startup.



See

For more information about GDS, see "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

15.2 System Design

This section describes the following items that must be considered in creating or using the GFS Shared File System:

- Effects of file system operation and system load
- Node layout
- Disk layout
- LAN selection
- **quota function**
- Future expansion

15.2.1 Effects of file system operation and system load

To maintain strict file system consistency, the GFS Shared File System processing sometimes significantly slows down during the following operations:

- When frequently creating or deleting a file or directory in the same directory from multiple nodes, and repeatedly monitoring directory updating with `readdir(3C)` or `stat(2)`.
- When frequently updating the same block in the same file from two or more nodes.
- When frequently referencing from other nodes a file that is frequently updated from one or more nodes.
- When an `ls(1)` command with the `-l` option and a command requiring attributes of files under the directory such as `cp(1)`, `du(1M)`, and `tar(1)` are both issued to a directory contains at least 10,000 files.

In the above cases, file system throughput may improve by changing the monitoring or access frequency or by dividing a target file.

CPU load or I/O load may be concentrated on the node where the **MDS** that manages the file system **meta-data** operates. A heavy load here indicates that operations accompanying updating of the file system meta-data, such as file creation, deletion, or extension are frequently being done. In such cases, file system throughput may improve by optimizing the layout of the **meta-data area** and **update log area**. An alternative is to increase CPU throughput of the node where the MDS operates.

15.2.2 Node layout

Up to eight nodes can simultaneously share one GFS file system, therefore, file system processing can continue even if the MDS node goes down. To set up for automatic recovery of the MDS, select two or more nodes on which MDS can operate, then set these nodes as **MDS nodes**. From the MDS nodes, select the primary MDS node (on which the usual primary MDS operates), the secondary MDS node (on which the usual secondary MDS operates, and all nodes as **standby MDS nodes** (backup MDS nodes if primary and secondary MDS nodes are unavailable).



See

For details on the effects of MDS node settings, see [Section 14.1, “MDS Failure Recovery.”](#)

As previously described in [Section 15.2.1, “Effects of file system operation and system load.”](#) the following should be taken into consideration:

- CPU and I/O loads accompanying updating of the meta-data of the file system that occur on the node where MDS operates
- CPU throughput, I/O throughput, memory capacity, and swap area



Note

MDS receives requests from all nodes sharing a file system and then updates the meta-data, therefore, install at least two CPUs on nodes on which MDS may operate.

To activate a node that accesses the GFS Shared File System, the node must have been set as an MDS node, or the primary MDS node, secondary MDS node, or standby MDS node (or all together) must be activated simultaneously, or activated in advance. To stop all nodes set as MDS nodes, all nodes that access the relevant file system must have been stopped.

By setting all nodes that share a file system as MDS nodes (i.e., primary **MDS node**, secondary MDS node or **standby MDS node**), each node can be activated or stopped regardless of the state of the other nodes. If both the primary and secondary MDS nodes are stopped during operation, the file system can still be used because standby MDS nodes will be activated as the primary MDS and secondary MDS nodes.



Note

When activating the primary or secondary MDS on a standby MDS node, activation of the primary MDS or secondary MDS may take some time.

15.2.3 Disk layout

The GFS Shared File System classifies the areas on disk media consisting of a file system into the following three areas:

- **meta-data area**
- **update log area**
- **data area**

The primary MDS references or updates two areas: the meta-data area and the update log area. The **AC** references or updates the data area.

The minimum disk configuration for the GFS Shared File System is one partition. However, the meta-data area and update log area can be separated and configured on a different partition. Also, the meta-data area and the data area can be split into multiple partitions.

In an environment where MDS I/O is expected to create a file system processing bottleneck, because of disk head positioning due to another process or shortened processing wait time is anticipated, throughput can be improved by separating the meta-data and update log areas on to independent volumes.

In an environment where data area I/O is expected to create a file system throughput bottleneck, throughput can be improved by distributing the data area in to multiple partitions on independent volumes.



See

For more information about available partition configurations, see [Section 13.1.6, "Partition configuration."](#)

15.2.4 Work Disk for fsck_sfcfs(1M)

When performing full check fsck of the GFS Shared File System, since the memory space of a fsck_sfcfs(1M) process may not be enough to manage intermediate data for full check fsck, work disks to manage intermediate data are needed. Prepare work disks for fsck in addition to the disk areas used as file systems, when the calculated size using the below-mentioned formula is 512 megabytes or more.

The work disks for fsck must be equal to or larger than the calculated size using the following formula from number of data partitions and file system size (gigabyte unit) of the GFS Shared File System.

$$40\text{MB} + [\text{number of data partitions}] \times 32\text{MB} + [\text{file system size (gigabyte unit)}] \times 65\text{MB}$$

A local disk or shared disk can be used as a work disk for fsck.

When using two or more file systems as the GFS Shared File System, calculate the required size for full check fsck of each file system using the above-mentioned formula. If full check fsck of two or more file systems are performed simultaneously, prepare work disks for fsck equal to or larger than calculated size for each file system and then specify different work disks to perform full check fsck of two or more file systems. If full check fsck of two or more file systems are performed one by one, prepare work disks for fsck equal to or larger than the maximum of calculated sizes and then specify the same work disks to perform full check fsck of two or more file systems. For example, when there are two file systems and both of them require 32 gigabytes work disk for fsck, two 32 gigabytes work disks for fsck are enough to perform full check fsck of two file systems simultaneously, only one 32 gigabytes work disk for fsck is enough to perform full check fsck one by one.

When full check fsck is performed, intermediate data is written on the work disks for fsck and the existing data on the work disks is changed. Specify an partition which is not used for other file systems or application programs as a work disk for fsck.



See

For details on full check fsck, see [Section 21.2, "Checking a File System for Consistency and Repairing It."](#)

15.2.5 LAN selection

In the GFS Shared File System, the **MDS** communications needed to obtain **meta-data** and to maintain consistency are processed through the LAN.

The communication path is specified by a host name. The **GDS** LAN can be used as a communication path.

Set up the LAN path while keeping the following in mind.

- LAN traffic volume and LAN load
- Redundant Configuration Definition for a LAN fault

The GFS Shared File System is designed to not increase the LAN load. But, the load can increase under certain operational conditions. When a higher LAN load is experienced due to other processing being performed, the response time of the access processing to the GFS Shared File System may be delayed. The LAN load status can be found by executing `netstat(1M)`.

The use of the following is recommended when the load on a file system is large or when the response time is more important than other factors.

- High-speed LAN
- Private LAN

In the GFS Shared File System, specific nodes block the file systems if an error occurs in the communications between MDS and node AC. This is done because AC determines file system processing cannot be continued. The followings are suggestions for avoiding problems in the communication paths.

- To establish a multiplex communication path by integrating several LAN paths into one local path using **GLS**.
- To set several communication paths by using the GFS Shared File System's multiple LAN support function.

The GFS Shared File System's multiple LAN support function realizes communication path redundancy in a file system. And, by using GLS, communication path redundancy including applications can be achieved.

To recover or to change the settings after a LAN fault occurs in an environment in which the GFS Shared File System's multiple LAN support function is used, see [Section 21.7 "Multiple LAN Setup"](#).

15.2.6 quota function

To use the quota function, the GFS Shared File System uses management data called the external quotas file and internal quotas file. During creation of the file system with `sfnewfs(1M)`, the required minimum area for the management data is allocated.

1. Structure of the quotas file

The quota information uses a 64-byte area for each user. The quota information of the user in the file is stored in the offset obtained by multiplying an UID value by 64 bytes.



Note

If possible, the UIDs of users should be consecutive values. If the values are not consecutive, the quota information will contain unused areas. In addition, the quota data may not be a cache hit if accessed and would require real input-output that may reduce system performance.

2. quotas file data storage area

The area of the internal quotas file is allocated from the **V-data area**. The area of the external quotas file is allocated from the data area.

The unit of acquisition from the V data area is 1 kilobytes; that from the file data area is 8 kilobytes.

Thus, if UID is limited to continuous users from X to Y, the V data area quantity used for internal quotas can be estimated as follows:

$$\text{Use of disk space} = (\text{INT}(Y/16) - \text{INT}(X/16) + 1) * 1\text{KB} + 1\text{KB}$$

(case of $X > 16$)

Use the following formula to estimate the file data area quantity needed in an external quotas file:

$$\text{Use of disk space} = (\text{INT}(Y/128) - \text{INT}(X/128) + 1) * 8\text{KB} + 8\text{KB}$$

(case of $X > 128$)

The following shows an example in which UID takes continuous values from 1000 to 2999.

<Internal quotas file>

Amount of user information in the 1 kilobytes blocks is $1024 / 64 = 16$.

If the value changes from 1000 to 2999 continuously:

$$1000 / 16 = 62.5$$

$$2999 / 16 = 187.4375$$

As shown above, 126 blocks from block 62 to block 187 are used.

Thus,

$$126 * 1\text{KB} + 1\text{KB}(\text{for the root}) = 127\text{KB}$$

an area of 127 kilobytes is needed.

<external quotas file>

Amount of user information in the 8 kilobytes blocks is $8192 / 64 = 128$.

If the value changes from 1000 to 2999 continuously:

$$1000 / 128 = 7.8125$$

$$2999 / 128 = 23.4...$$

As shown above, 17 blocks from block 7 to block 23 are used.

Thus,

$$17 * 8\text{KB} + 8\text{KB}(\text{for the root}) = 144\text{KB}$$

an area of 144 kilobytes is needed.



Note

If the quota function is used in an existing the GFS Shared File System, check for sufficient free area in the V-data area. If the **quota function** has been enabled, the number of usage files and number of user blocks is managed using the internal quotas file, even for users for which limit values have not been specified.



See

For details on the quota function, see [Section 21.4 "quota function."](#)

15.2.7 For future expansion

When the GFS Shared File System is created with `sfcnewfs(1M)`, the required minimum area is allocated according to the default settings. At file system creation, check that at least the parameter values listed in the table below are sufficient for future expansion. For example, the partition addition function cannot add partitions exceeding the `-l maxvol` parameter of `sfcnewfs`. In this case, to expand the file system, the current data must be backed up and then deleted, a new file system must be created, and the data must be restored.

As needed for future file system expansion, set larger than the default values for options, such as maximum number of partitions, to create a file system with sufficient disk space allocation.

sfcnewfs parameter	Default	Meaning
<code>maxdsz(-d)</code>	File data area size at specification	Maximum file data area size
<code>maxnode(-n)</code>	16	Maximum number of sharing nodes
<code>maxvol(-l)</code>	16	Maximum number of partitions

15.3 Backup Design

This section describes the hardware required to restore backed up file systems and the methods used to perform backups.

- Required hardware
 - A tape unit or hard disk unit is used to backup the GFS Shared File System.
 - A disk array equipment of our company such as ETERNUS3000/GR720/GR730/GR740 is used to perform high-speed backup in coordination with GDS through sfcproxyop (1M) and sfcproxyrestore (1M).

- Backup restoration procedure
 - The GFS Shared File System restores backed up file systems with the following methods:
 - Restoring the entire file system: dd(1M) or sfcmbkbackup(1M)
 - Restoring the file system on a file basis: sfccpio(1)
 - Restoring the file system on a file basis: sfcdump(1M) or sfcrestore(1M)
 - High-speed backup restoration of the file system unit by cooperation with GDS which used sfcproxyop (1M) and sfcproxyrestore (1M)

If the GFS Shared File System consists of multiple partitions, backups should be restored on a **partition** basis. By distributing restoration of a partition backup to multiple nodes, backup restoration can be performed in parallel, reducing the time required.



Note

If tar(1) or cpio(1) used, **extent** attribute does not copy.



See

For more information about restoring backups in the GFS Shared File System, see [Chapter 22 "File System Backing-up and Restoring."](#)

Chapter 16 Management Partition

This chapter explains the management partition.

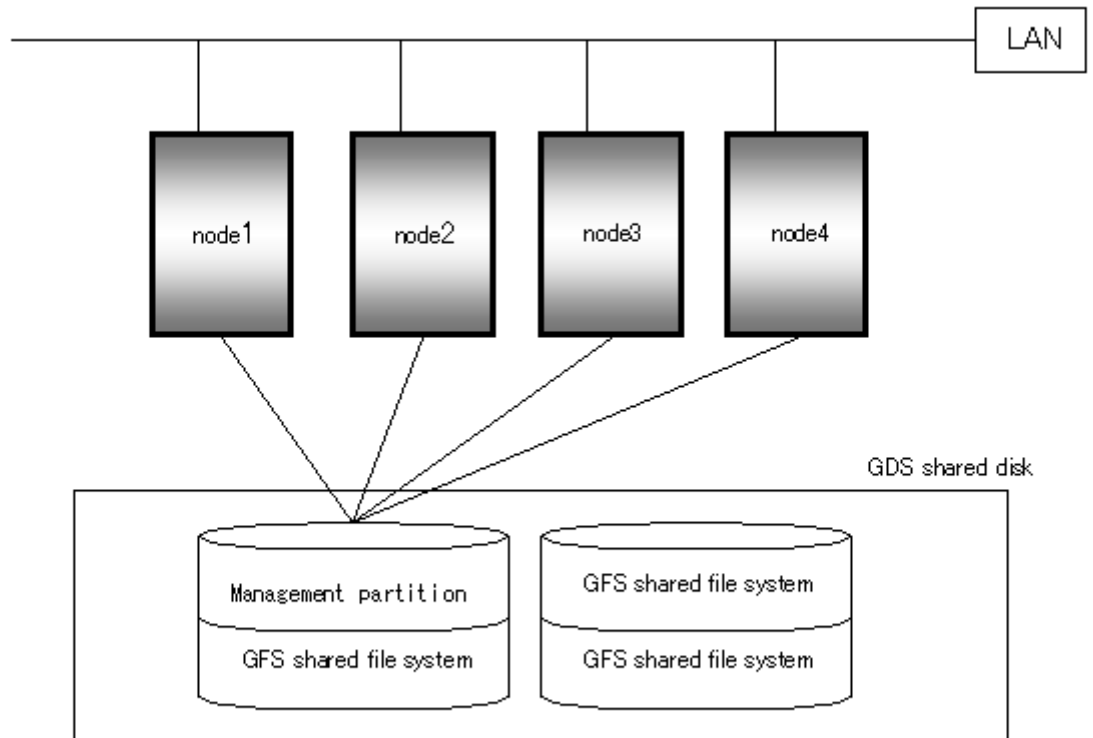
16.1 Management Partition

The management partition is used to administer the information that is necessary for the operation of the GFS Shared File System. The management partition is also used for the file system monitoring mechanism to maintain the consistency of information. The management partition contains the following information:

1. Information for each of the nodes constituting the GFS Shared File System.
2. Information linking to a shared node and a shared device in each GFS Shared File System
3. MDS allocation information for each GFS Shared File System

The node information in 1 must be set up using the `sfcsetup(1M)` command. The information in 2 and 3 can be set up or changed with the `sfcnewfs(1M)`, `mkfs_sfcfs(1M)`, or `sfcadm(1M)` commands. Also, use the `sfcgetconf(1M)` command to backup the information within the management partition.

Figure Configuration of the connection between nodes and the management partition



16.2 Resources Required for the Management Partition

The GFS Shared File System requires, in addition to the volumes that constitute the file system, one **GDS** Shared volume that is exclusively for the management partition for each cluster system. Even when using two or more file systems, the number of required management partitions is one.

The management partition uses disk space of forty megabytes. So, prepare a volume whose capacity is forty megabytes or more.



See

For details on how to set up the GDS Shared disk system, see the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."



Note

The GDS volume which comprises the management partition should belong to a disk class of type "shared" and all the nodes that share the GFS Shared File System must be specified in the **scope** of this disk class. Also, the GDS volume must be set up to automatically activate when the node is started.

16.3 Notes on Management

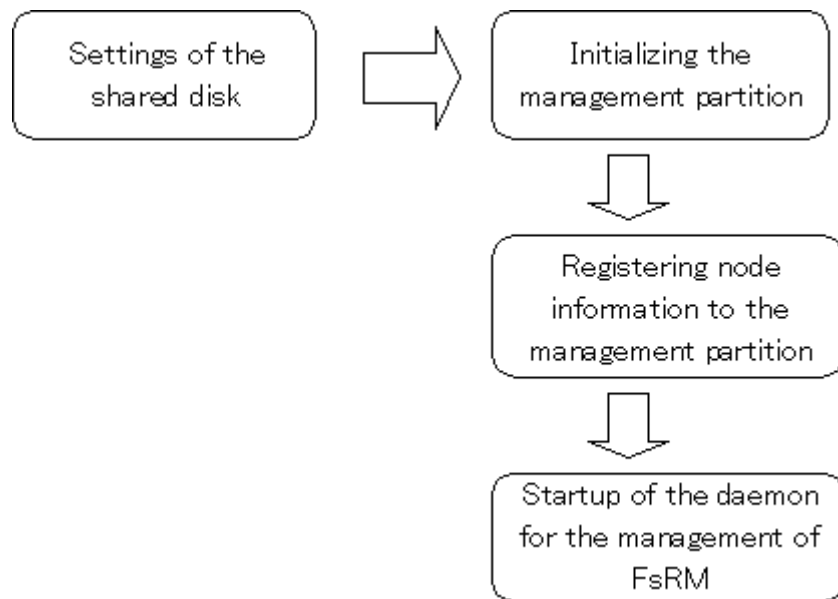
If a failure occurs in the management partition, the GFS Shared File System within a cluster system stops. To protect the management partitions, Fujitsu recommends mirroring of the GDS volumes used for management partitions.

Also, prior to changing the configuration of the file system, make a backup of the management partition information by executing the `sfcgetconf(1M)` command.

16.4 Setting up the Management Partition to Start the file system monitoring mechanism

The following sections explain the initializing of the management partition, adding or deleting node information, and starting/stopping the management partition.

Figure Setting up the management partition to start the file system monitoring mechanism



1. Initializing the management partition

Use the `sfcsetup(1M)` command to initialize the management partition.

The initialization of the management partition is explained below.



See

For details on `sfcsetup(1M)`, see [Appendix F.2.27 "sfcsetup\(1M\)."](#)



Note

To initialize the management partition, the file system monitoring mechanism must have already been stopped on the cluster system.

Specify the GDS Shared volume as the management partition. Note that a volume of at least 40 MB is necessary.

```
# sfcsetup -c /dev/sfdsk/gfs/rdsk/control
```

When a partition that is already initialized as the management partition is to be initialized again, execute the command with the `-f` option specified, as shown below:

```
# sfcsetup -c -f /dev/sfdsk/gfs/rdsk/control
```

2. Registering node information

sfcsetup(1M) command registers node information in the management partition.
 An example of registering node information in the management information is shown below.



Note

Register the node information on all the nodes on which the file system monitoring mechanism is to be executed.

- Registering a node in the management partition

```
# sfcsetup -a /dev/sfdsk/gfs/rdsk/control
```

- The path name of the management partition that has been set up can be confirmed by executing the sfcsetup(1M) command with the **-p** option specified.

```
# sfcsetup -p
/dev/sfdsk/gfs/rdsk/control
```

- Registered node information can be confirmed by executing sfcsetup(1M) with no option specified.

```
# sfcsetup
HOSTID          CIPNAME          MP_PATH
8038xxx         sunnyRMS         yes
8038yyy         moonyRMS         yes
```

3. Node deletion

sfcsetup(1M) command deletes node information from the management partition.
 The procedure for deleting node information from the management partition is explained below.



Note

On the node that is to be deleted, the operation of the file system monitoring mechanism must have been stopped. Or, the operation of the file system monitoring mechanism for the entire cluster system must have been stopped.

On the node that is to be deleted, execute the command as shown below:

```
# sfcsetup -d
```

4. Starting the file system monitoring mechanism

Use the sfcfrmstart(1M) command to start the file system monitoring mechanism.



See

For details on sfcfrmstart(1M), see [Appendix F.2.7 "sfcfrmstart\(1M\)."](#)



Note

To start the file system monitoring mechanism, node information must have been registered in the management partition.

Execute the command as follows on the node on which the file system monitoring mechanism is to be started.

```
# sfcfrmstart
```

5. Stopping the file system monitoring mechanism

Use `sfcrmstop(1M)` to stop the file system monitoring mechanism.



See

For details on `sfcrmstop(1M)`, see [Appendix F.2.8 "sfcrmstop\(1M\)."](#)

Execute the command as follows on the node on which the file system monitoring mechanism is to be stopped.

```
# sfcrmstop
```

16.5 Backup and restore of the management partition information

The following sections explain backup and restore of the management partition.

1. Backup of the management partition information

Use the `sfcgetconf(1M)` command to make a backup of the management partition information.

```
# sfcgetconf _backup_file_
```



See

For details on `sfcgetconf(1M)`, see [Appendix F.2.9 "sfcgetconf\(1M\)."](#)



Note

Make a backup of the management partition information, after changing the configuration of the file system by `sfcnewfs(1M)`, `mkfs_sfdfs(1M)`, `sfcadd(1M)`, `sfcadm(1M)`, `sfchost(1M)`, `sfcnode(1M)`, `sfcquotaon(1M)`, `sfcquotaoff(1M)`, etc.

`sfcgetconf(1M)` generate a shell script `_backup_file_`:

```
# cat _backup_file_
#!/bin/sh
# This file is made by:
#   sfcgetconf _backup_file_
# Tue Jun 18 09:08:06 2002

#---- fsid : 1 ----
# MDS primary   (port) : host1 (sfdfs-1)
# MDS secondary (port) : host2 (sfdfs-1)
# MDS other      :
# AC           : host1,host2
# options      :
# device       : /dev/sfdsk/c1/rdsk/v1
sfcadm -m host1,host2 -g host1,hostw -p sfdfs-1,sfdfs-1 /dev/sfdsk/c1/rdsk/v1
...
```



Note

The file system monitoring mechanism must be active on the cluster system for `sfcgetconf(1M)` to be able to backup the management partition information. If file system monitoring mechanism is not active, start it by using `sfcfmstart(1M)`.

2. Setting up of a new management partition and starting the file system monitoring mechanism

If a management partition is unusable, set up a new management partition on a GDS volume and start the file system monitoring mechanism.



For the details of management partition setting and the file system monitoring mechanism starting, refer to ["16.4 Setting up the Management Partition to Start the file system monitoring mechanism."](#)

3. Restoration of the management partition information

The shell script generated by `sfcgetconf(1M)` in "[1. Backup of management partition information](#)" is used for the restoration of management partition information.

The method of restoration of the management partition information is explained below.

```
# sh _backup_file_  
get other node information start ... end
```

Chapter 17 Starting and Exiting the Management View

The **GFS Management View** operates under a Web environment.

A web environment provides access via a browser from any location.

This chapter explains how to start and exit the GFS Management view.

For information about the following items, all of, which are necessary for using the GFS Management view, see the "*PRIMECLUSTER Web-Based Admin View Operation Guide*."

- Topology
- Preparation
- Operating Screen

17.1 Starting Web-Based Admin View

When all preparations are complete, start **Web-Based Admin View** using the following procedure.

1. Start the browser.
2. Specify the URL in the following format:

```
http://host-name:port-number/Plugin.cgi
```

Host name: The host name or IP address of the primary or secondary management server
Port number: **8081**

3. After starting the Web-Based Admin View, user input screen appears. Type the user name and password for the management server, and press <OK>. When certification terminates, the top menu of Web-Based Admin View appears.



Note

Usable client environment

For details about the environment in which the browser is started, refer to the “*PRIMECLUSTER Web-Based Admin View Operation Guide.*”



Note

When the Screen Stops Responding

With some browsers, the screen may stop responding.

- If the browser disappeared, restart the browser and reconnect to the URL of the management server.
- If the browser does not work (the browser fails to respond), reload the URL or restart the browser.
- If the screen turns black immediately after starting Web-Based Admin View, the process might resume if you move the mouse on the browser.



Note

Port Number

The default value for the port number is “**8081.**”
For information on how to change the port number, see “*PRIMECLUSTER Web-Based Admin View Operation Guide.*”

17.2 Web-Based Admin View Top Menu

After **Web-Based Admin View** has started, the “Web-Based Admin View operation screen” appears. Web-Based Admin View operates and manages GFS from a web browser. Top menu refers to the Web-Based Admin View operation menu.

17.2.1 Web-Based Admin View Operation Menu Functions

The Web-Based Admin View operation menu supports the following functions related with the GFS Shared File System:

Table Web-Based Admin View Operation Menu Description

Menu	Description
Global File Services	Set and operates GFS Shared File System.



Note

Note that the contents of the Web-Based Admin View operation menu varies depending on the installed products.

- When a dialog is displayed because of an error in Web-Based Admin View, the picture of the lamp near the top on the right side of the Web-Based Admin View top menu also turns red. If other screens are hiding the dialog box, clicking the red lamp brings the dialog box to the front. To monitor for problems always display the lamp on the Web-Based Admin View top menu.

17.2.2 Web-Based Admin View Tool Menu Functions

For information about the Web-Based Admin View Tool Menu, see “*PRIMECLUSTER Web-Based Admin View Operation Guide*.”



Note

In the “*PRIMECLUSTER Web-Based Admin View Operation Guide*,” replace the term “Cluster Management view” with “GFS Management view.”

17.3 Starting GFS Management View

Select the <Global File Services> Menu on the Web-Based Admin View operation menu, the GFS Management screen (hereafter referred to as the main screen) opens.

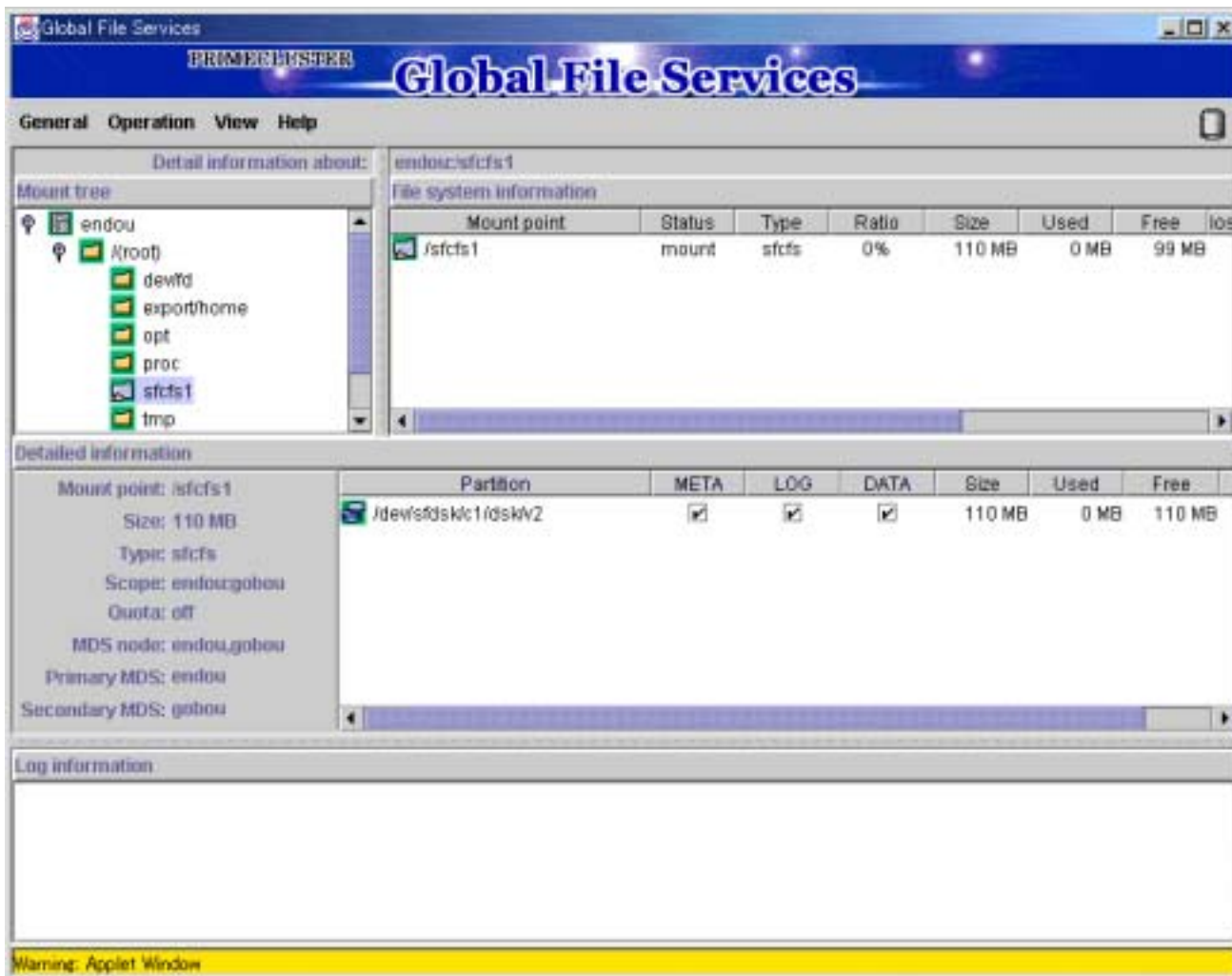
From the main screen, you can create a file system, change a file system's configuration, and delete a file system.



See

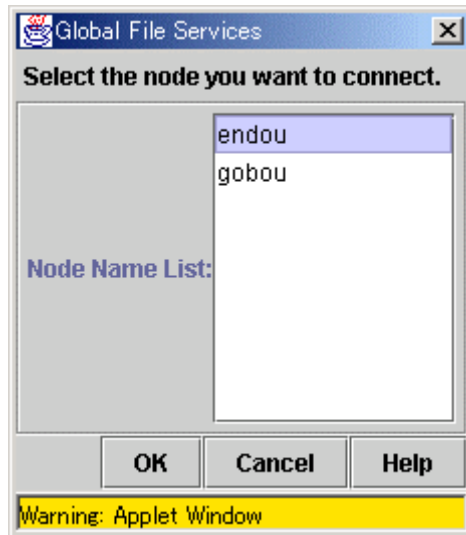
For more information, see [Chapter 19, "File System Operations \(GUI\)."](#)

Figure GFS Management Main Screen



If there is more than one node, [Select Node] Screen appears on top of the main screen. Select the desired node. The selected node can be changed from the main screen.

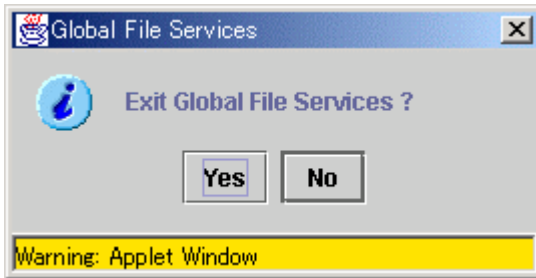
Figure Select Node Screen



17.4 Exiting GFS Management View

To exit the **GFS Management View**, click [Exit] on the [General] menu. The following message will appear.

Figure Exit Screen



If you press <Yes>, the Web-Based Admin View Screen (top menu) is displayed.
If you press <No>, the main screen is displayed again.

17.5 Exiting Web-Based Admin View

Exit Web-Based Admin View, as follows:

- Select the [Logout] button on the top menu.
- When the login screen appears, exit the browser or use the [Back] feature of the browser to exit Web-Based Admin View.



Note

If the login screen continues to be displayed

The login screen will remain displayed briefly after the browser exits. No additional action is required to close the login screen.

Chapter 18 Operation Management View Screen Elements

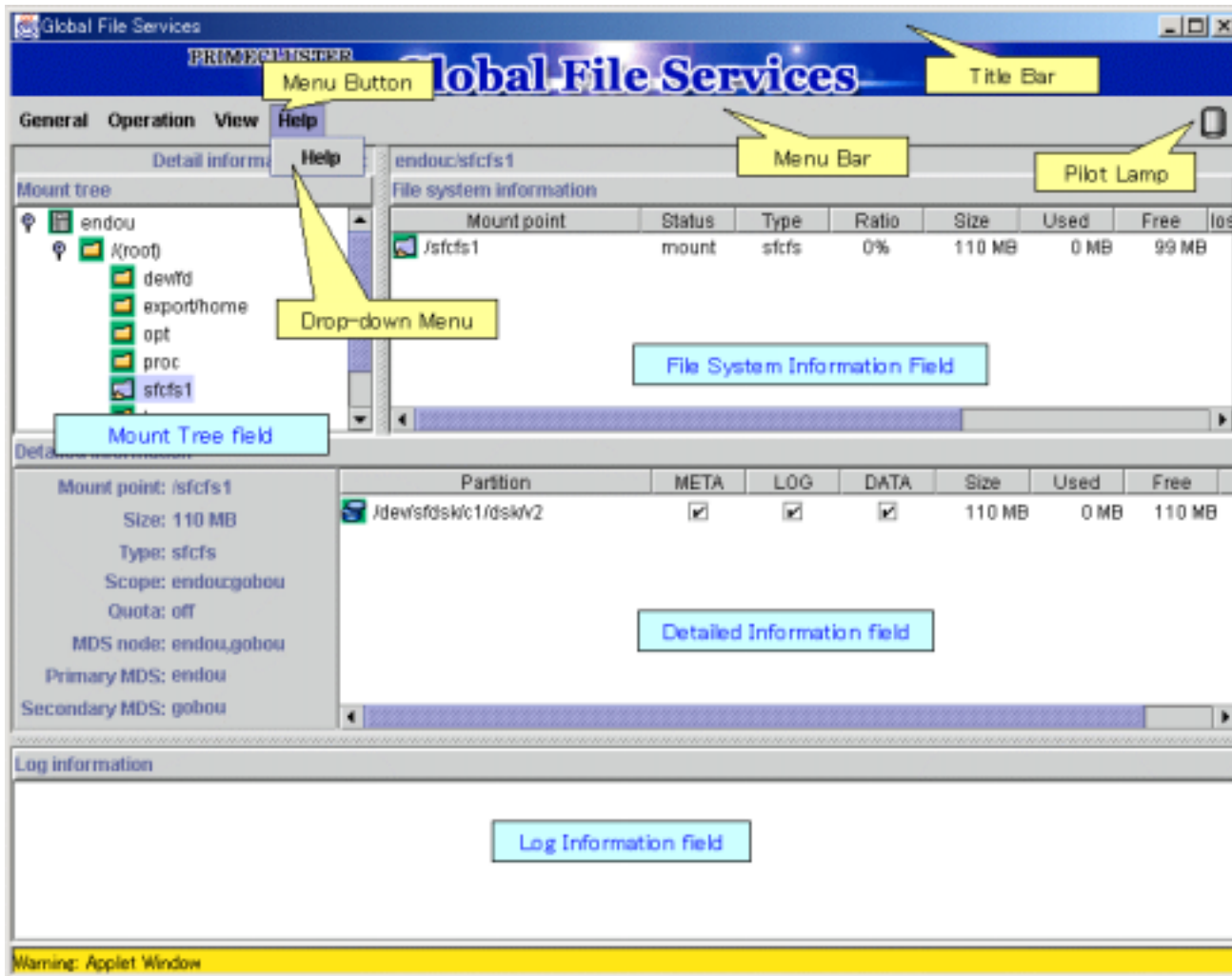
This chapter describes the screen elements of the **GFS Management View**.

18.1 Screen Configuration

Main screen

Select Global File Services from **Web-Based Admin View**, and the screen below appears. From this main screen, you can create a file system, change a file system's configuration, and delete a file system. Screen configuration of the main screen is shown below.

Figure GFS Management Screen (main screen)



Mount tree field

File systems that can be referenced by the selected node are displayed in a tree-structure. Only files systems that are listed in /etc/vfstab can be displayed. When the GFS Management appears, the [Select Node] screen appears first to select target node. To change the selected node, click [Select Node] on the [General] drop-down menu.

File System Information field

File system information is displayed for the node/directory selected from the Mount Tree field. If the file

system's status is unmount, 0 is displayed for the size.

Each object has an icon representing the object type. If the file system type is sfcs, icons are color-coded so that the status of the objects can be determined at a glance.



For an explanation of icon types and object status, see [Section 18.3, "Icon Types and Object Status"](#) in this manual.

Detailed Information field

When an sfcs file system is selected in the Mount Tree field or File System Information field, detailed information including resource name, registered cluster service name, and partition information is displayed.

If the file system status is unmounted, 0 is displayed for the size..

The size of the data area is displayed in the size column for each partition with a data area. The size column is not displayed for partitions without a data area.

Log Information field

Messages concerning the GFS Shared File System daemon program are displayed. Each message is preceded by the name of the node which generated the message.

Title Bar

Displays screen title (Global File Services).

Menu Bar

Displays the menu buttons.

Menu Button

Allows control of the objects selected on screen.

There are <General>, <Operation>, <View>, and <Help>.

Drop-down Menu

When a menu button from the Menu Bar is selected, a drop-down menu appears.








For details on the drop-down menu, see [Section 18.2, "Menu Configurations and Functions."](#)

Pilot Lamp

Shows the status of monitored objects.

The lamp can indicate the following status.

Table Types of Pilot Lamp

Pilot Lamp	Status	Meaning
 (Gray, lit up)	Normal	-
 (Red, blinking)	Abnormal	The file system is abnormal. (Unavailable)
 (Red, lit up)	Abnormal	A red blinking warning lamp was single-clicked.
 (Yellow, blinking)	Alarm	The utilization rate of the file system exceeds the threshold or the file system is abnormal at other node.
 (Yellow, lit up)	Alarm	A yellow blinking warning lamp was single-clicked.



Note

When a field cannot be fully displayed, move the mouse cursor to that part that is not fully displayed. A popup display will appear.

In a scope popup, information on the nodes sharing a file system is displayed in node (host) name format. When several host names are specified, they are displayed in parentheses.

18.2 Menu Configurations and Functions

Each menu button has a Drop-down Menu; you can operate the selected object on screen. This section explains the menu configuration and functions.

18.2.1 General

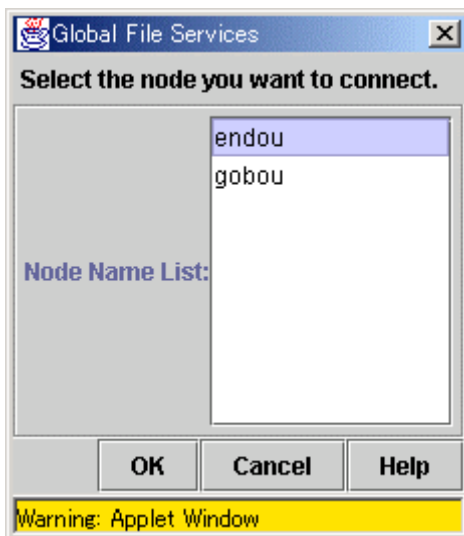
Figure General menu



Select Node

Select the target node and click <OK> button. Only one node can be selected at a time. To close without changing the selected node, click <Cancel> button.

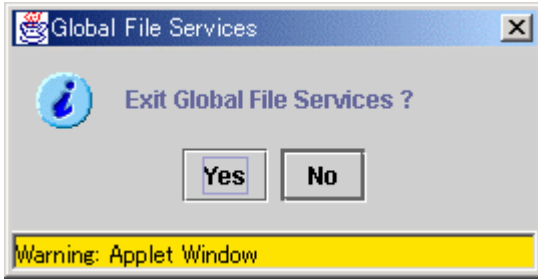
Figure [General]: [Select Node] Screen



Exit

Exits Global File Services.

Figure [General]: [Exit] Screen



18.2.2 Operation

Figure Operation menu



Create

Creates a file system.



See

For details, see [Section 19.2.1, "Creating a file system."](#)

Figure [Operation]: [Create] Screen

Global File Services

Create File System Wizard (1)
Set the node name and the priority that will share the file system,
and mount point.

	Node name	Host name		Primary MDS	Secondary MDS	MDS
<input checked="" type="checkbox"/>	endou	endou	Select	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	gobou	gobou	Select	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

MDS node: all nodes 2 nodes

Mount point:

Make directory: yes no

Cancel Reset Next Help

Warning: Applet Window

Change Configuration

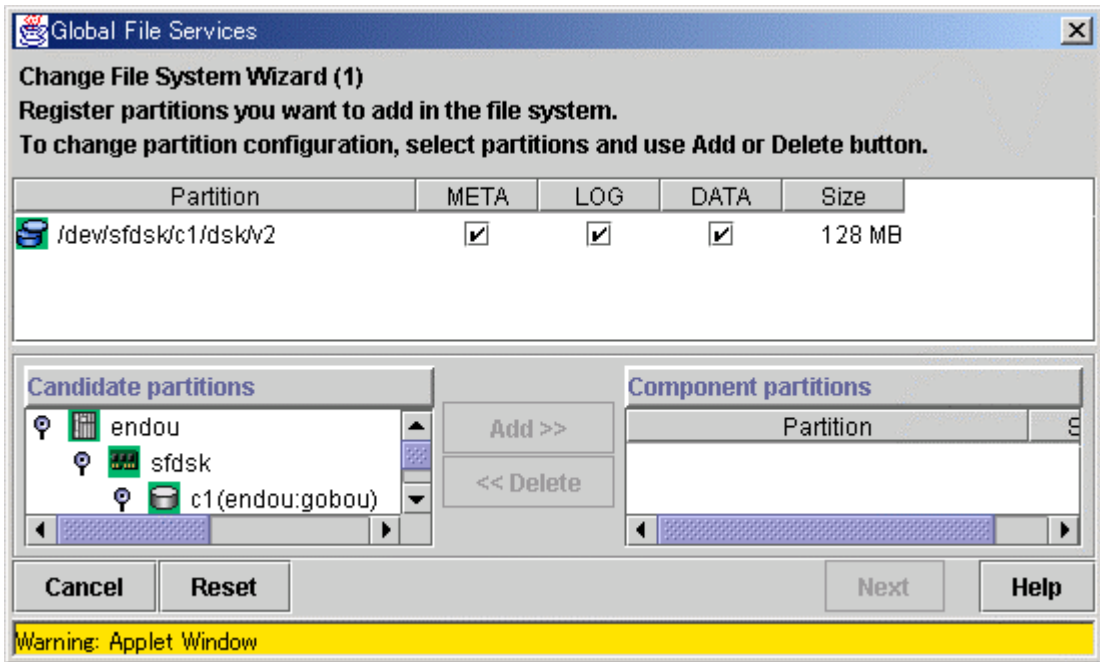
Changes the partition configuration of a file system.



See

For details, see [Section 19.3.2, "Changing the file system configuration \(partition addition\)."](#)

Figure [Operation]: [Change Configuration] Screen



Delete

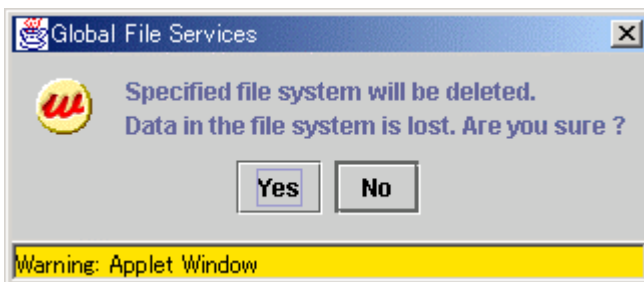
Deletes a file system.



See

For details, see [Section 19.4.1, "Deleting the file systems."](#)

Figure [Operation]: [Delete] Screen



Change Attributes

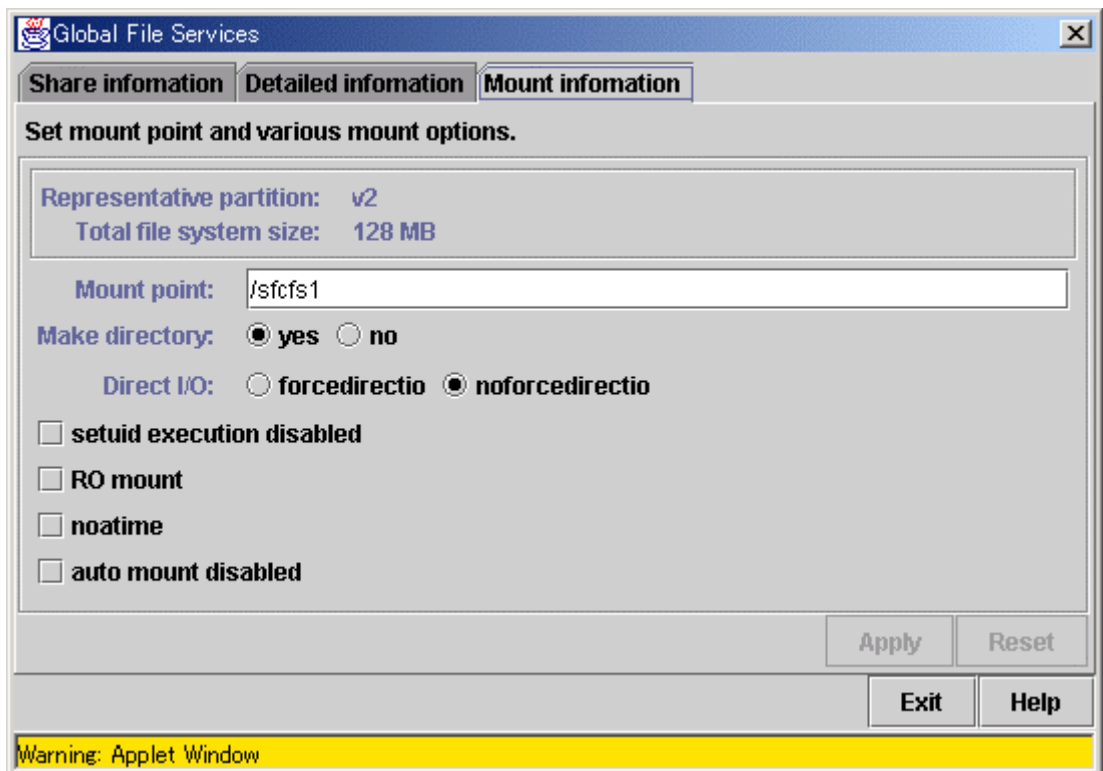
Changes the mount information, share information, and detailed information for a file system.



See

For details, see [Section 19.3.3, "Changing the sharing-node information,"](#) and [Section 19.3.1, "Changing in file system attributes."](#)

Figure [Operation]: [Change Attributes] Screen



18.2.3 View

Figure View menu



Abnormal Only

Only displays file systems with abnormalities.

Update Now

Obtain and display the latest file system information.

18.2.4 Help

Figure Help menu




Help

Displays help information.


18.3 Icon Types and Object Status

GFS Management View uses icons to show object types and status. The status and icons are shown below.






1. Node

Icon	Status	Meaning
	-	-



2. Adapter

Icon	Status	Meaning
	-	-


3. File system (sfcs)

Icon	Status	Meaning
 (Green)	Normal	The file system is working normally on all nodes.
 (Yellow)	Alarm	The usage rate of the file system exceeds the threshold or an error occurred at another node.
 (Red)	Abnormal	An error occurred in a node of the file system.
 (Blue)	In transition	The file system is processing a mount or unmount.
 (Light brown)	Inactive	The file system is unmounted.


4. File system (sfxfs, ufs)

Icon	Status	Meaning
 (Green)	Normal	-
 (Red)	Abnormal	-

5. Physical Disk

Icon	Status	Meaning
	-	-

6. Partition

Icon	Status	Meaning
	-	-



Note

File system status

A GFS shared (file system type: sfcfs) file system displays normal status until it is accessed and then the abnormality is detected. After an abnormality is detected, the icon shows abnormal status even if the file system is unmounted.

If file system type is ufs or sfxfs(GFS Local File System), the icon will not show abnormal status for an unmounted file system.

Chapter 19 File System Operations (GUI)

This chapter explains the operations of the GFS Management View.



Note

Before creating a file system, the management partition must be set up.



See

For details of the management partition, see [Section 16 "Management Partition."](#)

19.1 Flow of Operations

This section explains the flow of file system operations of the **GFS Management View**.

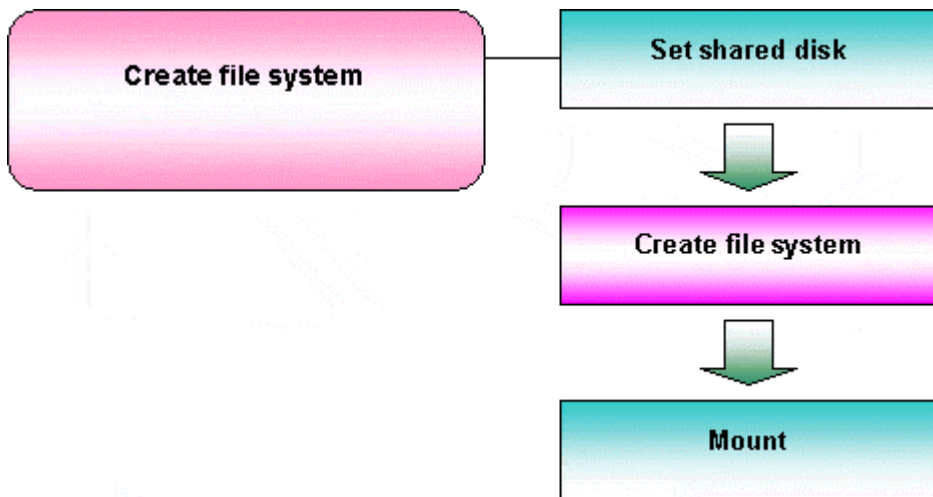
19.1.1 Creation

The following figure describes the flow of operations for creating a file system.



For details on operations in the following figure, see [Section 19.2 "Creation."](#)

Figure Operation flow for creating a file system operation



19.1.2 Change (in quota function status)

The figure below describes the flow of operations for changing quota function status. The quota function status can be changed for mounted or unmounted file systems.



See

For details on operations in the following figure, see [Section 19.3 “Change.”](#)

Figure Operation flow for changing quota function status (unmount file system)

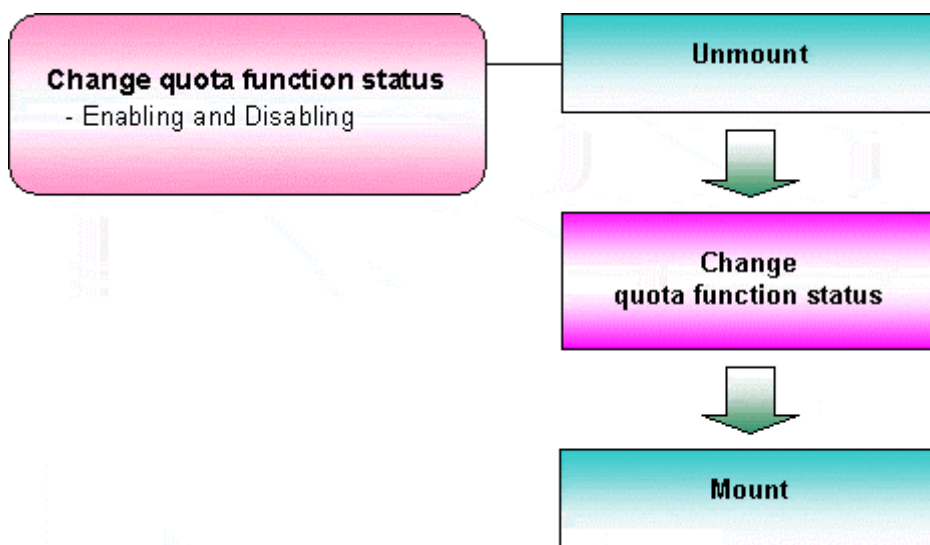
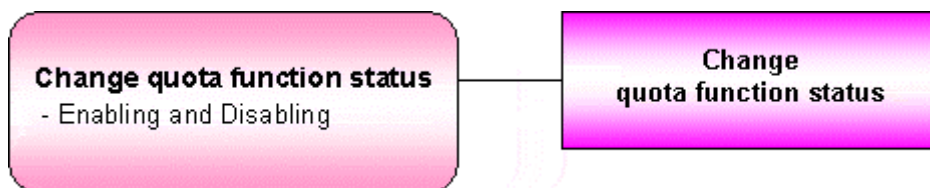


Figure Operation flow for changing quota function status (mounted file system)



Note

Disabling the quota function

If a file system is updated while the quota function is disabled and then the quota function is enabled, the matching of quota information is processed internally. While this quota information is being matched, limitation values set by the quota function are not enabled. Since the match processing of quota information takes time in proportion to the number of files involved, the quota function should not be disabled once it is enabled.

19.1.3 Change (in file system attributes)

The following describes the flow of operations for changing file system attributes. Using the **GFS Management View**, the following attributes can be changed:

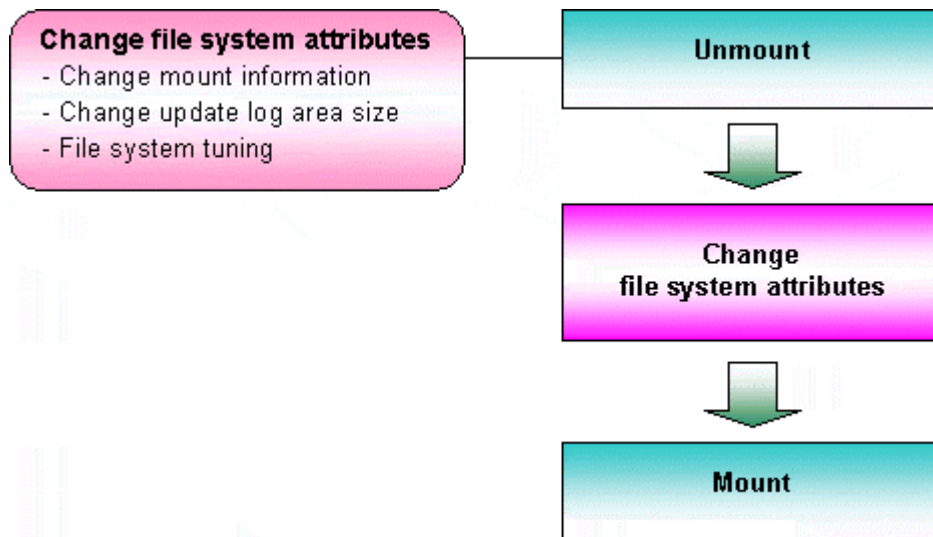
- Mount information
- Update log area size
- File system tuning



See

For details of the above changes and each operation in the flow described in the following figure, see [Section 19.3 "Change."](#)

Figure Operation flow for changing file system attributes



19.1.4 Change (partition configuration)

The following describes the flow of operations to change the partition configuration of a file system. Using the GFS Management View, the following configuration changes can be made:

- Addition of a file data partition
- Addition of a meta-data partition

A file data partition can be added to a mounted or unmounted file system.



See

For details on each operation in the flow described in the following figure, see [Section 19.3 "Change."](#)

Figure Operation flow for changing partition configuration (unmounted file system)

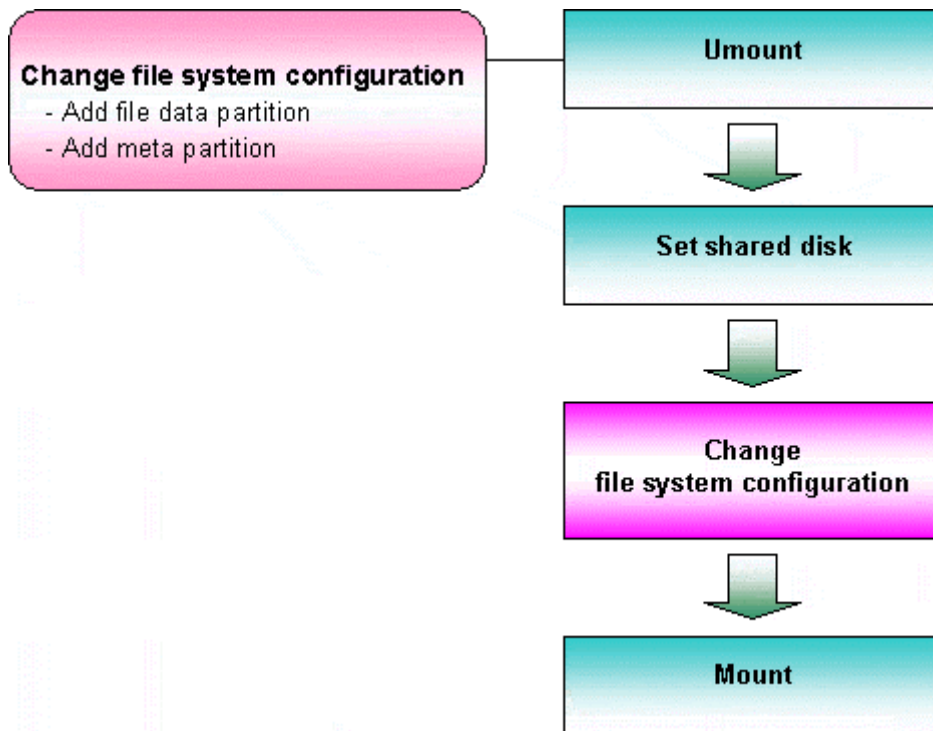
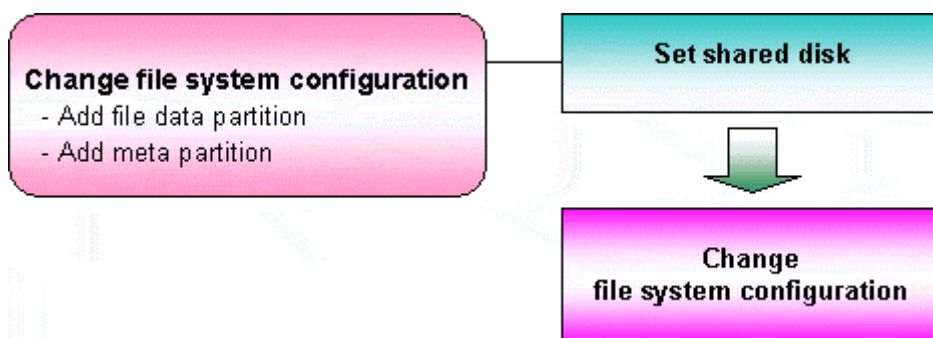


Figure Operation flow for changing partition configuration (mounted file system).



19.1.5 Change (shared node information)

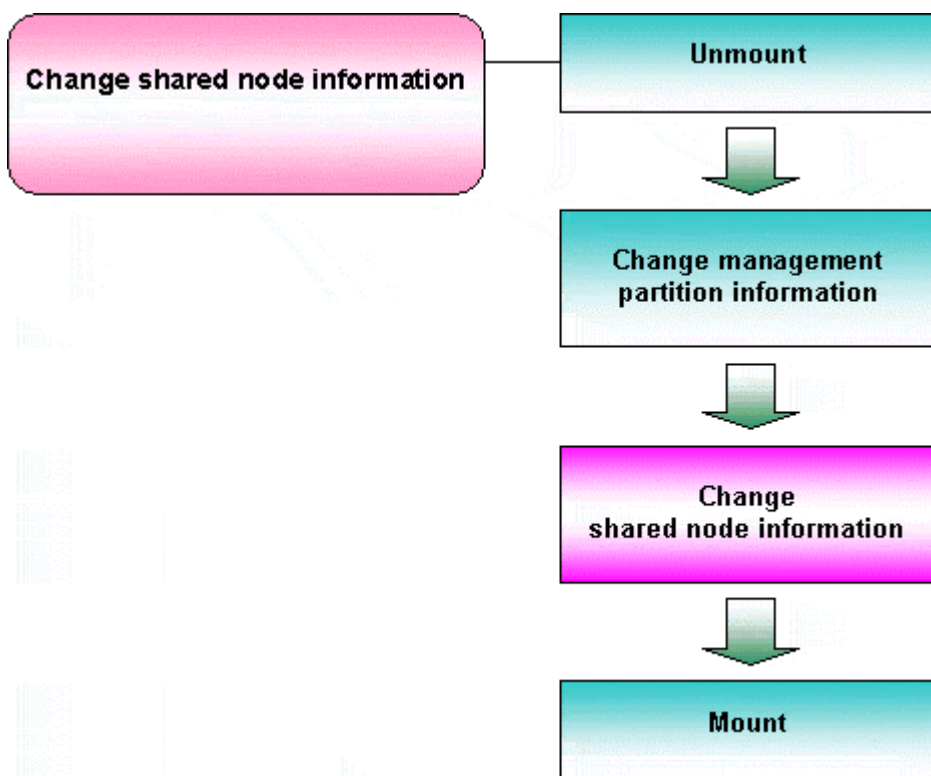
The following describes the flow of operations to change information about nodes sharing a file system. Information about nodes sharing a file system can be changed only when the file system is unmounted on all nodes.



See

For details on each operation in the flow described in the following figure, see [Section 19.3 "Change."](#)

Figure Operation flow for changing shared node information



19.1.6 Deletion

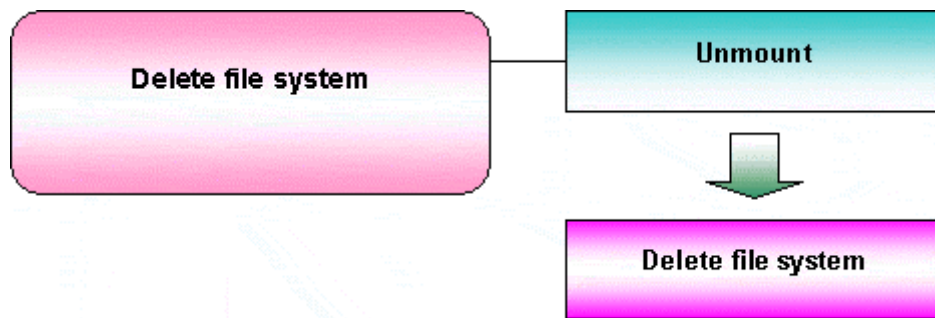
The following describes the flow of operations to delete a file system. A file system can be deleted only when the file system is unmounted on all nodes.



See

For details on each operation in the flow described in the following figure, see [Section 19.4 "Deletion."](#)

Figure Operation flow for deleting a file system



19.2 Creation

This section explains how to create a file system.

It is easy to create a file system because the file system creation operation uses a wizard.

To start the file system creation wizard, Click [Create] on the [Operation] drop-down menu.



Note

Note of Operation

- If the assigned values are invalid, the <Back> button may be clicked to display the previous screen.
- The information for the Management View must be updated before the changes are reflected. To immediately update the information, on the main screen click [Update Now] on the [View] drop-down menu.



Note

When a file system is created using the GUI, it may affect the file system access performance because the initialization processing of a **meta-data area** is done after the **MDS** has been started. Create the file system by issuing the `scnewfs(1M)` command if you want to complete the initialization processing of the meta-data area when the file system is created. The initialization processing of the meta-data area is completed by specifying the `-c` option for the `scnewfs(1M)` command.

19.2.1 Creating a file system

The following describes the procedure for creating a file system.

(1) Setting a shared disk device

The partition in which a GFS Shared file system is created and operated must be on a **shared disk** recognized by **PRIMECLUSTER**. To specify a GDS logical volume as the target device, it must be active.



Note

Check that the disk is set up as shared for simultaneous access.

If a partition has set up the disk as shared by automatic detection of shared disks, it is already set as a shared disk for simultaneous access and no further action is needed.

When registering a GDS logical volume with a class, set the disk type as shared and the scope to include all the nodes that share the GFS Shared File System.



See

For GDS logical volume operations, see the applicable items under "Operation using Global Disk Services Management View" in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

For the logical volume operations of GDS, see the applicable items under "Operation using Global Disk Services Management View" in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

(2) Setting share information

Click [Create] on the [Operation] menu and the Create File System Wizard screen appears. It is shown in the following Figure Create File System (1).

In the Create File System Wizard screen, set the node information and enter the mount point.

Figure Create File System Wizard (1)

	Node name	Host name	Primary MDS	Secondary MDS	MDS
<input checked="" type="checkbox"/>	endou	endou <input type="button" value="Select"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	gobou	gobou <input type="button" value="Select"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

MDS node: all nodes 2 nodes

Mount point:

Make directory: yes no

Buttons: Cancel, Reset, Next, Help

Warning: Applet Window



See

For an explanation of each parameter, see the explanation of sfcnewfs (1M) in [Appendix F.2.14, "sfcnewfs\(1M\)."](#)

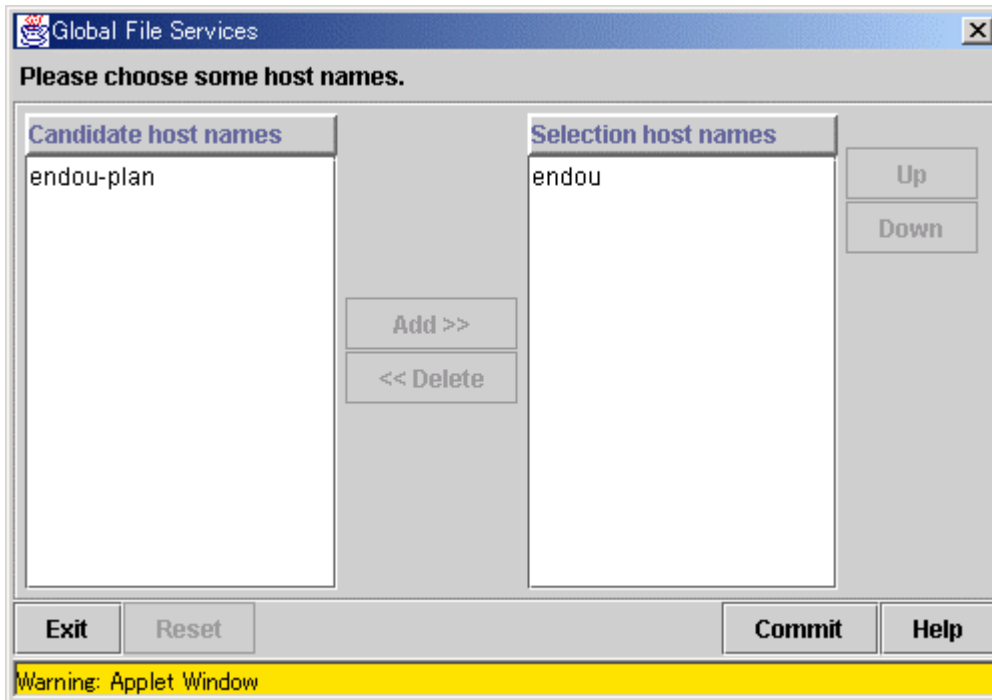
Selecting node names

In the "Node name", select the names of the sharing nodes. Be sure to select at least two nodes. It is not possible to deselect the local node.

Selecting host names

To select a host name other than the displayed host name or to select more than one host name, click the <Select> button and then select a host name or names to use in each node. When several host names have already been selected, position the mouse cursor to a host name to popup all host names.

Figure Host Name Selection Wizard



Please choose the host name of LAN to be used from the left-hand side candidate host name list and click <add> to add it to the right-hand side selected host name list. Also, set the priority of the host names by using <up> and <down>. The order of preference is from top down.

Selecting the primary MDS and secondary MDS

Specify the nodes on which the Primary MDS and Secondary MDS will be started. A single, unique node must be selected for each **MDS**.

Selection of "MDS placement target" and "MDS placement nodes"

Specify the range of nodes on which MDS may execute. If all nodes are selected as "MDS placement nodes," MDS may execute on all shared nodes. In such case, priority will given to the nodes specified as "Primary MDS" and "Secondary MDS." If exactly two nodes are selected as "MDS placement nodes", they will be designated as the "Primary MDS" and "Secondary MDS" nodes.

Setting mount point and creating a directory

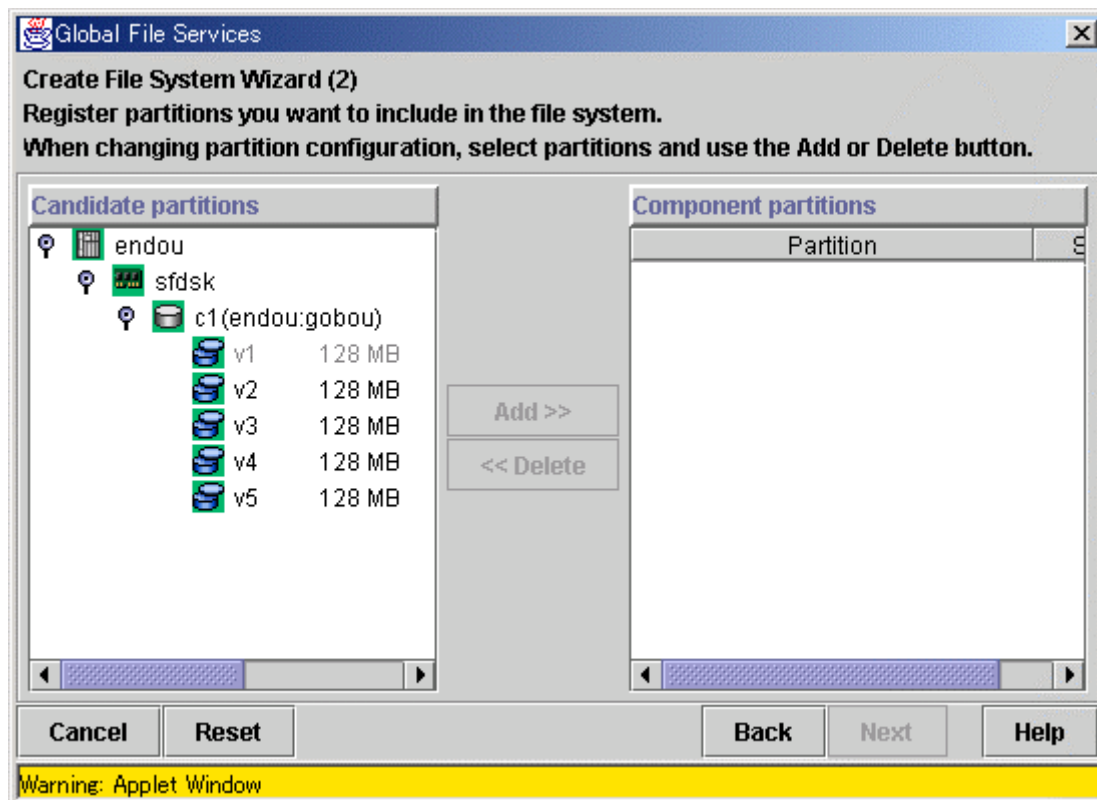
Specify the full path name of a mount point in the "Mount point" field. If "yes" is selected for "Make directory", a directory is created with the following attributes:

Owner: root
Group: sys
Access right: 775

(3) Selecting a partition

After completing the MDS configuration, the register **partition** screen will be displayed. It is shown in the following Figure Create File System Wizard (2).

Figure Create File System Wizard (2)



Select the partition from [Candidate partitions] field, and click the <Add> button. More than one partition can be selected at a time. However, a partition or a management partition selected cannot already be part of an existing file system.

After selecting a partition, click the <Next> button to continue. The next screen is shown in Figure Create File System Wizard (3).

To return to the Create File System Wizard (1) screen, click the <Back> button.

To cancel the file system creates operation, click the <Cancel> button.



Note

A GDS logical volume must be active to be registered as a partition.

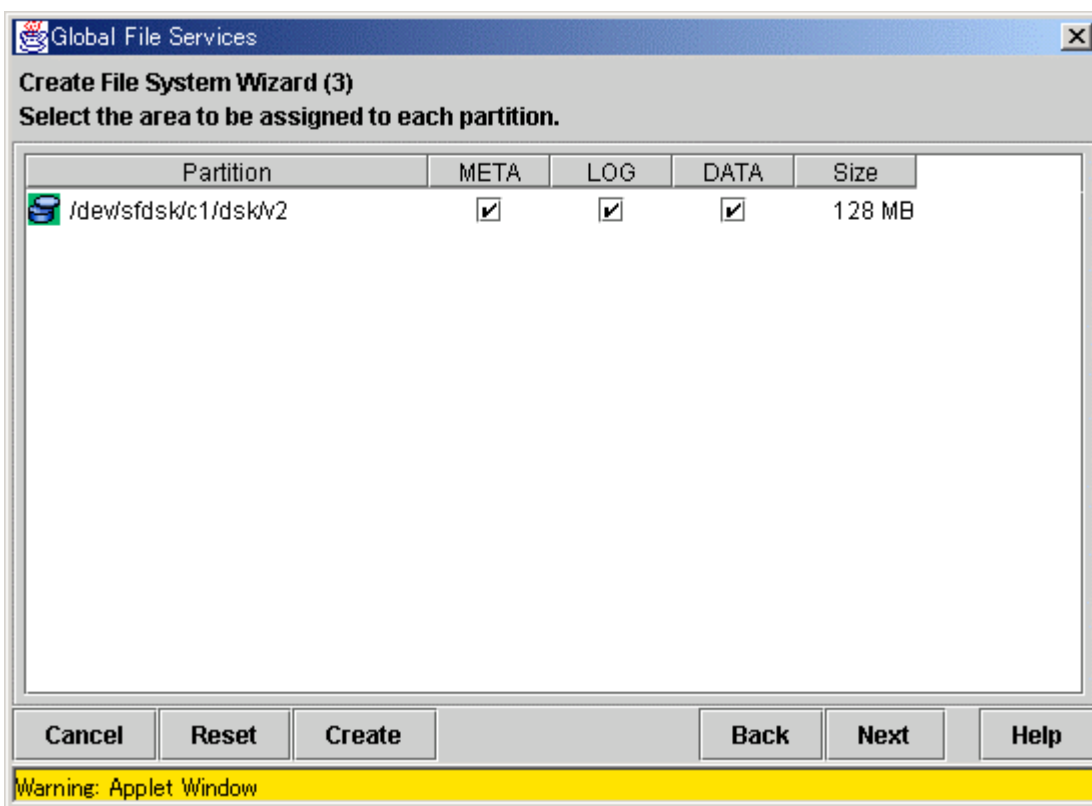
(4) Setting partition information

In the Create File System Wizard (3) screen, select the area (META/LOG/DATA) to be assigned to each partition selected in the Create File System Wizard (2).
The partition to which META is assigned becomes a **representative partition**.



For an explanation of the combinations of areas (META/LOG/DATA) assigned to partitions, see [Section 13.1.6, "Partition configuration."](#)

Figure Create File System Wizard (3)



After setting the area information items, click the <Next> button to proceed to the next screen which is shown in the following Figure Create File System Wizard (4).

To restore the settings to their default values, click the <Reset> button.

To return to the Create File System Wizard (2) screen, click the <Back> button.

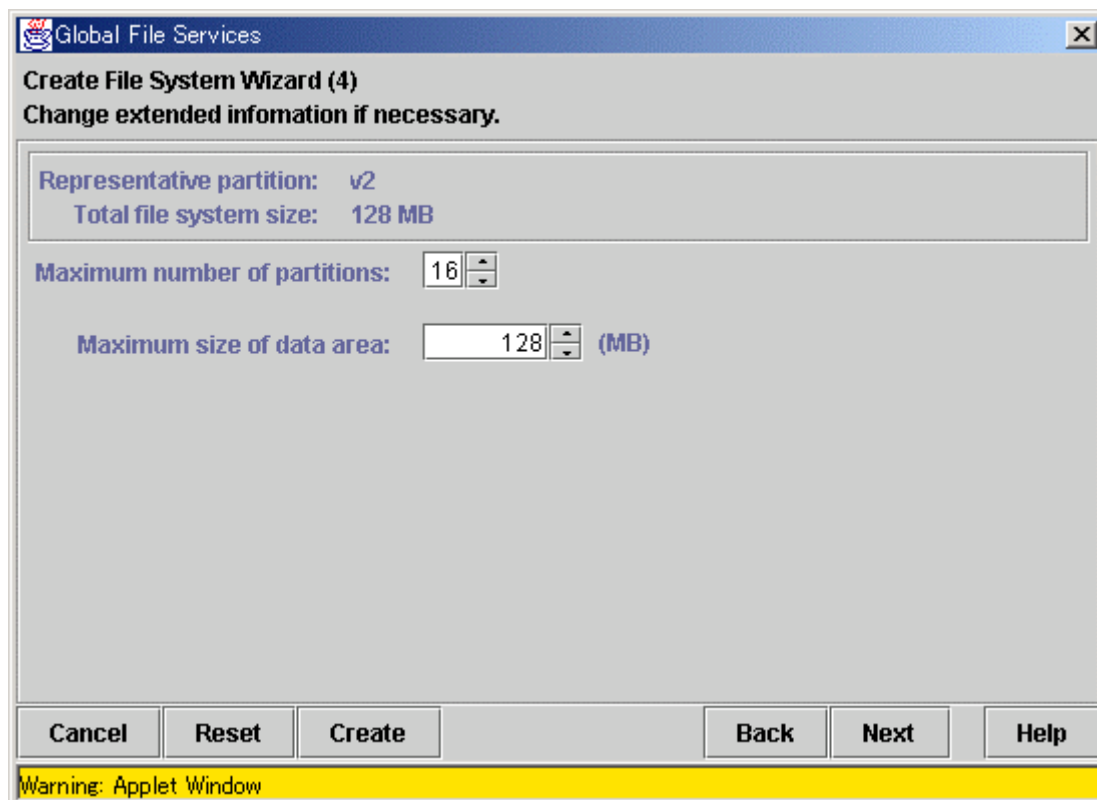
To cancel the file system creation operation, click the <Cancel> button.

If you do not need to change extended information, detailed information and mount information, click the <Create> button to create the file system.

(5) Setting extended information

In the Create File System Wizard (4) screen, the extended information values should be set such that they allow for future expansion of the file system and configuration changes.

Figure Create File System Wizard (4)

**Note**

Set the maximum number of partitions and maximum data area size after adequately considering future data area expansion for the file system to be created. For details, see [Section 15.2.7, "For future expansion."](#)

**See**

For an explanation of each parameter, see the explanation of `sfnewfs(1M)` in [Appendix F.2.14, "sfnewfs\(1M\)."](#)

When the extended information items are correct, click the <Next> button to proceed. The next screen is shown in the following Figure Create File System Wizard (5).

To restore the settings to their default values, click the <Reset> button.

To return to the Create File System Wizard (3) screen, click the <Back> button.

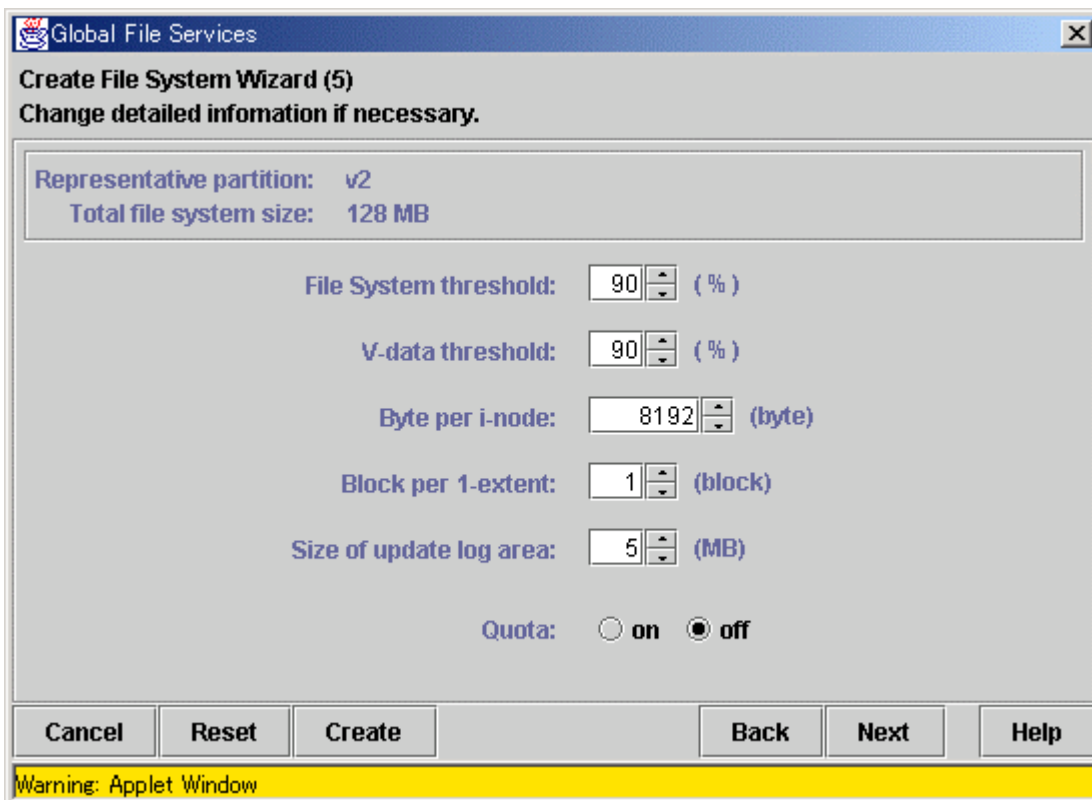
To cancel the file system creation operation, click the <Cancel> button.

If you do not need to change the detailed information and mount information, click the <Create> button to create the file system.

(6) Setting detailed information

In the Create File System Wizard (5), set the detailed information.

Figure Create File System Wizard (5)



See

For an explanation of each parameter, see the explanation of `sfcnewfs(1M)` in [Appendix F.2.14, "sfcnewfs\(1M\),"](#) and [Section 21.4, "quota function."](#)



Note

The file system usage limit is the value obtained by subtracting from 100 the minimum percentage of free area in the file system specified by the `-f` option of `sfcnewfs(1M)`.
 The V-data area usage limit is the value obtained by subtracting from 100 the percentage of minimum free area of the V-data area specified by the `-m` option of `sfcnewfs(1M)`.

When the detailed information items are correct, click the <Next> button to continue. The next screen is shown in the following Figure Create File System Wizard (6).

To restore the settings to their default values, click the <Reset> button.

To return to the Create File System Wizard (4) screen, click the <Back> button.

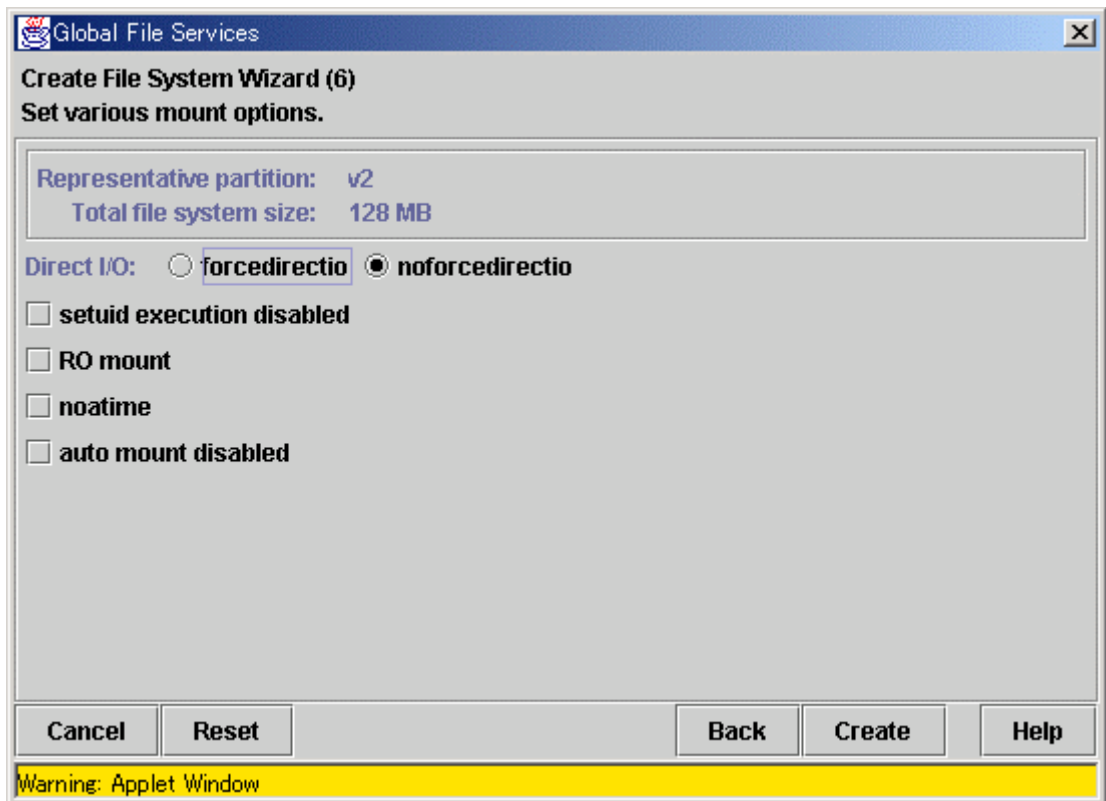
To cancel the file system creation operation, click the <Cancel> button.

If you do not need to change the mount information, click the <Create> button to create the file system.

(7) Setting mount information

In the Create File System Wizard (6) screen, set the mount information.

Figure Create File System Wizard (6)



After setting the mount information, click the <Create> button to create the file system.

To restore the settings to their default values, click the <Reset> button.

To return to the Create File System Wizard (5) screen, click the <Back> button.

To cancel the file system creation operation, click the <Cancel> button.

19.3 Change

The **GFS Management View** provides three change operations:

- Changing the file system attributes (changing the mount information, changing the size of the update log area, file system tuning, **quota function**)
- Changing file system configuration (file data partition addition)
- Changing **shared node** information



Note

Note of Operation

- The information for the Management View must be updated before the changes are reflected in the display of the Main Screen. To immediately update the information, on the Main Screen click [Update Now] on the [View] drop-down menu.
- If the **sfcs** information is not available, the file system can not be changed. Take action according to the output error message.

19.3.1 Changing in file system attributes

The following describes the procedure for changing file system attributes using the **GFS Management View**.

Changing the file system attributes includes changing the mount information, the update log area size, the file system tuning and the setting of the quota function.

(1) Unmount of a file system

The setting of the quota function can be changed while a file system is mounted.
Before changing any other file system attribute, unmount the file system.

(2) Selecting a file system

From the [Mount tree] or [FileSystem Information] field in the main screen, select the file system you want to change. Only file systems of the **sfcs** type can be selected.
Click [Change Attributes] on the [Operation] menu, and the [File System Attributes] screen appears.
From the [File system Attribute] screen, the mount information, share information, and detailed information can be selected using the screen switching tabs.

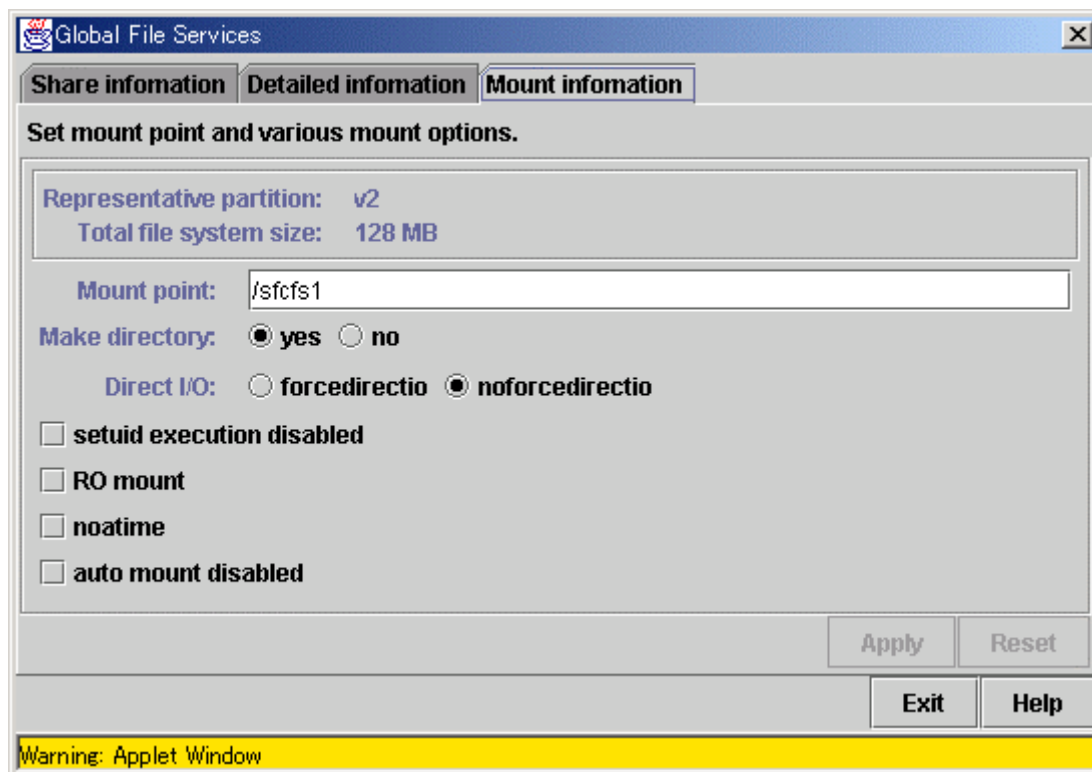
(3) Changing the mount information

In the [Mount information] screen, the mount information can be changed.
To change the settings, change the corresponding parameters.
To restore the settings to their previous values, click the <Reset> button.
To execute the mount information change operation, click the <Apply> button.

Specify the full path name of a mount point in the "Mount point" field. If "yes" is selected for "Make directory", a directory is created with the following attributes:

Owner: root
Group: sys
Access right: 775

Figure Mount information for file system attributes



(4) Changing detailed information

In the [Detailed information] screen, change the update log area size, tune the file system, and change the setting of the quota function.

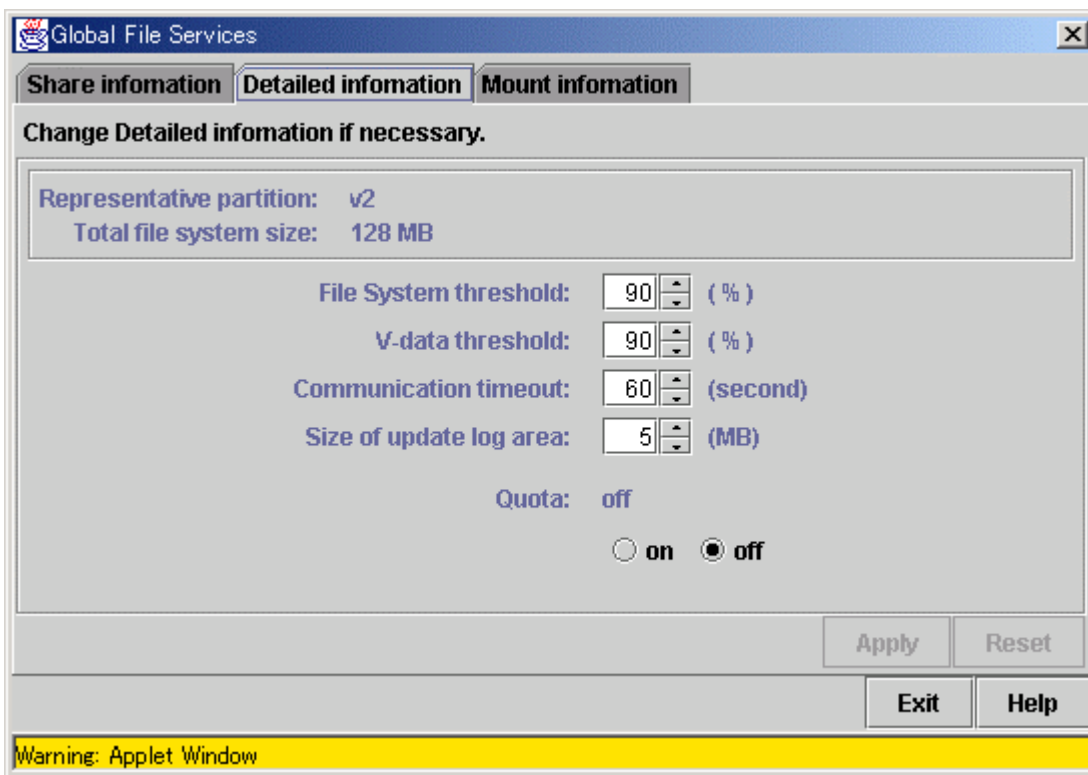
To change the settings, change the corresponding parameters.

To restore the settings to their previous values, click the <Reset> button.

To execute the detailed information change operation, click the <Apply> button.

To close the [File System Attributes] screen, click the <Exit> button.

Figure Detailed information for file system attributes



Note

Update log area

The update log area size can be specified only if the **update log partition** is separated.



See

For an explanation of each parameter, see [Appendix F.2.2 "sfcadm\(1M\)," "F.2.29 sfctune\(1M\),"](#) and [Section 21.4, "quota function."](#)

(5) Mount of a file system

After the completion of the file system attribute change operation, a file system is in mounted.

After completion of the file system attribute change operation, mount the file system if it is unmounted.

19.3.2 Changing the file system configuration (partition addition)

This section explains the procedure for adding **file data partitions**.

It is easy to change a file system's configuration because the file system configuration change operation uses a wizard.

To start the Change File System Configuration Wizard, click [Change Configuration] on the [Operation] drop-down menu.



Note

Note of Operation

If the assigned values are invalid, the <Back> button may be clicked to display the previous screen.

(1) Unmount of a file system

Before the update log area is separated, the file system must be unmounted on all nodes.

File data and meta-data partitions can be added to a mounted file system.

(2) Setting shared disk device

The disk containing the **partitions** to be added to a GFS shared file system must already be set a shared **GDS** disk.

To specify a GDS logical volume as the target device, it must be active.



Note

Check that the disk is set up as shared for simultaneous access.

If a partition has set up the disk as shared by automatic detection of shared disks, it is already set as a shared disk for simultaneous access and no further action is needed.

When registering a GDS logical volume with a class, set the disk type as shared and the scope to include all the nodes that share the GFS Shared File System.



See

For GDS disk class operations, see the applicable items under "Operation using Global Disk Services Management View" in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

For GDS logical volume operations, see the applicable items under "Operation using Global Disk Services Management View" in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

(3) Selecting a file system

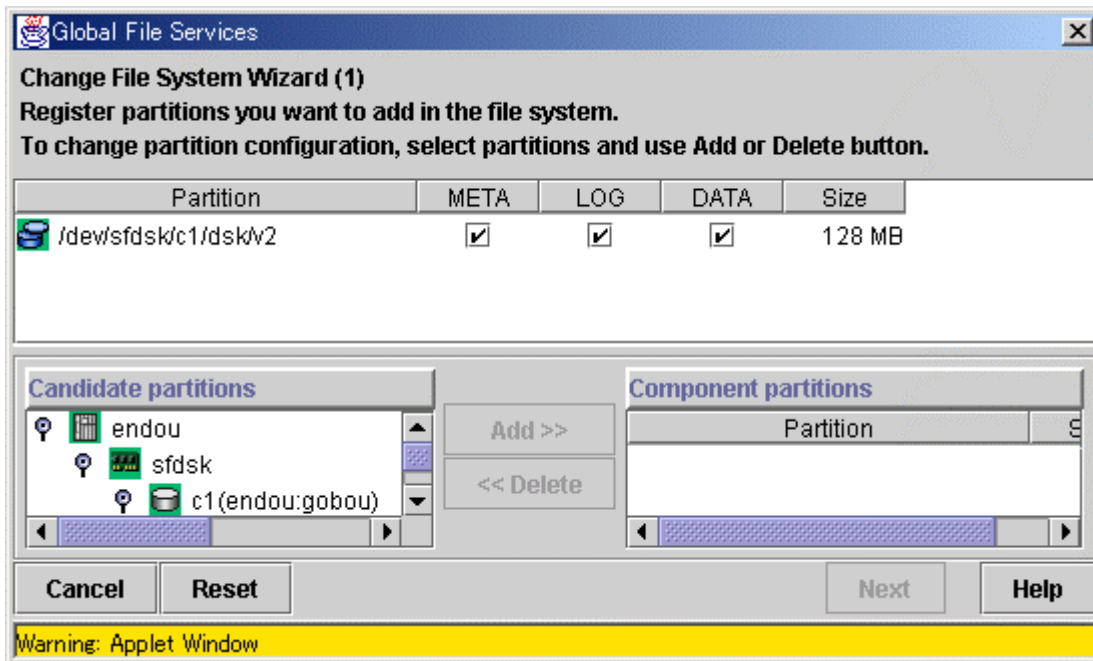
From the [Mount tree] or [FileSystem Information] field in the main screen, select the file system to be changed. Only file systems of the sfcs type can be selected.

Click [Change Configuration] on the [Operation] menu, the Change File System Configuration Wizard (1) screen appears.

(4) Selecting a partition

From the [Candidate partitions] field, select the **partition** to be added and click the <Add> button. More than one partition can be selected at a time. Note that those partitions that are already in use for file systems cannot be selected.

Figure Change File System Configuration Wizard (1)



After selecting the partitions, click the <Next> button to continue. The next screen is shown in the following Figure Change File System Configuration Wizard (2).

To cancel the file system configuration change operation, click the <Cancel> button.

(5) Setting partition information

In the Change File System Configuration Wizard (2) screen, select the area to be assigned to each partition added with the Change File System Configuration Wizard (1) screen. By default, DATA is set for the area assigned to an added partition.



See

For an explanation of the combinations of areas (META/DATA) assigned to partitions, see [Section 13.1.6, "Partition configuration."](#)



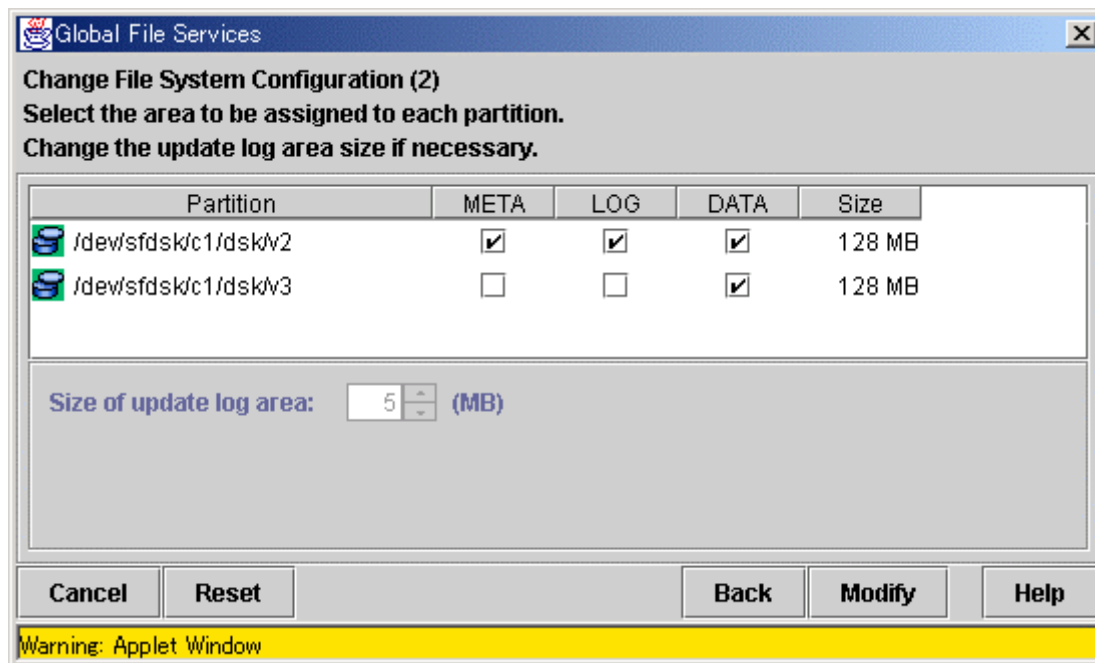
Note

Areas assigned to partitions

The following rules apply to the areas assigned to partitions:

- A DATA area is assigned to an added partition.
- The META area cannot be changed.

Figure Change File System Configuration Wizard (2)



For an explanation of each parameter, see the explanation of `sfcsadd(1M)` in [Appendix F.2.1, "sfcsadd\(1M\)."](#)

After setting the information items, click the <Modify> to execute the file system configuration change operation.

To restore the settings to their previous values, click the <Reset> button.

To return to the Change File System Configuration Wizard (1), click the <Back> button.

To cancel the file system configuration change operation, click the <Cancel> button.

(6) Mount of a file system

After completion of the file system attribute change operation, mount the file system if the file system is unmounted.

19.3.3 Changing the shared node information

The following describes the procedure for changing sharing-node information using the **GFS Management View**.

(1) Unmount a file system

To change the **shared node** information, the file system must be unmounted on all nodes.

(2) Selecting a file system

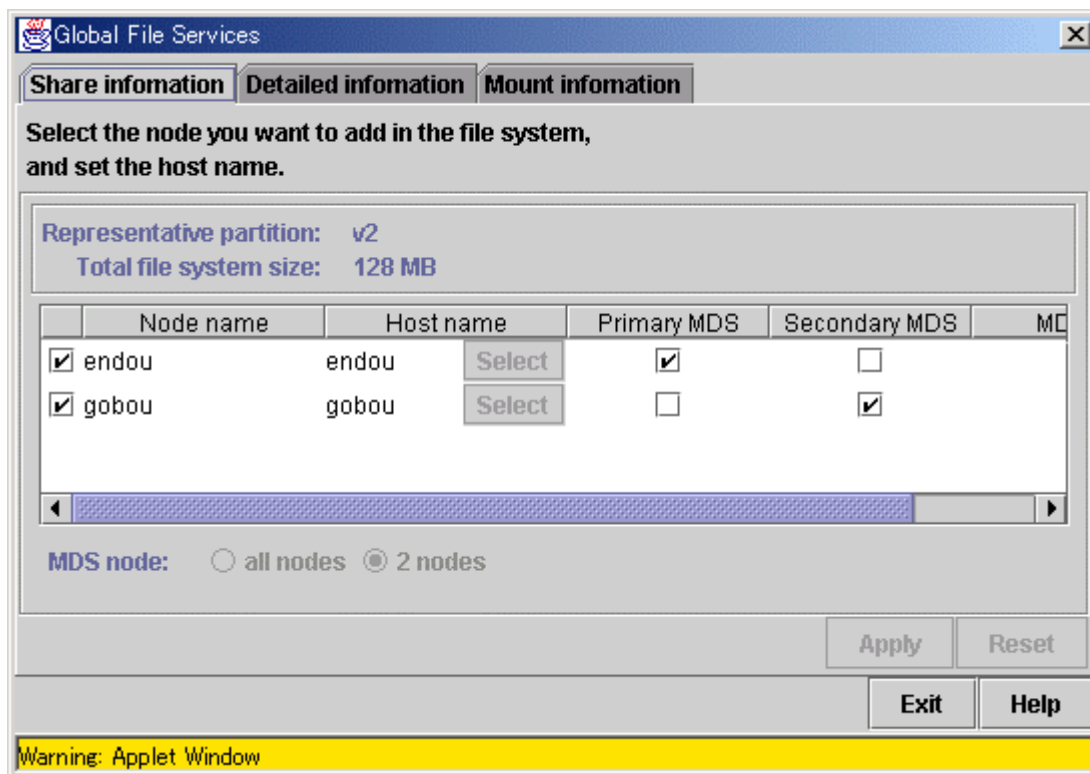
From the [Mount tree] or [Filesystem Information] field in the main screen, select the file system you want to change. Only file systems of the `sfcs` type can be selected.

Click [Change Attributes] on the [Operation] menu and go to [Share information] screen using the screen-switching tab.

(3) Changing share information

In the [Share information] screen, the information on the **nodes** sharing the file system can be changed. To change the settings, change the corresponding parameters. To restore the settings to their previous values, click the <Reset> button.

Figure Share information for file system attributes



Selecting node names

In the "Node name" column, select the names of the sharing nodes. Be sure to select at least two nodes. It is not possible to deselect the local node, the primary MDS or the secondary MDS.

Selecting a host name

In the "Host name" column, select the LAN host name for the new hosts. The host name for each node already selected cannot be changed.



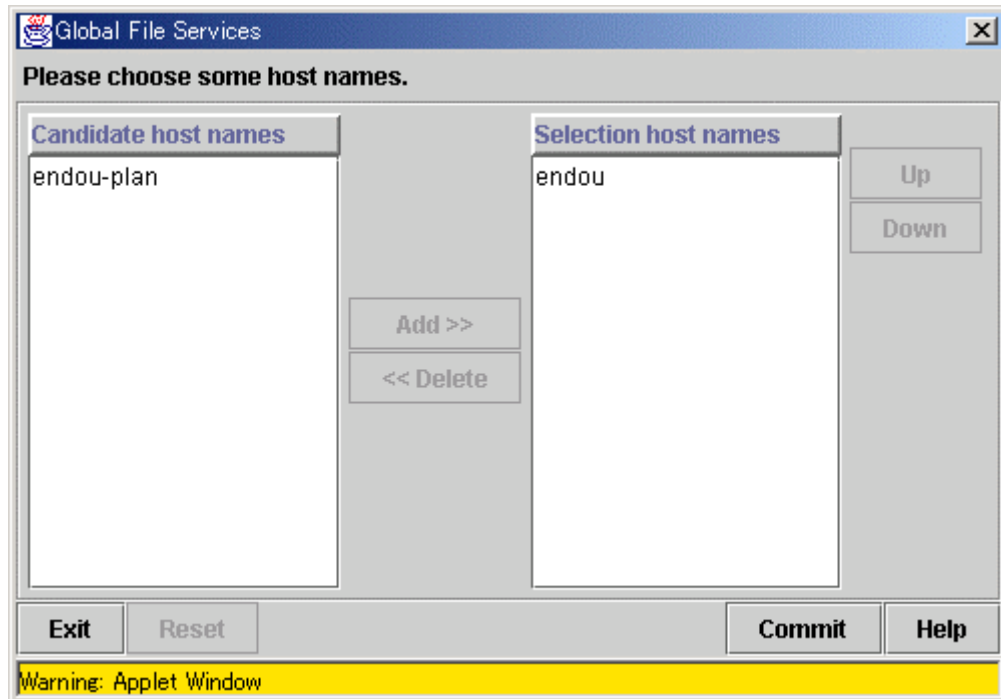
Primary MDS and secondary MDS

The primary MDS and secondary MDS cannot be changed. If a change is necessary, the file system must be deleted and re-created.



For an explanation of each parameter, see the explanation of sfcadm(1M) in [Appendix F.2.2, "sfcadm\(1M\)."](#)

Figure Host Name Selection Wizard

**(4) Confirming the sharing-node information change operation**

To execute the sharing-node information change operation, click the <Apply> button.
To close the [File System Attributes] screen, click the <Exit> button.

(5) Mount a file system

After completion of the file system attributes change operation, mount the file system.

19.4 Deletion

19.4.1 Deleting the file system

The following describes the procedure for deleting a file system using the **GFS Management View**.



Note of Operation

- The information for the Management View must be updated before the changes are reflected in the display of the Main Screen. To immediately update the information, on the Main Screen click [Update Now] on the [View] drop-down menu.
- If the sfcs information is not available, the file system can not be changed. Take action according to the output error message.

(1) Unmount the file system

Unmount the file system on all nodes before deletion.



For detail on the unmount procedure, see [Section 20.2.4.4, "Unmount of all nodes."](#)

(2) Selecting the file system

From the [Mount tree] or [File system Information] field in the main screen, select the file system to be deleted. Only file systems of the sfcs type can be selected.

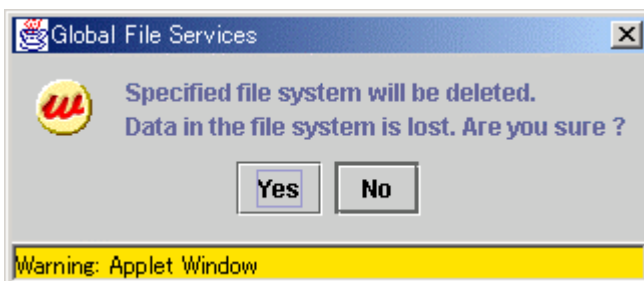
(3) Deleting the file system

Click [Delete] on the [Operation] menu and the following warning message appears.

To delete the file system, click the <Yes> button.

To cancel the deletion of the file system, click the <No> button.

Figure File system deletion warning message



Chapter 20 File System Operations (Command)

This chapter describes the GFS Shared File System executed by command.



Note

Before creating a file system, the management partition must be set up.



See

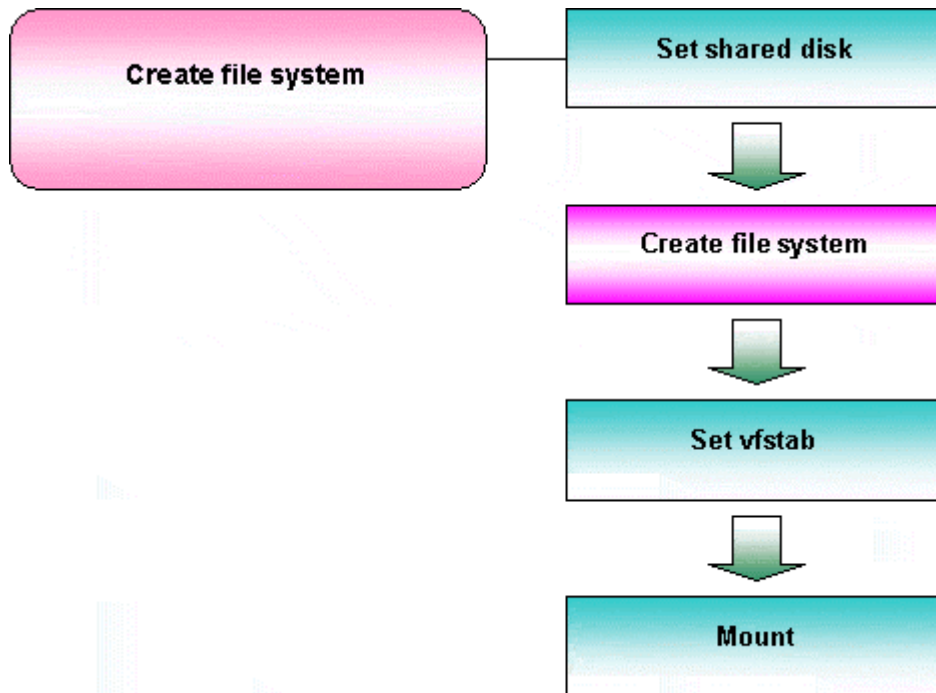
For details on the management partition, see [Section 16 “Management Partition.”](#)

20.1 Operation flow

This section describes the flow of operations in the GFS Shared File System.

20.1.1 Creation

Figure Flow of the create file system operation



20.1.2 Change (in quota function status)

As shown below, the quota function status can be changed while the file system is mounted or unmounted.

Figure Flow of the change in quota function status (mounted file system)

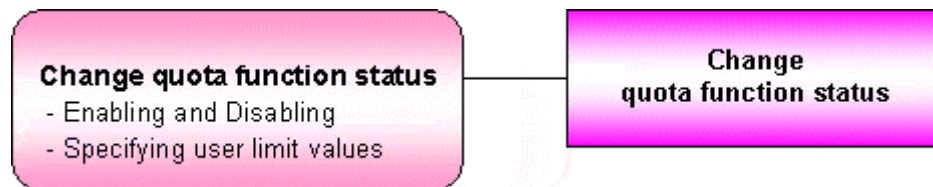
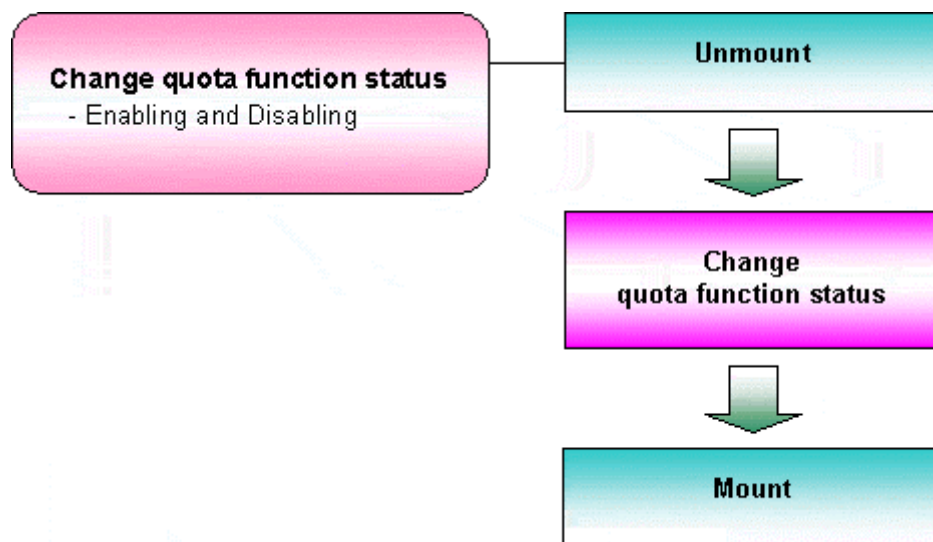
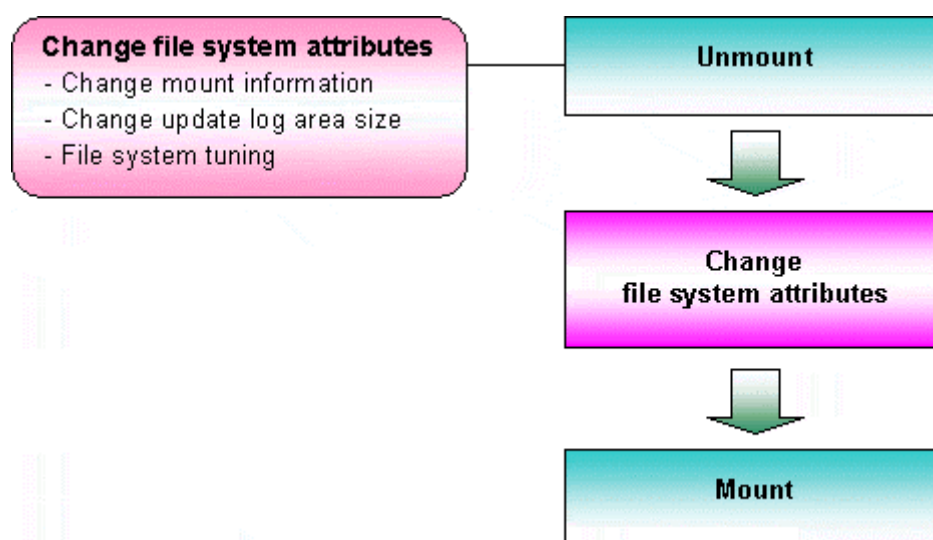


Figure Flow of the change in quota function status (unmounted file system)



20.1.3 Change (in file system attributes)

Figure Operation flow for change in file system attributes



20.1.4 Change (partition configuration)

The following describes the flow of operations to change the partition configuration of a file system. In **GFS Management View**, the following configuration changes can be made:

- Addition of a **file data partition**
- Addition of a **meta-data partition**

A file data partition can be added to a mounted or unmounted file system. An update log area can be separated only when the file system is unmounted.



For details on each operation in the flow described in the following figure, see [Section 20.5, "Change \(partition configuration\)."](#)

The following figures show the operation flow for adding partitions to mounted and unmounted file systems.

Figure Flow of the change partition configuration (unmounted file system)

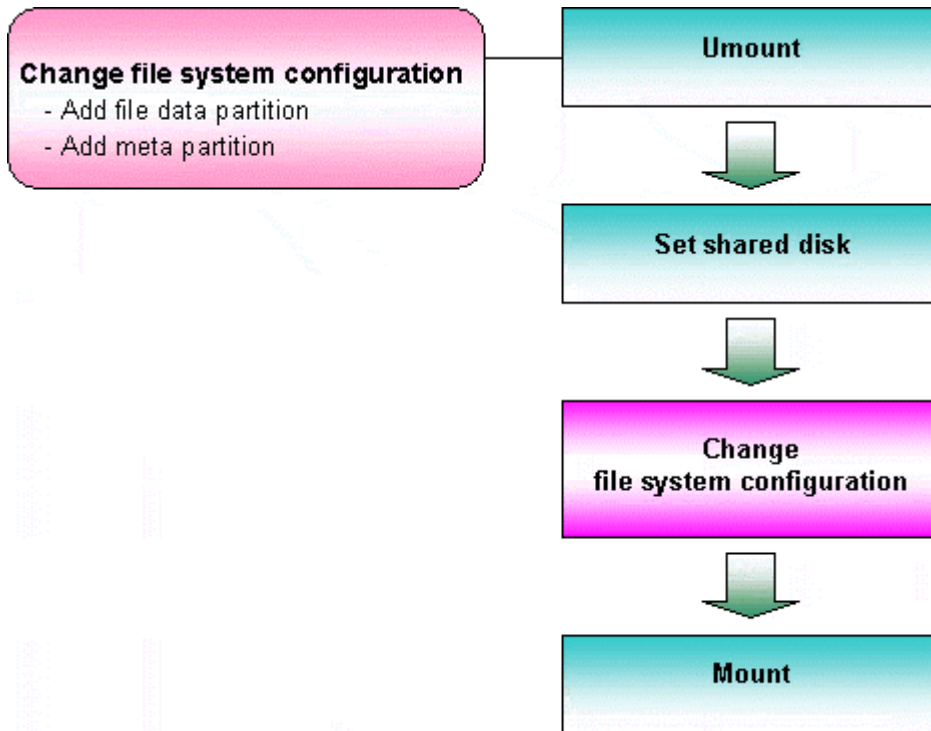
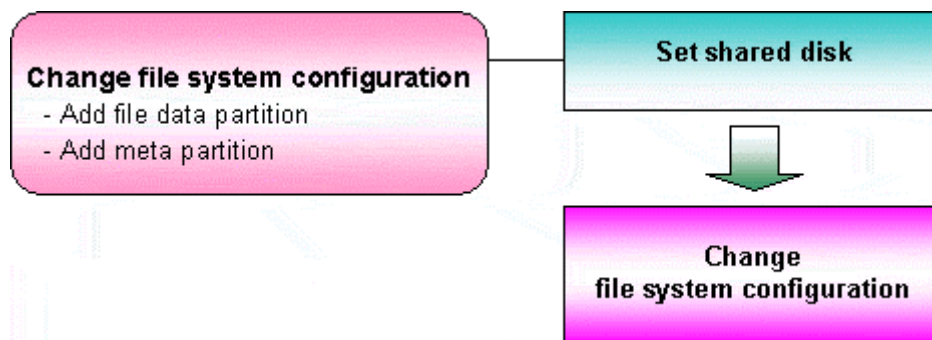
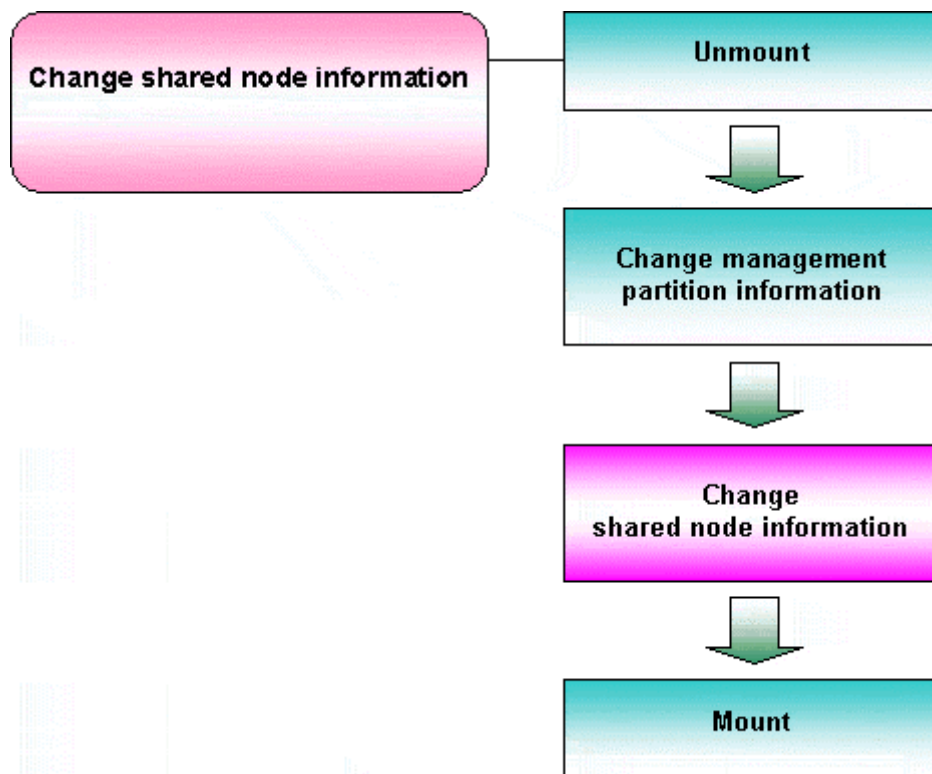


Figure Flow of the change partition configuration (mounted file system)



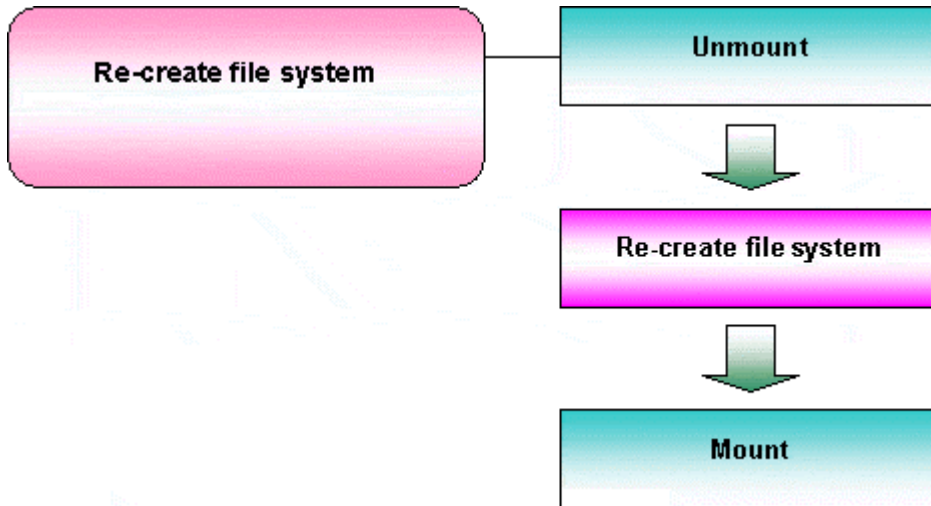
20.1.5 Change (shared node information)

Figure Operation flow for changing the shared node information



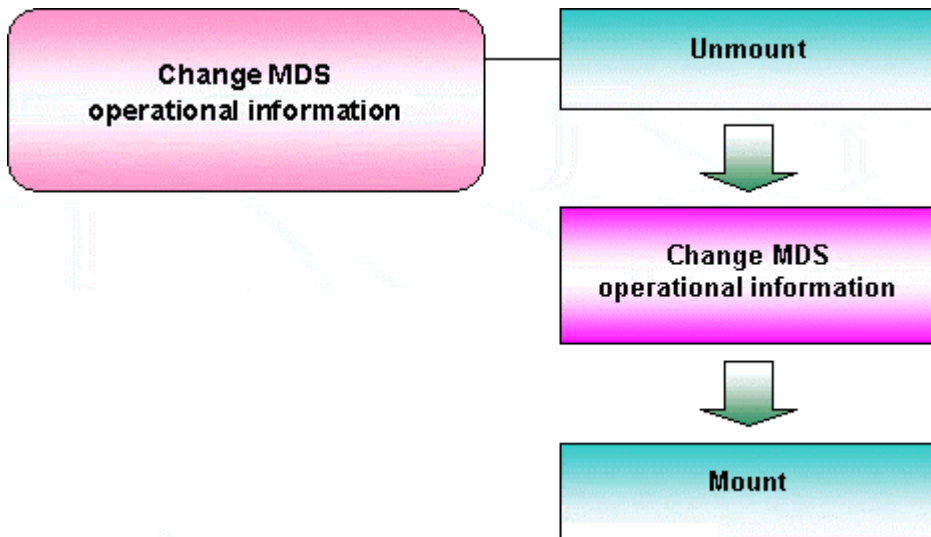
20.1.6 Changes (re-creating a file system)

Figure Operation flow for re-creating a file system



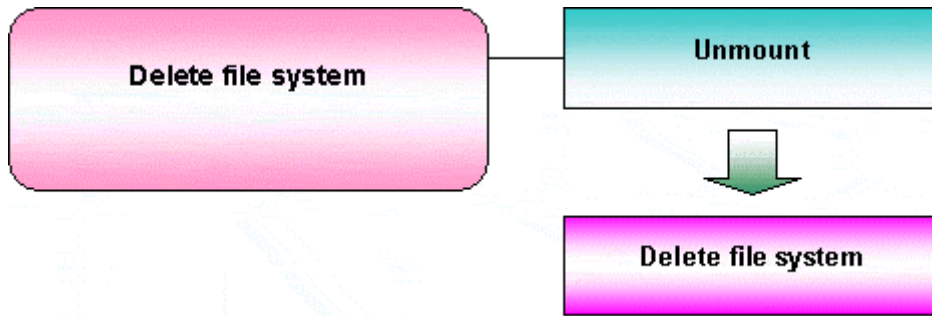
20.1.7 Change (MDS operational information)

Figure Operation flow for changing the MDS operational information



20.1.8 Deletion

Figure Operation flow for deleting a file system



20.2 Flow from file system creation to mount/unmount

The flow from the GFS File System creation to mount/unmount is explained below, in the following order:

1. Setting up a shared disk
2. File system creation with `sfcnewfs(1M)`
3. Setting up `/etc/vfstab`
4. `mount/umount/all nodes (u)mount`

20.2.1 Setting shared disks

The GFS Shared File System uses GDS logical volumes as shared devices. The status of the GDS logical volumes must be ACTIVE on each node.



Note

Check that the disk is set up as shared for simultaneous access.

If a partition has set up the disk as shared by automatic detection of shared disks, it is already set as a shared disk for simultaneous access and no further action is needed.

When registering a GDS logical volume with a class, set the disk type as shared and the scope to include all the nodes that share the GFS Shared File System.



See

For GDS disk class operations, see the applicable items under "Operation using Global Disk Services Management View" in the *"PRIMECLUSTER Global Disk Services Configuration and Administration Guide."*

For GDS logical volume operations, see the applicable items under "Operation using Global Disk Services Management View" in the *"PRIMECLUSTER Global Disk Services Configuration and Administration Guide."*

20.2.2 Creating a file systems

To create a GFS shared file system, use `sfcnewfs(1M)`.



See

For details on `sfcnewfs(1M)`, see [Appendix F.2.14, "sfcnewfs\(1M\)."](#)



Note

When the following error message is displayed, the `sfcnewfs(1M)` command was unable to create a file system. `sfcnewfs(1M)` was unable to obtain sufficient meta-data area which is approximately 10% of the file system size.

```
mkfs_sfcs: ERROR: Master partition is small, meta area not enough
```

In such a case, create a file system after reducing the size of the specified file data partition or add meta-data partitions by executing the `asfcadd(1M)` command.



See

For details on partition addition, see ["20.5 Change \(partition configuration\)."](#) For details on the `sfcadd(1M)` command, see [Appendix F.2.1 "sfcadd\(1M\)."](#)

20.2.2.1 Defaults of parameters used by sfcnewfs(1M)

The defaults of parameters used by sfcnewfs(1M) are as follows:

Table Defaults of parameters used by sfcnewfs(1M)

Parameter	Default
Data block size	Always 8192 bytes
Minimum free space (-f <i>free</i>)	10%
Optimization type (-o <i>space time</i>)	time
Number of i-nodes (-i <i>nbpi</i>)	One per data block (8192 bytes)
Update log area size (-g <i>logsize</i>)	1% of the file system size However, the size must be at least 5 megabytes but not more than 50 megabytes.
Meta-data area size (-a <i>metasz</i>)	About 10% of the file system size
Maximum number of partitions (-l <i>maxvol</i>)	16
Number of MDS place in nodes (-S <i>all num</i>)	2



Note

When building a file system, the file system should be configured to operate MDS on all nodes.

20.2.2.2 Examples of creating a representative file system

Examples of creating a typical GFS shared file system are given below.

- Single partition configuration

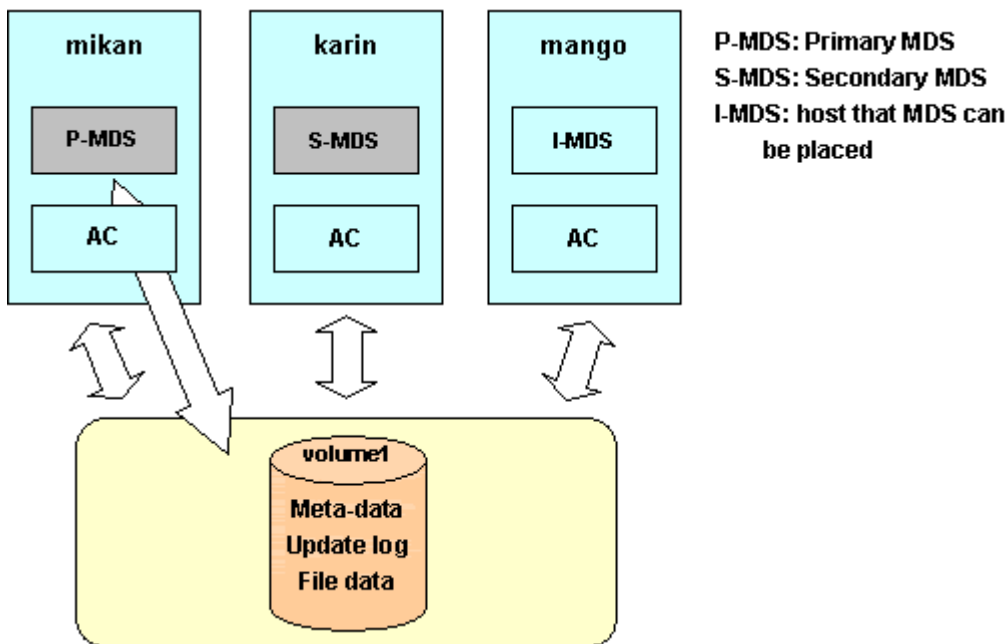
To create a GFS shared file system with a single partition, specify the **partition** configuring the file system. In this case, areas for all types of data (**meta-data area**, **update log area**, and **file data area**) are created in the **representative partition**.

The following is an example of using `sfnewfs(1M)` to create a file system with a single partition.

```
# sfnewfs -H mikan,karin,mango -S all /dev/sfdsk/gfs/rdisk/volume1

Representative partition (meta-date, log, and data) : /dev/sfdsk/gfs/rdisk/volume1
Shared hosts : mikan, karin, and mango
Primary MDS : mikan
Secondary MDS : karin
MDS operational nodes : mikan, karin, and mango
```

Figure Single partition configuration

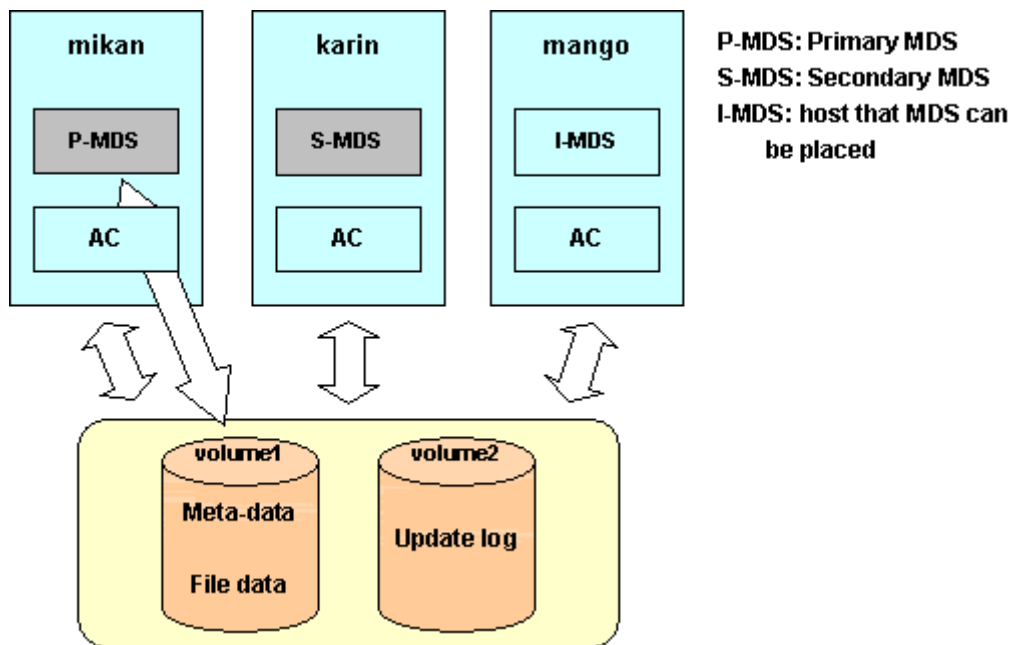


- Separating the update log area
 To separate the update log area as another partition, specify both a representative partition and an **update log partition**. In the representative partition, a meta-data area and data area are created.
 The following is an example of using `sfnewfs(1M)` to create a file system in which the update log area is separated as another partition.

```
# sfnewfs -H mikan,karin,mango -S all -L /dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/
gfs/rdsk/volume1

Representative partition (meta-data, and data) : /dev/sfdsk/gfs/rdsk/volume1
Log partition : /dev/sfdsk/gfs/rdsk/volume2
Shared hosts : mikan, karin, and mango
Primary MDS : mikan
Secondary MDS : karin
MDS operational nodes : mikan, karin, and mango
```

Figure Separating the update log area



- Multiple file data partitions configuration

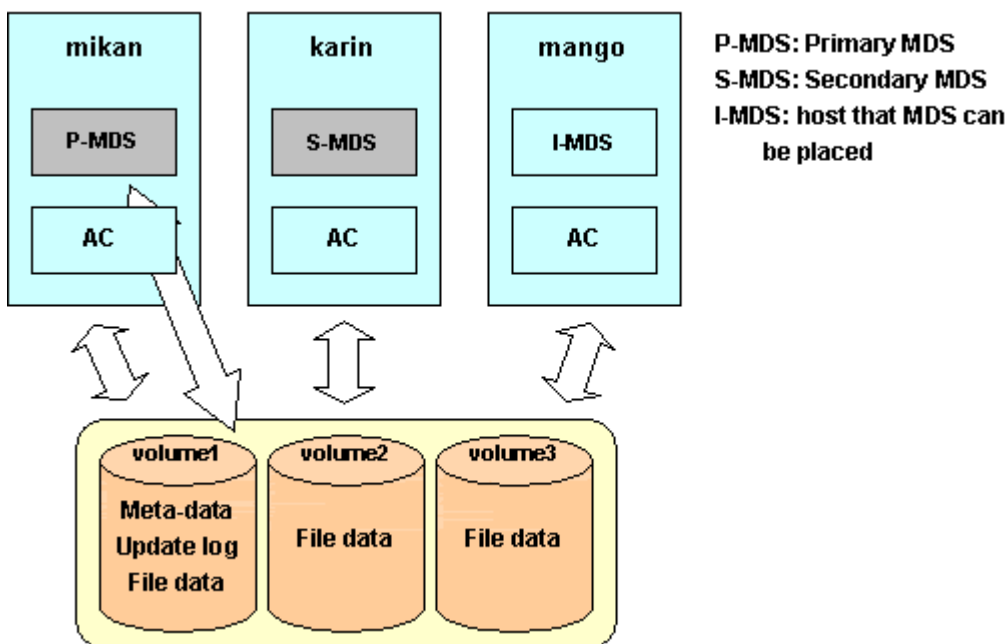
To specify multiple partitions in the data area, specify a representative partition and **file data partitions**. In this case, a meta-data area, update log area, and data area are created in the representative partition.

The following is an example of using `sfnewfs(1M)` to create a file system with multiple file data partitions.

```
# sfnewfs -H mikan,karin,mango -S all -D /dev/sfdsk/gfs/rdsk/volume2,/dev/sfdsk/gfs/rdsk/volume3 /dev/sfdsk/gfs/rdsk/volume1
```

Representative partition (meta-data, log, and data) : /dev/sfdsk/gfs/rdsk/volume1
 Data partition : /dev/sfdsk/gfs/rdsk/volume2, /dev/sfdsk/gfs/rdsk/volume3
 Shared hosts : mikan, karin, and mango
 Primary MDS : mikan
 Secondary MDS : karin
 MDS operational nodes : mikan, karin, and mango

Figure Multiple data partitions configurations



- Separating the update log area and multiple file data partitions configuration

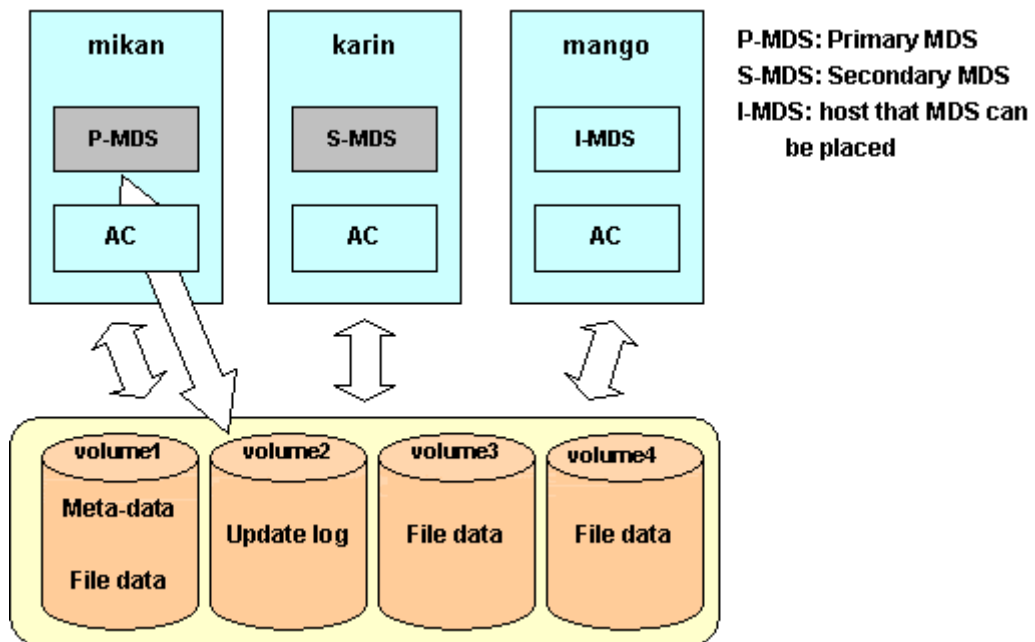
To separate the update log area as another partition and specify multiple partitions in the area, use the specification below.

The following is an example of using `sfcnewfs(1M)` to create a file system in which an update log partition and multiple file data partitions are specified.

```
# sfcnewfs -H mikan,karin,mango -L /dev/sfdsk/gfs/rdsk/volume2 -S all -D /dev/sfdsk/gfs/rdsk/volume3, /dev/sfdsk/gfs/rdsk/volume4 /dev/sfdsk/gfs/rdsk/volume1
```

Representative partition (meta-data and data) : /dev/sfdsk/gfs/rdsk/volume1
 Log partition : /dev/sfdsk/gfs/rdsk/volume2
 Data partition : /dev/sfdsk/gfs/rdsk/volume3, /dev/sfdsk/gfs/rdsk/volume4
 Shared hosts : mikan, karin, and mango
 Primary MDS : mikan
 Secondary MDS : karin
 MDS operational nodes : mikan, karin, and mango

Figure Separating the update log area and multiple file data partitions configuration



- Separating the data area

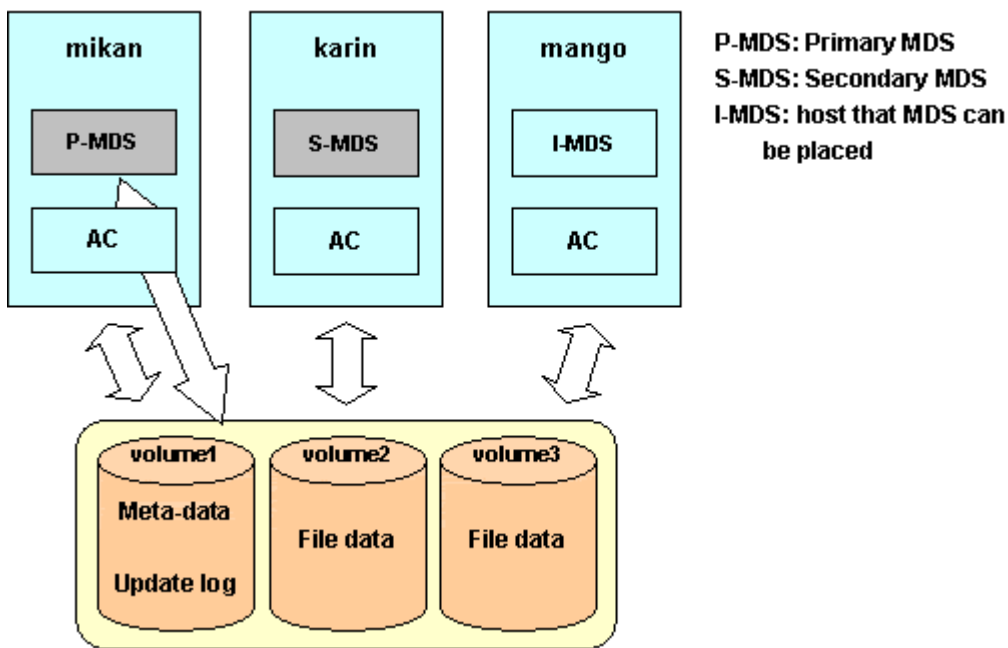
If multiple partitions in a data area are specified with the **-A** option, the data area is not included in the representative partition.

The following is an example of using `sfcnewfs(1M)` to create a file system in which multiple partitions are specified and the data area is not included in the representative partition.

```
# sfcnewfs -A -H mikan,karin,mango -S all -D /dev/sfdsk/gfs/rdsk/volume2,/dev/sfdsk/
gfs/rdsk/volume3 /dev/sfdsk/gfs/rdsk/volume1

Representative partition (meta-data and log) : /dev/sfdsk/gfs/rdsk/volume1
Data partition : /dev/sfdsk/gfs/rdsk/volume2, /dev/sfdsk/gfs/rdsk/volume3
Shared hosts : mikan, karin, and mango
Primary MDS : mikan
Secondary MDS : karin
MDS operational nodes : mikan, karin, and mango
```

Figure Separating the data area



- Separating the file data and update log areas

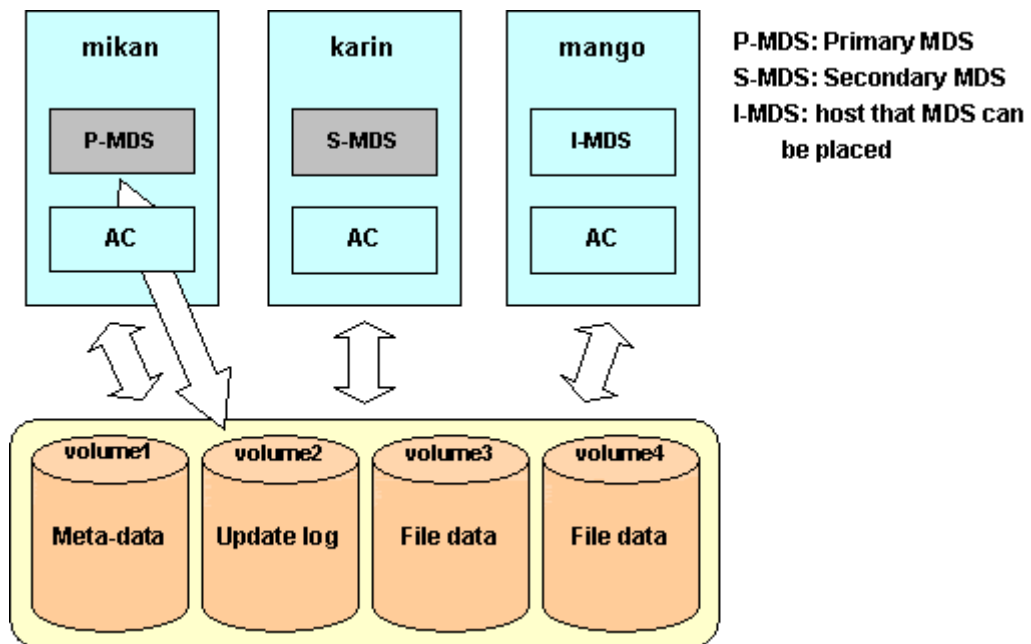
If a data area is not included in the representative partition by specifying the `-A` option, the meta-data area, update log area, and data area can be configured on separate partitions. This is done by specifying a separate update log area.

The following is an example of using `sfcnewfs(1M)` to create a file system in which multiple file data partitions and an update log partition are specified and the data area is not included in the representative partition.

```
# sfcnewfs -A -H mikan,karin,mango -S all -L /dev/sfdsk/gfs/rdsk/volume2 -D /dev/sfdsk/gfs/rdsk/volume3,/dev/sfdsk/gfs/rdsk/volume4 /dev/sfdsk/gfs/rdsk/volume1
```

Representative partition (meta-data) : /dev/sfdsk/gfs/rdsk/volume1
 Log partition : /dev/sfdsk/gfs/rdsk/volume2
 Data partition : /dev/sfdsk/gfs/rdsk/volume3, /dev/sfdsk/gfs/rdsk/volume4
 Shared hosts : mikan, karin, and mango
 Primary MDS : mikan
 Secondary MDS : karin
 MDS operational nodes : mikan, karin, and mango

Figure Separate file data and update log areas



Note

The file system access performance may be affected when the initialization processing of a **meta-data area** is done after MDS has been started. However, access is not affected if you specify the `-c` option for the `sfcnewfs(1M)` command to complete the initialization processing of the meta-data area.

20.2.2.3 Customizing a GFS shared file system

Parameters for customizing a file system are provided by `sfcnewfs(1M)`, which is used to create a GFS shared file systems. Typical parameters are described below.

Update log area size (-g *logsz*)

Specify the size of the **update log area** with a value from **5** to **100** megabytes. The default size of an update log area is of 1% of the file system size. If, however, 1% of the file system is less than **5** megabytes, **5** megabytes is set. Also, if the value is greater than **50** megabytes, **50** megabytes is set.

Meta-data area size (-a *metasz*)

Specify the size of the **meta-data area** in the representative partition. The default value is about 10% of the file system size. If the maximum **file data area** size is specified, about 10% of the maximum data area size is set.

Maximum data area size (-d *maxdsz*)

Specify the maximum total size of the data area when adding a **file data partition** as an expansion to a GFS shared file system. To add a file data partition, use `sfcadd(1M)`. If the user is planning to add a file data partition, specify this parameter. Up to **32** terabyte can be specified.

Maximum number of partitions (-l *maxvol*)

Specify the maximum number of partitions that may ever comprise this GFS shared file system. The types of partitions configuring the GFS Shared File System are the representative partition, **update log partition**, and file data partition. The value specified here is the total of the partitions of all types. The default value is **16**. Specifiable values are **1** to **32**.

20.2.2.4 Enabling the quota function

The quota function can be enabled immediately after file system creation. In this case, consistency processing is executed for the quota information when the first file system service starts without files. An advantage of this method is that completion of the consistency processing requires a minimal amount of time.



See

For details on enabling the quota function, see [Section 20.3.2, "Enabling the quota function."](#)

20.2.2.5 Setting MDS operational information

The following shows an example of creating a typical GFS Shared File System with MDS operation information specified to improve availability:

- Priority settings for MDS placement

Specify the priority of placing the primary MDS and secondary MDS using the shared host name in the **-H** option of the `sfcnewfs(1M)` command. The primary MDS and secondary MDS candidates are selected in the order the shared host names are specified. The primary MDS is ultimately determined based on the MDS placement range (described below) and operation status of actual shared nodes when starting the file system operation.

The following shows an example of creating a file system using `sfcnewfs(1M)` specifying mikan as the primary MDS candidate and karin as the secondary MDS candidate.

```
# sfcnewfs -H mikan,karin,mango,akebi /dev/sfdsk/gfs/rdisk/volume1

Representative partition (meta-data,log and data) : /dev/sfdsk/gfs/rdisk/volume1
Shared hosts   : mikan, karin, mango and akebi
Primary MDS   : mikan
Secondary MDS : karin
```

- Configuration that enables MDS placement on all nodes:

To enable MDS placement on all nodes, use the **-S** option of the `sfcnewfs` command.

The following shows an example of creating a file system using `sfcnewfs(1M)` when MDS is placed on all nodes:

```
# sfcnewfs -S all -H mikan,karin,mango,akebi /dev/sfdsk/gfs/rdisk/volume1

Representative partition (meta-data,log and data) : /dev/sfdsk/gfs/rdisk/volume1
Shared hosts   : mikan, karin, mango and akebi
Primary MDS   : mikan
Secondary MDS : karin
Host on which MDS can be placed: mikan, karin, mango and akebi
```

- Configuration that restricts the range of MDS placement:

To restrict the MDS placement range to specific shared hosts, use the **-S** option of the `sfcnewfs` command to specify the number of nodes.

The following shows an example of creating a file system using `sfcnewfs(1M)` with the placement range of MDS restricted to three nodes:

```
# sfcnewfs -S 3 -H mikan,karin,mango,akebi /dev/sfdsk/gfs/rdisk/volume1

Representative partition (meta-data,log and data) : /dev/sfdsk/gfs/rdisk/volume1
Shared hosts   : mikan, karin, mango and akebi
Primary MDS   : mikan
Secondary MDS : karin
Host on which MDS can be placed: mikan, karin, and mango
```

20.2.3 Setting /etc/vfstab

In the GFS Shared File System, it is necessary to code an entry that is to be mounted on /etc/vfstab of all the nodes that share a file system. Also, the information that is specified for the mount options of /etc/vfstab are used as the mount option when the mount command is executed. Other than the mount option, noauto can be specified for the mount options. Specify the noauto option when mounting is not to be performed when the node is started.

Examples of /etc/vfstab entries are shown below:

- When mounting in rw mode at startup of the node:

Parameter	Value
device to mount	/dev/sfdsk/gfs/dsk/volume1
device to fsck	/dev/sfdsk/gfs/rdisk/volume1
mount point	/mnt/fs1
FS type	sfdfs
fsck pass	-
mount at boot	no
mount options	rw

- When mounting is not performed at startup of the node

Parameter	Value
device to mount	/dev/sfdsk/gfs/dsk/volume1
device to fsck	/dev/sfdsk/gfs/rdisk/volume1
mount point	/mnt/fs1
FS type	sfdfs
fsck pass	-
mount at boot	no
mount options	rw, noauto

Table Mount options

Option	Description
closestync	All non-updated data in the relevant file is reflected when the file is finally closed.
intr nointr	Allows (intr) or prevents (nointr) a keyboard interruption to forcibly terminate a file operation queue process on the locked file system. intr is used as the default.
noatime	Does not update the file access time.
noforcedirectio forcedirectio	Mounts a file system with a forcible direct I/O specification (forcedirectio) or with a non-forcible direct I/O specification (nonforcedirectio). If forcedirectio is specified, access to the file system will be attempted with forcible direct I/O. If an access request does not satisfy the I/O requirements, normal access will be performed. Direct I/O refers to the direct data transmission of a file system between a user space and a disk, without passing through the data transmission buffer that is prepared in the kernel space. By default, normal access is performed with noforcedirectio.
nosuid	A file system will be mounted with setuid execution permission. Note, however, that if nosuid is specified, mounting will be executed with setid execution inhibited.
rw ro	Mounts in read/write mode (rw) or read only mode (ro). By default, rw is used.
noauto	Does not mount when a node is started. By default, mounting is performed when a node is started.



Note

Specify “-“ to fsck pass and “no” to mount at boot.

20.2.4 mount/umount/all nodes (u)mount

Use `mount(1M)` and `umount(1M)` commands to mount or unmount a GFS shared file system on a node. Also, use `sfcmntgl(1M)` and `sfcumntgl(1M)` to mount or unmount a GFS shared file system on all the nodes. These commands can be executed from any node on which the GFS shared file system can be used. As the extension of `mount` (all nodes mount), `sfcsd` and `MDS` will be started while `sfcsd` and `MDS` will be terminated as the extension of `umount` (all nodes unmount).



Note

If no entry is described in `/etc/vfstab`, mounting is disabled. Also, no mount options can be specified with the `mount` command. The contents of the mount options in `/etc/vfstab` are used as the mount options.

20.2.4.1 Mount



See

For details on `mount_sfcs(1M)`, see [Appendix F.1.5, "mount_sfcs\(1M\)."](#)

Perform mounting as shown below:

- When the mount point is specified:

```
# mount /mnt/fs1
Mount point: /mnt/fs1
```

- When a representative partition is specified:

```
# mount /dev/sfdsk/gfs/dsk/volume1
Representative partition: /dev/sfdsk/gfs/dsk/volume1
```

- When both the mount point and a representative partition are specified:

```
# mount /dev/sfdsk/gfs/dsk/volume1 /mnt/fs1
Representative partition: /dev/sfdsk/gfs/dsk/volume1
Mount point: /mnt/fs1
```

20.2.4.2 Mount of all nodes



See

For details on `sfcmntgl(1M)`, see [Appendix F.2.13, "sfcmntgl\(1M\)."](#)

Mount all nodes as shown below:

- When the mount point is specified:

```
# sfcmntgl /mnt/fs1
Mount point: /mnt/fs1
```

- When a representative partition is specified:

```
# sfcmntgl /dev/sfdsk/gfs/dsk/volume1
Representative partition: /dev/sfdsk/gfs/dsk/volume1
```

- When both the mount point and a representative partition are specified:

```
# sfcmntgl /dev/sfdsk/gfs/dsk/volume1 /mnt/fs1
Representative partition: /dev/sfdsk/gfs/dsk/volume1
Mount point: /mnt/fs1
```

If mounting fails for a node, the name of the node on which the failure occurred is reported.

20.2.4.3 Unmount



See

For details on `umount_sfdfs(1M)`, see [Appendix F.1.6, "umount_sfdfs\(1M\)."](#)

Unmounting must be performed as shown below:

- When the mount point is specified:

```
# umount /mnt/fs1
Mount point: /mnt/fs1
```

- When a representative partition is specified:

```
# umount /dev/sfdsk/gfs/dsk/volume1
Representative partition: /dev/sfdsk/gfs/dsk/volume1
```

20.2.4.4 Unmount of all nodes



See

For details on `sfcumntgl(1M)`, see [Appendix F.2.30, "sfcumntgl\(1M\)."](#)

- When the mount point is specified:

```
# sfcumntgl /mnt/fs1
Mount point: /mnt/fs1
```

- When a representative partition is specified:

```
# sfcumntgl /dev/sfdsk/gfs/dsk/volume1
```

```
Representative partition: /dev/sfdsk/gfs/dsk/volume1
```

If unmounting fails on a node, the name of the node on which the failure occurred is reported.

20.2.5 Checking file system status

sfcrcinfo(1M) can check if a GFS shared file system can be mounted.

The following example shows how to check the mount status of the GFS shared file system whose **representative partition** is /dev/sfdsk/gfs/rdsk/volume1 with sfcrcinfo(1M).

```
# sfcrcinfo -m /dev/sfdsk/gfs/rdsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID  MDS/AC  STATE  S-STATE  RID-1  RID-2  RID-N  hostname
  1  MDS(P)  run    -         0       0       0  prime01
  1  AC      run    -         0       0       0  prime01
  1  MDS(S)  wait   -         0       0       0  prime02
  1  AC      run    -         0       0       0  prime02
  1  MDS(I)  imag   -         0       0       0  prime03
  1  AC      run    -         0       0       0  prime03
```


20.2.6 Notes applying when the partitions of a created file system are used

20.2.6.1 The GFS Shared File System

To use a **partition** that is already in use by an existing GFS shared file system for a new GFS shared file system, delete the existing file system and then create the new file system.



For information about deleting a file system, see [Section 20.9, "Deleting."](#)

20.2.6.2 The GFS Local File System

To use a **partition** that is already in use by an existing GFS local file system for a new GFS shared file system, delete the existing GFS local file system and then create the new GFS shared file system.

To delete a GFS local file system, use `sfxadm(1M)`.

The following example shows how to delete the GFS local file system that includes `/dev/sfdsk/gfs/rdisk/volume1`.

```
# sfxadm -D /dev/sfdsk/gfs/rdisk/volume1
```



For details on `sfxadm(1M)`, see [Appendix B.2.2, "sfxadm\(1M\)."](#)

For data conversion, see [Section 24.2, "Migration from GFS Local File System."](#)

20.2.6.3 File systems other than GFS Local File System and GFS Shared File System

To use a partition that is in use by a file system other than the GFS Local File System and the GFS Shared File System, unmount the file system on all nodes and then delete in accordance with the procedure for that file system. Also, delete the definition in `/etc/vfstab` and then create the GFS shared file system.

20.3 Changes (in quota function status)

This section explains how to set the **quota function**. Set the quota function during mount operations. Only enabling or disabling of the quota is possible while a file system is unmounted.

20.3.1 Setting the quota function

This section explains how to set the quota function. Set the quota function as follows:

1. Create the quotas file
2. Set user limit values
3. Copy user limit values
4. Set user limit time

The quota function can only be set during mount operations.

20.3.1.1 Creating the quotas files

As a superuser, create a quotas file with touch(1) in the root directory of the GFS shared file system that uses the quota function. Set the quotas file access right with chmod(1) to rw-----. The following example shows a quotas file created in the mount point /sfcfs directory.

```
# touch /sfcfs/quotas
# chmod 0600 /sfcfs/quotas
```



For details on touch(1) and chmod(1), see touch(1) and chmod(1) of the “*Solaris X Reference Manual Collection*.”

For the procedure for restoring and creating a quotas file that is backed up, see [Section 22.6, "Backing up and Restoring User Limit Values of the quota function."](#)

20.3.1.2 Setting user limit values

To set user limit values, use sfcquota(1M). An example of setting a limit value for user1 is provided below.



For details on sfcquota(1M), see [Appendix F.2.6, "sfcquota\(1M\)."](#)

```
# sfcquota user1
```

When the above example is executed, the editor displays the contents shown below. Change the limit values whose initial values are 0 to the desired values.

```
fs /sfcfs  blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

blocks: Specify the limit value for the number of blocks (in kilobytes).

inodes: Specify the limit value for the number of files (including directories).

soft: Specify a soft limit value (for issuing a warning message).

hard: Specify a hard limit value (actual limit value).

The following example shows that the limit for the number of usable blocks is less than **5** megabytes (warning for **3** megabytes) and the limit for the number of creatable files is **99** (warning for **80**).

```
fs /sfdfs blocks (soft = 3072, hard = 5120) inodes (soft = 80, hard = 100)
```

20.3.1.3 Copying user limit values

To copy limit values set for user1 to another user, use the **-p** option of `sfcedquota(1M)`. The following example shows that how to copy a limit value set for user1 to user2 and user3.

```
# sfcedquota -p user1 user2 user3
      ----  -----
      |          |
      |          +-- User name of the copy destination
      |
      +-- User name of the copy source
```

20.3.1.4 Setting user limit time

To set the user limit time, use the **-t** option of `sfcedquota(1M)`. This setting is intended for all users other than superusers. Assume that the soft limit value of a user's limit value is less than or equal to the hard limit value. In this case, a limit time that is equal to or greater than the soft limit value is validated. A user can only perform allocation during this limit time until the maximum hard limit even if the soft limit value is exceeded. The default of the limit time is one week. An example of setting a user limit time is provided below.

```
# sfcedquota -t
```

When the above example is executed, the editor displays the contents listed below. Change the limit times whose initial values are **0**.

```
fs /sfdfs blocks time limit = 0 (default),files time limit = 0 (default)
```

Default "**0**" indicates one week. Specify the unit following a number. The usable units are **month**, **week**, **day**, **hour**, **min**, and **sec**.

The following example shows that the limit time for the number of blocks is set to two weeks and the limit time for the number of files is set to 16 days:

```
fs /sfdfs blocks time limit = 2 week, files time limit = 16 day
```

20.3.2 Enabling the quota function

To enable the quota function, use `sfcquotaon(1M)`. Because the quota function is recognized as an MDS option, the quota function is enabled.

The example provided below shows that the quota function of the GFS shared file system whose mount point is `/sfdfs` is enabled. Use `sfcinfo(1M)` to check the quota function status

```
# sfcquotaon /sfdfs
# sfcinfo -q /sfdfs
FSID special                quota mount
  2 /dev/sfdsk/gfs/dsk/volume1  on   /sfdfs
```



Note

After executing `sfcquotaon(1M)`, the consistency processing for the quota information is executed internally. The limit values set by the quota function are not validated during the consistency processing for this quota information. Because the consistency processing for the quota information requires longer processing time in proportion to the number of files, it is recommended that the quota function be enabled immediately after file system creation and not be disabled if at all possible.

The GFS Shared File System does not enable the quota function even if `rq` or `quota` is specified in the `mntopts` field of `/etc/vfstab`.

As with the UFS file system, `mount(1M)` enables each node to check the quota function.

20.3.3 Disabling the quota function

To disable the quota function, use `sfcquotaoff(1M)`.

The following example show disabling of the quota function for the GFS shared file system whose mount point is `/sfdfs`.

```
# sfcquotaoff /sfdfs
```



Note

After executing `sfcquotaoff(1M)`, assume that the file system is updated and `sfcquotaon(1M)` is executed again. In this case, the consistency processing for the quota information is executed internally. The limit values set by the quota function are not validated during the consistency processing for this quota information. Because the consistency processing for the quota information requires greater processing time in proportion to the number of files, once the quota function is enabled, it is recommended that the quota function not be disabled if at all possible.

20.4 Change (file system attributes)

20.4.1 Unmount

To modify file system attributes of a GFS shared file system, unmount the file system.

20.4.2 Change the file system attributes

20.4.2.1 Change the update log area size

To change the size of an **update log area**, use `sfcadm(1M)`.

The following example shows how to change the update log area size of the GFS shared file system whose **representative partition** is `/dev/sfdsk/gfs/rdsk/volume1` to 100 megabytes.



For details on `sfcadm(1M)`, see [Appendix F.2.2, "sfcadm\(1M\)."](#)

```
# sfcadm -L 100 /dev/sfdsk/gfs/rdsk/volume1
```

20.4.2.2 Tuning file system

To tune a GFS shared file system, use `sfctune(1M)`.

In the following example, the representative partition is `/dev/sfdsk/gfs/rdsk/volume1`. The example shows how to modify the percentage of the area usually allowed for use by a user of the GFS shared file system to five %.

```
# sfctune -m 5 /dev/sfdsk/gfs/rdsk/volume1/GLOBAL03
```



For details of `sfctune(1M)`, see [Appendix F.2.29, "sfctune\(1M\)."](#)

20.4.3 Mount

After changing the GFS shared file system attributes, mount the file system.

20.5 Change (partition configuration)

20.5.1 Changing an unmounted file system

This section explains the procedure for adding partitions to an unmounted GFS shared file system.

20.5.1.1 Unmount

To change a GFS shared file system, unmount the file system.



Before adding partitions to a file system that did not stop normally, correct the file system status with `fsck_sfscfs(1M)`.

20.5.1.2 Setting shared disk

The **partition** in which a GFS shared file system is created and operated must be on a **shared disk** recognized by **PRIMECLUSTER**. To use a GDS logical volume as a target device, it must be active.



Check that the disk is set up as shared for simultaneous access.

If a partition has set up the disk as shared by automatic detection of shared disks, it is already set as a shared disk for simultaneous access and no further action is needed.

When registering a GDS logical volume with a class, set the disk type as shared and the scope to include all the nodes that share the GFS Shared File System.



For GDS disk class operations, see the applicable items under "Operation using Global Disk Services Management View" in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

For GDS logical volume operations, see the applicable items under "Operation using Global Disk Services Management View" in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

20.5.1.3 Change the partition configuration

This section explains the procedure for adding a new **file data partition** to an existing GFS shared file system.

20.5.1.3.1 Adding a file data partition

A new **file data partition** can be added to an existing GFS shared file system without saving and restoring data.

To add a file data partition, use `sfcadd(1M)`.

How to add a file data partition (`/dev/sfdsk/gfs/rdisk/volume1`) to an existing file system (`/dev/sfdsk/gfs/rdisk/volume2`) by `sfcadd(1M)` is shown below.

[See](#)

For details on `sfcadd(1M)` options, see [Appendix F.2.1, "sfcadd\(1M\)."](#)

- Check the configuration of the current file system.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special                size  Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)  407144  META  ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)  3761416  DATA  ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    5120  LOG   ----
```

- Add a file data partition as file data area.

```
# sfcadd -D /dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/gfs/rdsk/volume1
```

- Confirm that the file data partition has been added.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special                size  Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)  407144  META  ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)  3761416  DATA  ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    5120  LOG   ----
  1 /dev/sfdsk/gfs/dsk/volume2(3fa0082)   65512  DATA  ----
```

**Note**

A file data partition addition is not possible if doing so would exceed the maximum number of partitions (`-l maxvol`) for the file system, as specified by `sfcnewfs(1M)`.

When adding file data partitions, confirm that the new total size of the file system does not exceed the maximum file data area size (`-d maxdsz`) for the file system, as specified by `sfcnewfs(1M)`.

20.5.1.3.2 Adding a meta-data partition

Use `sfcadd(1M)` to add partitions to the meta-data area.

The procedure shown below adds a partition (`/dev/sfdsk/gfs/rdsk/volume2`) to an existing file system (`/dev/sfdsk/gfs/rdsk/volume1`) as a meta-data area using `sfcadd(1M)`.

[See](#)

For details on the `sfcadd(1M)` options, see [Appendix F.2.1, "sfcadd\(1M\)."](#)

- Check the current file system configuration.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special                size  Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)  407144  META  ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)  3761416  DATA  ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    5120  LOG   ----
```

- Add a partition as a meta-data area.

```
# sfcadd -M /dev/sfdsk/gfs/dsk/volume2 /dev/sfdsk/gfs/dsk/volume1
```

- Check that a partition has been added.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special size Type mount
1 /dev/sfdsk/gfs/dsk/volume1(3fa0081) 407144 META -----
1 /dev/sfdsk/gfs/dsk/volume1(3fa0081) 3761416 DATA -----
1 /dev/sfdsk/gfs/dsk/volume1(3fa0081) 5120 LOG -----
1 /dev/sfdsk/gfs/dsk/volume1(3fa008c) 65512 META -----
```



Note

Note the following limitation for meta-data partition addition. Addition is not possible if doing so would exceed the maximum number of partitions (-I **maxvol**) in the file system, as specified by **sfcnewfs(1M)**.

20.5.1.4 Mount

After changing the GFS shared file system partition configuration, mount the file system.

20.5.2 Changing while mounted status

A new **file data partition** can be added to an existing mounted GFS shared file system as a **file data** area without backing-up and restoring data.

20.5.2.1 Setting shared disk

The **partition** in which the GFS Shared File System is created and operated must be on a **shared disk** recognized by **PRIMECLUSTER**. To use a GDS logical volume as a target device, it must be active.



Note

Check that the disk is set up as shared for simultaneous access. If a partition has set up the disk as shared by automatic detection of shared disks, it is already set as a shared disk for simultaneous access and no further action is needed. When registering a GDS logical volume with a class, set the disk type as shared and the scope to include all the nodes that share the GFS Shared File System.



See

For GDS disk class operations, see the applicable items under "Operation using Global Disk Services Management View" in the *"PRIMECLUSTER Global Disk Services Configuration and Administration Guide."*

For GDS logical volume operations, see the applicable items under "Operation using Global Disk Services Management View" in the *"PRIMECLUSTER Global Disk Services Configuration and Administration Guide."*

20.5.2.2 File system change

The following procedures describe how to add a new file data or **meta-data partition** to an existing GFS shared file system.

20.5.2.2.1 Adding a file data partition

A new **file data partition** can be added to an existing mounted GFS shared file system, use `sfcadd(1M)`.

How to add a file data partition (`/dev/sfdsk/gfs/rdsk/volume2`) to an existing file system (`/dev/sfdsk/gfs/rdsk/volume1`) using `sfcadd(1M)` is shown below.



See

For details on the `sfcadd(1M)` option, see [Appendix F.2.1, "sfcadd\(1M\)."](#) Notes are the same for mounted and unmounted file systems. See Notes in [Section 20.5.1.3.1, "Adding a data partitions."](#)

- Check the configuration of the current file system.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special                                size Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    407144 META  ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    3761416 DATA ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)     5120 LOG   ----
```

- Add a file data partition as file data area.

```
# sfcadd -D /dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/gfs/rdsk/volume1
```

- Confirm that the file data partition has been added.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special                                size Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    407144 META  ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    3761416 DATA ----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)     5120 LOG   ----
  1 /dev/sfdsk/gfs/rdsk/volume2(3fa00c1)    65512 DATA  ----
```

20.5.2.2.2 Adding a meta-data partition

Use `sfcadd(1M)` to add partitions to the meta-data area while the file system is mounted.

The procedures for adding a meta partition (`/dev/sfdsk/gfs/rdsk/volume2`) to an existing file system (`/dev/sfdsk/gfs/rdsk/volume1`) are shown below.



See

For details on the `sfcadd(1M)` options, see [Appendix F.2.1, "sfcadd\(1M\)."](#)

- Check the current file system configuration.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special                size  Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)  407144  META  -----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081) 3761416  DATA  -----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)   5120   LOG   -----
```

- Add a meta-partition

```
# sfcadd -M /dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/gfs/rdsk/volume1
```

- Check that a meta-data partition has been added.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special                size  Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)  407144  META  -----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081) 3761416  DATA  -----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)   5120   LOG   -----
  1 /dev/sfdsk/gfs/dsk/volume2(3fa008c)  65512  META  -----
```



Note

Note the following restriction for meta-data partition addition.
A meta-data partition addition is not possible if doing so would exceed the maximum number of partitions (`-l maxvol`) for the file system, as specified by `sfcnewfs(1M).(1M)`.

20.6 Change (shared node information)

20.6.1 Unmount

To change the GFS Shared File System, unmount the file system.

20.6.2 Changing the management partition information

When a new-shared node is to be added, register the node information in the partition by first executing **sfcsetup -a** on the node to be added.



See

For details, see [Chapter 16, "Management Partition."](#)

20.6.3 Changing the shared node information

The GFS Shared File System stores information about **nodes** that share the file system in each partition.

The node information mainly contains the following information:

- Host ID
- Host name

To add and delete node information, use `sfcnode(1M)`.



See

For details on the `sfcnode(1M)` options, see [Appendix F.2.15, "sfcnode\(1M\)."](#)

How to add and delete node information using `sfcnode(1M)` is shown below.

20.6.3.1 Adding node information

- Check the configuration of the current file system.

```
# sfcinfo -n /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID  hosted  status  hostname
  1  80804266  STOP   mango
  1  807e7937  STOP   karin
```

- Add the **node** information.

```
# sfcnode -a mikan /dev/sfdsk/gfs/rdsk/volume1
```

- Confirm that the node information has been added.

```
# sfcinfo -n /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID  hostID   status  hostname
  1  80804266  STOP   mango
  1  807e7937  STOP   karin
  1  80807658  STOP   mikan
```

Use [sfcadm\(1M\)](#) to enable MDS operation on an added node. For node changes that enable MDS operation, see [Section 20.8.2, "Changing the MDS operational information."](#)

20.6.3.2 Deleting node information

- Check the configuration of the current file system.

```
# sfcinfo -n /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID  hostID  status  hostname
  1   80804266  STOP   mango
  1   807e7937  STOP   karin
  1   80807658  STOP   mikan
```

- Delete the node information.

```
# sfcnode -d -h mikan /dev/sfdsk/gfs/rdisk/volume1
```

- Confirm that the node information has been deleted.

```
# sfcinfo -n /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID  hosted  status  hostname
  1   80804266  STOP   mango
  1   807e7937  STOP   karin
```



Note

Node information about the primary MDS node and secondary MDS node cannot be deleted. Use [sfcadm\(1M\)](#) to change the primary MDS node or secondary MDS node to another node, then delete the related information.

20.6.4 Mount

After the GFS Shared File System has been modified, mount the file system

20.7 Change (re-creating a file system)

20.7.1 Unmount

To re-create the GFS Shared File System, unmount the file system.

20.7.2 Re-creating the file system

To re-create the file system, use `sfcnewfs(1M)`. Using `sfcnewfs(1M)` with the `-Z` option enables the user to mount the file system.

The following is an example of using `sfcnewfs(1M)` to re-create a file system with a single partition.

```
# sfcnewfs -Z -H mikan,karin,mango /dev/sfdsk/gfs/rdisk/volume1
```

```
Representative partition (meta-data, log, and data) : /dev/sfdsk/gfs/rdisk/volume1
```

```
Shared hosts : mikan,karin,mango
```

```
Primary MDS : mikan
```

```
Secondary MDS : karin
```



Re-creating a file system means creating a new file system. Make backup copies because the data in the previous file system is deleted.

20.7.3 Mount

After the GFS Shared File System has been re-created, mount the file system.

20.8 Change (MDS operational information)

20.8.1 Unmount

To modify the GFS Shared File System, unmount the file system.

20.8.2 Changing the MDS operational information

In the GFS Shared File System, information about all nodes that operate **MDS** is retained.

- Whether MDS placement on all nodes is set
- Number of nodes on which MDS is placed
- Priority of nodes on which to place MDS as primary or secondary MDS

Use `sfcadm(1M)` to change information about nodes on which MDS is placed.



See

For details on `sfcadm(1M)` options, see [Appendix F.2.2, "sfcadm\(1M\)."](#)

The following shows how to change information about nodes on which MDS is placed using `sfcadm(1M)`.

20.8.2.1 Enabling MDS placement on all nodes

- Check the current file system configuration. Two nodes (mikan and karin) are currently in the MDS placement range.

```
# sfcinfo -n /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID  hostID  status  hostname
  1   80807658  STOP   mikan
  1   807e7937  STOP   karin
  1   80804266  STOP   mango
  1   80b6324a  STOP   ringo

# sfcinfo -c /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
node=mikan,karin
```

- Change the MDS placement range to all shared hosts.

```
# sfcadm -c all /dev/sfdsk/gfs/dsk/volume1
```

- Confirm that the MDS placement range has been changed. "**all**" is now displayed to confirm that the MDS placement range includes all nodes.

```
# sfcinfo -c /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
node=mikan,karin,mango,ringo
mdsnode=all
```

If all nodes are set in the MDS placement range, an added node is automatically integrated into the range when a shared node is subsequently added using [sfcnnode\(1M\)](#).

20.8.2.2 Changing the MDS placement range

Specify the MDS placement range using the number of nodes from the first node in the shared host list specified at file system creation.

- Check the current file system configuration. “all” is currently displayed to confirm that the MDS placement range includes all nodes.

```
# sfcinfo -n /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID  hostID  status  hostname
  1  80807658  STOP   mikan
  1  807e7937  STOP   karin
  1  80804266  STOP   mango
  1  80b6324a  STOP   ringo

# sfcinfo -c /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
node=mikan,karin,mango,ringo
mdsnode=all
```

- Change the MDS placement range from all nodes to three nodes (mikan, karin, and mango).

```
# sfcadm -c 3 /dev/sfdsk/gfs/dsk/volume1
```

- Confirm that the MDS placement range has been changed.

```
# sfcinfo -c /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
node=mikan,karin,mango
```

20.8.2.3 Changing the MDS placement priority

- Check the current file system configuration.

```
# sfcinfo -n /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID  hostID  status  hostname
  1  80807658  STOP   mikan
  1  807e7937  STOP   karin
  1  80804266  STOP   mango
  1  80b6324a  STOP   ringo

# sfcinfo -c /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
node=mikan,karin,mango
```

- Change the priorities of nodes on which MDS is placed.

```
# sfcadm -g mango,ringo,mikan /dev/sfdsk/gfs/dsk/volume1
```

- Confirm that the MDS placement range has been changed.

```
# sfcinfo -c /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
node=mango,ringo,mikan
```

20.8.3 Mount

After the GFS Shared File System has been modified, mount the file system.

20.9 Deleting

20.9.1 Unmount

To delete a GFS Shared file system, unmount the file system.

20.9.2 Deleting the file system

To delete a file system that is not being used in the GFS Shared File System, use `sfcadm(1M)` with the `-D` option specified.



See

For details on the `sfcadm(1M)` options, see [Appendix F.2.2, "sfcadm\(1M\)."](#)

How to delete a file system using `sfcadm(1M)` is shown below.

- Check the current file system information.

```
# sfcinfo -a
/dev/sfdsk/gfs/dsk/volume1:
FSID special                                size  Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    407144 META  -----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    3761416 DATA  -----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)     5120 LOG  -----
/dev/sfdsk/gfs/dsk/volume2:
FSID special                                size  Type  mount
  2 /dev/sfdsk/gfs/rdsk/volume2(3fa0082)    51394 META  -----
  2 /dev/sfdsk/gfs/rdsk/volume3(3fa0083)    523776 DATA  -----
  2 /dev/sfdsk/gfs/rdsk/volume4(3fa0084)     5237 LOG  -----
```

- Delete the file system whose representative partition is `/dev/sfdsk/gfs/rdsk/volume2`.

```
# sfcadm -D /dev/sfdsk/gfs/rdsk/volume2
```

- Confirm that the file system has been deleted.

```
# sfcinfo -a
/dev/sfdsk/gfs/dsk/volume1:
FSID special                                size  Type  mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    407144 META  -----
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    3761416 DATA  -----
  1 /dev/sfdsk/gfs/rdsk/volume1(3fa0081)     5120 LOG  -----
```

Chapter 21 File System Management

This chapter describes the procedures for managing the GFS Shared File System using basic commands.

21.1 File System Management Commands

For each file system, there are two different types of management commands, namely, the general purpose management commands and the system specific management commands. The general purpose management commands that provided by Solaris OE are used for basic functions and the system specific commands are called by specifying options or file system types.

The GFS Shared File System features an exclusive management commands that act as the general purpose management command and the system specific management command for the GFS Shared File System's original functions.

Specify "**sfdfs**" for **-F** option as a file system type when using the general purpose management command to a GFS shared file system. If you omit the **-F** option, a search is made of those entries matched with "*special*" or "*mount_point*" in */etc/vfstab*," after which the registered file system types are automatically selected.

General purpose file system management commands

Command	Description
df(1M)	Displays the number of free disk blocks, number of files, etc.
fsck(1M)	Checks a file system for consistency and repairs it.
fstyp(1M)	Determines the file system type.
mkfs(1M)	Creates a file system.
mount(1M)	Mounts a file system.
umount(1M)	Unmounts a file system.



See

For details of the general-purpose management commands, see each command's page of "*Solaris X Reference Manual*."

GFS Shared File System management commands

Command	Description
df_sfdfs(1M)	Displays the number of free disk blocks, number of files, etc.
fsck_sfdfs(1M)	Checks a file system for consistency and repairs it.
fstyp_sfdfs(1M)	Determines the file system type.
mkfs_sfdfs(1M)	Creates a file system.
mount_sfdfs(1M)	Mounts the GFS Shared File System on the self-node.
umount_sfdfs(1M)	Unmounts the GFS Shared File System from the self-node.

GFS Shared File System specific management commands

Command	Description
sfcadd(1M)	Adds file data partitions.
sfcadm(1M)	Changes partition information settings.
sfcddr(1M)	Restores backup data and checks device restoring.
sfcdu(1M)	Summarizes disk usage.
sfcdump(1M)	Dumps a GFS shared file system.
sfcquota(1M)	Edits quota limits for each user.
sfcfrmstart(1M)	Starts FsRM on the self-node.
sfcfrmstop(1M)	Stops FsRM on the self-node.
sfcgetconf(1M)	Backs up the management partition.
sfchost(1M)	Displays and changes a connection host name with MDS.
sfcinfo(1M)	Displays partition information.
sfcmbkup(1M)	Generates a shell script for backing up.
sfcmrstr(1M)	Generates a shell script for restoring.
sfcmntgl(1M)	Mounts the GFS Shared File System on all nodes.
sfcnewfs(1M)	Builds the GFS Shared File System (front end of mkfs_sfcs).
sfcnode(1M)	Adds, deletes, and alters node configuration information.
sfcproxybreak(1M)	Cancel the relationship for high speed backup and restoration.
sfcproxyinfo(1M)	Displays information about the relationship for high speed backup and restoration.
sfcproxyjoin(1M)	Sets up the relationship for high speed backup and restoration.
sfcproxyop(1M)	Executes operations for high speed backup and restoration.
sfcproxyrestore(1M)	Restores a file system at high speed.
sfcquota(1M)	Displays the quota limit value and uses amount for each user.
sfcquotadm(1M)	Manages quotas files.
sfcquotaon(1M)	Validates the quota function.
sfcquotaoff(1M)	Invalidates the quota function.
sfcprepquota(1M)	Displays the quota function limits and use amounts of all users.
sfcrestore(1M)	Restores from dump created with sfcdump.
sfcrcinfo(1M)	Displays file system information.
sfcsetup(1M)	Initializes the management partition; adds, deletes, and displays node information; displays path of the management partition.
sfcstat(1M)	Reports statistics on the GFS Shared File Systems.
sfc tune(1M)	Tunes the GFS Shared File System.
sfcumntgl(1M)	Unmounts the GFS Shared File System from all nodes.

Original commands for general users of the GFS Shared File System original functions are listed below.

GFS Shared File System management commands

Command	Description
sfccp(1)	Copies files with extent attribute information.
sfccpio(1)	Extracts or restores files from an archive with extent attribute information.
sfcgetext(1)	Displays extent attribute information.
sfcgetfattr(1)	Reports file extension attribute information of a file.
sfcmv(1)	Moves files with extent attribute information.
sfcsetext(1)	Specifies extent attribute information.
sfcsettime(1)	Sets or deletes ignore update time attribute information.
sfcsetvolume(1)	Sets or deletes file data allocation attribute information.

21.2 Checking a File System for Consistency and Repairing It

If a file system is damaged and consistency is lost, for example following automatic error recovery after a primary MDS fails, the file system must be checked and repaired using `fsck_sfdfs(1M)`.

The GFS Shared File System provides the **update log** function to ensure high-speed recovery if an error such as a system failure occurs. If the update log function has been enabled, the file system can be repaired quickly regardless of the size of the file system. This is possible because of **update log replay**, which updates the un-updated portion of the **meta-data** located in the update log area.

By default or when "**-o log**" is specified, `fsck` repairs the file system by replaying the update log.

If the update log data has been physically damaged, the `fsck` command does not execute update log replay, but instead automatically performs a full check on the file system. To meet the need for resuming system operation immediately, an option "**-o elog**" that prohibits file system repair without update log replay is provided. If this option is specified, `fsck(1M)` terminates immediately without performing a check or recovery if the update log data is physically damaged. In this event, the file system cannot be mounted unless a check and repair are performed using `fsck`. Specify the "**-o nolog**" option to perform a full check on the file system without update log replay, repair the file system, and mount the file system.

The following example repairs a file system with log replay.

```
# fsck -F sfdfs /dev/sfdsk/gfs/rdsk/volume1
```

The following example performs a full check on the file system and repairs it without log replay.

```
# fsck -F sfdfs -o nolog /dev/sfdsk/gfs/rdsk/volume1
```

The following example shows how to specify one raw partition as a work disk to check a full file system.

```
# fsck -F sfdfs -o nolog,work=/dev/rdsk/c2t3d0s1 /dev/sfdsk/gfs/rdsk/volume1
```



See

For details on the `fsck_sfdfs(1M)` options, see [Appendix F.1.2, "fsck_sfdfs\(1M\)."](#)

21.2.1 Action to be taken if `fsck` terminates abnormally

The following describes the action to be taken if `fsck_sfdfs(1M)` terminates abnormally.

21.2.1.1 Memory allocation request error

The following message is displayed if a request to allocate memory for the internal table for `fsck_sfdfs(1M)` fails.

```
Can't allocate memory for <xxxx>
xxxx: name-of-internal-table-for-fsck dup table, bta dup table, etc.
```

Add a swap area.

21.2.1.2 File system configuration information acquisition failure

The following message is output if acquisition of the file system configuration information fails.

```
fsck: ERROR: cannot connect to sfcfmd.
```



See

For details, see [Section 16, "Management Partition,"](#) and restore.

21.2.1.3 File system partition configuration data error

This message indicates that the command terminated abnormally because a mismatch was found in the **partition configuration data** for the GFS shared file system partitions.

```
Can't open <device-name>: <errno>
Can't get the data of partition <device-name>
setup: Volume information error
setup: fsck quit due to unrecoverable error!
```

Use `sfcadm(1M)` to recover the partition information.

For a single partition, perform the following operation:

```
# sfcadm -r -a <device-name>
```

For a multi-partition configuration, use `sfcinfo(1M)` to check the partitions and then use `sfcadm(1M)` to perform recovery using the **representative partition**.

```
# sfcinfo <device-name>
# sfcadm -r <partition-used-for-recovery device-name>
```

21.2.1.4 Node addition not performed

The following message is displayed if a **node** cannot be added.

```
No node volume entry for localhost, file system access forbidden!
```

Use `sfcnode(1M)` to add the node.

21.2.1.5 Irreparable file system destruction

The following message indicates that the file system is irreparable because of the GFS Shared File System data has been destroyed.

```
No Meta volume information available!
No Log volume information available!
Only found <num> data volume(s), total <num> in superblock!
The data volume <device-name>; is not found!
setup_data: fs_dsize < sum(dvol->sdata)!
```

The file system cannot be repaired. If a backup of the file system is available, recover the file system from the backup.

21.2.1.6 The amount of work disks is too small

The following message indicates that the amount of work disks for fsck is too small.

```
fsck workarea <size to add> bytes size short. need <required total size> bytes.
```

Please specify larger work disks or additional work disks as the amount of work disks for fsck is larger than insufficient size by size to add and larger than required total size, and then re-execute the fsck command.

21.3 File Extension Attribute Information

This section describes the extent related commands for general users provided by the GFS Shared File System.

The following table lists the extent related commands provided. For details on each command, refer to [Appendix F, "Reference Manual."](#)

The extent related commands

Command	Description
sfccp	Copies files with extent attribute information .
sfccpio	Extracts or restores a file from an archive with file extension attribute information.
sfcgetext	Displays extent attribute information.
sfcgetfattr	Reports file extension attribute information of a file.
sfcmv	Moves files with file extension attribute information.
sfcsetext	Specifies extent attribute information.
sfcsettime	Sets or deletes ignore update time attribute information.
sfcsetvolume	Sets or deletes file data allocation attribute information.

The extent attribute information of the GFS Shared File System is compatible with that supported by the GFS Local File System.

The commands above can be used to copy, move, save, and restore files with the extent attribute information between the GFS Shared File System and the GFS Local File System.



Note

File extension attributes other than the extent attribute information are not maintained when transferring or restoring to a GFS local file system.

21.3.1 Setting the Extent Attribute Information

The GFS Shared File System has functions that allocate a data area beforehand to a file, extend a data area in a certain proportion, and guarantee data area continuity. The information that enables these functions is called **extent attribute information**. These functions increase performance, by ensuring data area allocation and data area continuity.

Use `sfcsetext(1)` to set the extent attribute information. However, you cannot use it to set this information for an existing file. To set the extent attribute information in an existing file, execute `sfcsetext(1)` with an alias, copy the data with `sfccp(1)` and change the name of alias to the original with `sfcmv(1)`.

When an extent attribute information is set for a directory, it is automatically reflected in the files and directories created under the directory.

An example for using `sfcsetext(1)` is given below:

```
% sfcsetext -a 100,25 -f contig /mnt/file
```



Note

The extent attribute information and the file allocation property cannot be set in a single file at the same time.



For details on the sfcsetext(1) options, see [Appendix F.3.6, "sfcsetext\(1\)."](#)

21.3.2 Reference of the Extent Attribute Information

Use sfcgetext(1) or sfcgetattr(1) to reference the extent attribute information of a file set by sfcsetext(1). This command can also be used for files for which the **extent attribute information** has not been set.

An example for using sfcgetext(1) is given below:

% sfcgetext -H /mnt/file					
Blocksize(KB)	init-blocks	inc-blocks	flags	filename	
8	100	25	contig	/mnt/file	



For details on sfcgetext(1), see in [Appendix F.3.3, "sfcgetext\(1\)."](#) For details on sfcgetattr(1), see in [Appendix F.3.4, "sfcsetattr\(1\)."](#)

21.3.3 Ignore Update Time Attribute Information

A file can be set so that the access time/modify time should not be updated when the file is accessed /modified, reducing access load on a file system by reducing the frequency of meta-data modification.

Use sfcsettime(1) to set ignore update time attribute information. When ignore update time attribute information is set for a directory, it is automatically reflected for the files and directories created under such a directory.

Examples of sfcsettime(1) application are shown below.

Setup to suppress modification of a file's last modified time.

```
% sfcsettime -a /mnt/file
```

Setup to suppress modification of the last access time for files to be created in a specified directory.

```
% sfcsettime -a /mnt/dir
```



For details on sfcsettime(1), see [Appendix F.3.7, "sfcsettime\(1\)."](#)

The mount option of "noatime" can be used when file access time modification is not needed for an entire file system.

21.3.4 File Data Allocation Attribute Information

You can specify the allocation of specific **file data** from specific **file data partitions**. This is effective when you want to specify disk devices in a unit of directory in a file system composed of several file data partitions.

Use `sfcsetvolume(1)` to set file data allocation information. When file data allocation attribute information is set for a directory, that allocation information is automatically reflected for the files and directories created under the directory.

There are forced and suggested specifications in file data allocation attribute information setup. In forced specification, an error (**ENOSPC**) occurs when there's insufficient space on a file data partition that is specified when new data allocation is performed, even if the free space exists in other file data partitions. In the suggested specification, data allocation is done from other file data partitions when there's insufficient area in a specified file data partition.



For details on `sfcsetvolume(1)`, see [Appendix F.3.8, "sfcsetvolume\(1\)."](#)

21.3.5 Reference of the File Extension Attribute Information

The file extension property information set with `sfcsettext(1)`, `sfcsettime(1)`, and `sfcsetvolume(1)` can be accessed with `sfcgetfattr(1)`. `sfcgetfattr(1)` can be used for files without file extension attribute information settings.

An example of `sfcgetfattr(1)` application is shown below.

% sfcgetfattr -A /mnt/file						
Blocksize(KB)	init-blocks	inc-blocks	flags		filename	
8	100	25	contig		/mnt/file	



For details on `sfcgetfattr(1)`, see [Appendix F.3.4, "sfcgetfattr\(1\)."](#)

21.3.6 Command for Reflecting File Extension Attribute Information

The file extension attribute information is exclusive data for the GFS Shared File System, therefore, turnout, restoration and duplication of the file extension attribute information are not possible with the standard commands such as `tar(1)`, `cpio(1)`, `cp(1)` and `mv(1)`. The standard commands just carry out their normal operations.

To perform file turnout, restoration and duplication, use `sfccpio(1)`, `sfccp(1)`, and `sfcmv(1)` to reflect the file extension attribute information. A command equivalent to `tar(1)` is not supported.



For details on each command, refer to [Appendix F, "Reference Manual."](#)

21.4 quota function

This section explains the quota function for limiting the use of the file system.

21.4.1 Quota Function

The quota function is used to limit the total number of files or blocks that can be created by a general user. This total number can be limited for each user.

There are two types of limit values: soft limit value and hard limit value. The soft limit value is used to output a warning message to the user if the number of files or blocks exceeds the limit. The hard limit value is the limit where further files cannot be created or data added. In addition, if a specific interval (time limit) has elapsed after the number of files or blocks has exceeded the soft limit value, file creation or data addition is disabled.

21.4.2 Quota Management Method for the GFS Shared File System

To manage the limit values, number of usage files, and number of usage blocks for each general user, the UFS quota function uses a quotas file in the root directory of the file system. GFS manages information using a quota file that is extended compared with UFS and the quota file is hidden within the file system. The GFS Shared File System uses an external quotas file in addition to the above quota file which is called the internal quota file. The external quotas file is used to manage the limit values for each general user. The internal quotas file is used to manage the limit values, number of usage files, and number of usage blocks for each general user.

Using two files to manage the usage of general users has the following benefits:

- Even if a command such as the `rm(1)` command deletes the external quotas file by mistake, the internal quotas file remains. As a result, quota data does not have to be set up again or matched. In addition, the external quotas file can easily be restored from the internal quotas file.
- Because the external quotas file manages only the limit values, the number of usage files or blocks is not changed if the external quotas file is backed up or restored during operation. As a result, quota data does not have to be matched.
- Update data in the internal quotas file is written to the update log area. As a result, the integrity of the file system can be quickly restored in the event of a system failure.

21.4.3 Quota Function Commands

The table below lists the quota related commands provided by the GFS Shared File System.

For details on setting up quotas, see [Section 20.3 "Change \(in quota function status\)"](#) and for information on each command, see [Appendix F, "Reference Manual."](#)

The quota related commands

Command	Description
<code>sfcquota</code>	Edits quota limits for each user.
<code>sfcquota</code>	Displays the quota limit value and use amount for each user.
<code>sfcquotadm</code>	Manages the quotas file
<code>sfcquotaon</code>	Validates the quota function.
<code>sfcquotaoff</code>	Invalidates the quota function.
<code>sfcprepquota</code>	Displays the quota function limit values and amount used for all users.

21.4.4 Checking the usage conditions for user

- Checking the usage conditions for each user

Execute the `sfcquota(1M)` command to check the usage conditions for each user. In the following example, the usage conditions are checked for user1.

```
# sfcquota -v user1
Disk quotas for user1 (uid 1000):
Filesystem      usage  quota  limit  timeleft  files  quota  limit  timeleft
/sfcs           2258  3072  5120             71    80   100
```

- Checking the usage conditions in units of file systems

Execute the `sfcprepquota(1M)` command to check the usage conditions in units of file systems. In the following example, the usage conditions are checked for mount point /sfcs.

```
# sfcprepquota -v /sfcs
/dev/sfdsk/gfs/rdsk/volume1 (/sfcs):
          Block limits
User      used  soft  hard  timeleft  used  soft  hard  timeleft
user1    --  2258  3072  5120             71    80   100
user2    --  1345  3072  5120             38    80   100
user3    ++  3424  3072  5120    7.0 days    81    80   100    7.0 days
:
```



Note

The number of blocks calculated in the quota function is that of the data area and directories. The number is equal to or less than the value obtained by summing up the numbers of blocks displayed by `ls(1)`, `du(1)`, etc. for each user.

21.4.5 Notes on Operation of the Quota Function

21.4.5.1 Matching quota data

To enable the quota function for the first time, consistency must be established between the number of user files and the number of blocks used. The processing time required to match quota data depends on the number of files.

A log of the internal quotas file is collected if the quota function is enable, therefore, quota data does not have to be matched even in the event of a system failure.

If the quota function is disabled and the system subsequently updated for any reason (e.g., file creation, deletion), the numbers of files and blocks used by a user do not match the values in the internal quotas file, therefore, additional time is required to match the data if the quota function is enabled again.

Specifically, the `sfcquotaon(1M)` or `mount(1M)` command takes additional time to complete, therefore, the quota function must not be disabled if possible once it has been enabled.

21.4.5.2 Messages

The kernel Information messages relating to the quota function are assigned message numbers from 0142 to 0147. For details on the kernel information messages, see [Appendix E, "List of Messages."](#)

The quota related messages are displayed on the terminal of the relevant user. The messages are not displayed on the console. The messages, however, are stored in `/var/adm/messages`.

21.4.5.3 Number of blocks in quota function

The number of blocks calculated in the quota function is that of the data area and directories. The number is equal to or less than the value obtained by summing up the numbers of blocks displayed by `ls(1)`, `du(1)`, etc. for each user.

21.4.5.4 Others

Note the following when using the quota function:

- Disk use amount
 - 1) If possible, the UIDs of users should be consecutive values. If the values are not consecutive, the quota data will contain unused areas. In addition, the quota data may not be a cache hit if accessed and would require real disk I/O that may reduce system performance.
 - 2) If the quota function is used in an existing GFS Shared file system, check for sufficient free area in the V-data area.



See

For details, see [Section 15.2.6, "quota function."](#)

21.4.6 Differences from UFS

21.4.6.1 Quotas files

The GFS Shared File System enlarges and optimizes the UFS quotas file format, therefore, the GFS Shared File System cannot use the UFS quotas file intact.

21.4.6.2 quotacheck(1M)

The GFS Shared File System does not provide a quotacheck(1M) command. Quota data is automatically matched in GFS Shared File System as required.

21.4.6.3 quota(1M)

If the UFS quota(1M) command is executed on an NFS client, the quota data on the NFS server can be collected and displayed.

The GFS Shared File System sfcquota(1M) command cannot collect or display the quota data on an NFS server.

21.4.6.4 sfcquotadm(1M)

The sfcquotadm(1M) command is specific to the GFS Shared File System. This command is executed to match the limit values of all users between the external quotas and internal quotas files. This command can also delete the internal quotas file.

21.5 Extending a File System

The GFS Shared File System can increase the size of an existing file system without saving or restoring data.

Use the `sfcadd` command to enlarge the file system. The file system can be enlarged by adding new file data partitions while the file system is mounted (online) or unmounted (offline).



See

For the execution procedure using WebView, see [Section 19.3.2, "Changing the file system configuration \(partition addition\)."](#) For the execution procedure using line commands, see [Section 20.5, "Change \(partition configuration\)."](#)



Note

If the file system is probably going to be extended, estimate the maximum size of the extension. Then, specify the data size of the file system when creating the file system (when the `sfcnewfs` or `mkfs_sfcs` command is executed). The GFS Shared File System records a free data area as an extent. For the free data management area, the area size is calculated automatically so that data in the specified data area can be managed when the file system is created. However, if the file system is extended, this management area may become insufficient. (Because a free data area is recorded as an extent, it can be managed if it is not fragmented. If fragmentation is severe, the free data area cannot be used because of the insufficient management area.)

To check whether the management area is insufficient, specify the `-ov` option in the `df_sfcs(1M)` command to display the total number of leafs and number of free leafs.

21.6 Displaying File System Information

21.6.1 Acquiring file system information

Information about a file system, including the use of disk blocks, the state of free disk blocks, and the number of files, can be acquired using `df_sfcfs(1M)`.

If a file system consists of multiple partitions, `df_sfcfs(1M)` will display information about each partition.

The following information is displayed for a meta-data partition:

- Use of i-nodes (number of i-nodes, number of free i-nodes)
- Free **extent** status (maximum number of leafs and amount used)
- Use of **V-data** (maximum number of V-data blocks and number used)
- unused area (size of the unassigned area)

The following information items are displayed for a **file data partition**:

- Total number of blocks and number of free blocks

Only the number of assigned blocks is displayed for an **update log partition**.

In the following example, `df_sfcfs(1M)` is used to display how a file system is being used.

```
# df -k /mnt
Filesystem      kbytes  used  avail capacity  Mounted on
/dev/sfdsk/gfs/rdisk/volume1
                113992  62664  39936    62%  /mnt

# df -F sfcfs -o v /mnt
Filesystem:/mnt
                inodes   free  vdata  free   leaf   free   Type
/dev/sfdsk/gfs/rdisk/volume1
                14248   12192  4781   4615   7128   7127   META
                kbytes   used   free  %used
/dev/sfdsk/gfs/rdisk/volume1
                113992  62664  51328   55%
                kbytes
/dev/sfdsk/gfs/rdisk/volume1
                5120
                                LOG

# df -F sfcfs -o v /mnt
Filesystem:/mnt
                inodes   free  vdata  free   leaf   free   Type
/dev/sfdsk/gfs/rdisk/volume1
                29686   27630  9922   9756  14848  14846  META
                kbytes   used   free  %used
/dev/sfdsk/gfs/rdisk/volume1
                105752  31832  73920   30%
                kbytes
/dev/sfdsk/gfs/rdisk/volume2
                5120
                                LOG
                kbytes   used   free  %used
/dev/sfdsk/gfs/rdisk/volume2
                131736  30832  100904  23%
                                DATA
```



See

For details on the `df_sfcfs(1M)` options, see [Appendix F.1.1, "df_sfcfs\(1M\)."](#)

21.6.2 Displaying partition/node information

The partition information about the group of partitions making up the GFS Shared File System can be displayed using `sfcinfo(1M)`. Node information is also displayed.

In the following example, `sfcinfo(1M)` is used to display the partition and node information about a file system.



For details on the `sfcinfo(1M)` options, see [Appendix F.2.11, "sfcinfo\(1M\)."](#)

```
# sfcinfo /dev/sfdsk/gfs/rdsk/volume1
FSID special                size Type  mount
  1 /dev/sfdsk/gfs/rdsk/volume1(3fa0081)  25986 META  /mnt
  1 /dev/sfdsk/gfs/rdsk/volume1(3fa0081) 105751 DATA /mnt
  1 /dev/sfdsk/gfs/rdsk/volume2(3fa0082)   5120 LOG   /mnt
  1 /dev/sfdsk/gfs/rdsk/volume2(3fa0083) 131736 DATA /mnt

# sfcinfo -n /dev/sfdsk/gfs/rdsk/volume1
/dev/sfdsk/gfs/rdsk/volume1:
FSID hostID  status  hostname
  1  80804266  RUN    mikan
  1  807e7937  RUN    karin
  1  80807658  RUN    mango
```

21.6.3 Displaying file system management state

The management state of the GFS Shared File System can be checked using `sfcrcinfo(1M)`.

Check file system configuration changes by displaying the file system management information using `sfcrcinfo(1M)` before mounting or unmounting a file system.

In the following example, `sfcrcinfo(1M)` is used to check the management state.

```
# sfcrcinfo -am
/dev/sfdsk/gfs/dsk/volume2:
FSID  MDS/AC  STATE  S-STATE  RID-1  RID-2  RID-N  hostname
  2  MDS(P)  run    -         0       0       0  shoga
  2  AC      run    -         0       0       0  shoga
  2  MDS(S)  wait   -         0       0       0  ichou
  2  AC      run    -         0       0       0  ichou
  2  MDS(I)  imag   -         0       0       0  karin
  2  AC      run    -         0       0       0  karin

- MDS/AC
  MDS(P) : The primary MDS on a definition
  MDS(S) : The secondary MDS on a definition
  MDS(I) : The other nodes to which MDS operates
  AC     : The node that mount the file system
- STATE/SUB-STATE of MDS
  run    : The primary MDS is running
  wait   : The secondary MDS is waiting for failure recovery of the primary MDS
  imag   : Waiting for failure recovery of the primary MDS or the secondary MDS
  stop   : MDS stopped
  none   : no relation with MDS
  If SUB-STATE is other than "-", MDS is in state-transition.
- STATE/SUB-STATE of AC
  run    : file system is in mounted
  stop   : file system is in unmounted
  If SUB-STATE is other than "-", AC is in state- transition.
```



See

For details on the `sfcrcinfo(1M)` options, see [Appendix F.2.26, "sfcrcinfo\(1M\)."](#)

21.7 Multiple LAN Setup

From the GFS Shared File System **AC**, multiple communication paths to the **MDS** can be set. By setting more than one LAN, file system management can be continued because it automatically searches for usable communication paths when a LAN fault occurs.

21.7.1 Setup

To set multiple LANs, specify multiple **MDS** host names with `sfnewfs(1M)` or `mkfs_sfdfs(1M)`.

Several host names can be specified, not only to add nodes for a file system but also to add **nodes** for a file system with `sfnode(1M)`.

Select host names in the order of their specification for the host names set previously.



For details on `sfnewfs(1M)`, `mkfs_sfdfs(1M)` and `sfnode(1M)`, see [Appendix F, "Reference Manual."](#)

21.7.2 Reference

Use `sfchost(1M)` to check the host name by which the **AC** is connected to the **MDS**.



For details on `sfchost(1M)`, see [Appendix F.2.10, "sfchost\(1M\)."](#)

21.7.3 Change

Use `sfchost(1M)` to connect with the **MDS** using a different host name. The new host name must be in the list of host names that were previously set.



For details on `sfchost(1M)`, see [Appendix F.2.10, "sfchost\(1M\)."](#)

21.8 Using the GFS Shared File System for RMS Cluster Operation

This section explains how to use the GFS Shared File System for RMS cluster operation.

21.8.1 Before using a GFS shared file system in RMS cluster operation

To use the GFS Shared File System in RMS cluster operation, the following RMS environment setting must be performed:

- Registering an RMS host name in /etc/hosts on each node
RMS recognizes each node that constitutes the RMS cluster from the name of the host to which RMS is attached, as shown in the definition below:

```
192.168.0.1 sunny sunnyRMS
192.168.0.2 moony moonyRMS
```

- Setting of cluster.config file or .rhosts file for each node
In order to distribute RMS composition definition information to each node, the userApplication Configuration Wizard uses CF remote services or rcp (1).
Either a .rhost file including the RMS host name is set up on each node or the cluster.config file is set up to enable CF remote services.



See

For details on cluster operation, see "CF security" in the "*PRIMECLUSTER Cluster Foundation Configuration and Administration Guide*."

21.8.2 To enable use as a shared file system for RMS cluster operation

When the GFS Shared File System is to be used as the shared file system for RMS cluster operation, disk resources and mount point resources must not be defined at RMS application creation.

The mount information must be registered in /etc/vfstab for the GFS shared file system and "no" must be specified in "mount at boot."

Note, however, that "noauto" must not be specified in "mount options" as the file system description in /etc/vfstab so that the file system registered in /etc/vfstab should be set to automatic mount after active processing of the GFS Shared File System.

21.8.3 Setup flow when a GFS shared file system is to be used in an RMS cluster operation

The following shows the flow of the setup procedure for creating a GFS shared file system on a GDS volume for use in an RMS cluster operation.

The flow of the setup procedure for creating a GFS shared file system on a GDS volume for use in an RMS cluster operation

Shared operation (to be executed on node A)	Node A (active node)	Node B (standby node)
-	Creating the GFS Shared File System management partition (See "Procedure 1" in "21.8.4.")	-
-	Node registration to the GFS Shared File System (See "Procedure 1" in "21.8.4.")	Node registration to the GFS Shared File System (See "Procedure 1" in "21.8.4.")
-	Creating the GFS Shared File System (See "Procedure 1" in "21.8.4.")	-
-	Adding relevant GFS Shared File System to /etc/vfstab (See "Procedure 2" in "21.8.4.")	Add relevant the GFS Shared File System to /etc/vfstab (See "Procedure 2" in "21.8.4.")
RMS configuration definition settings (RMS application switchover definition) (See the " <i>PRIMECLUSTER Configuration and Administration Guide</i> " and " <i>PRIMECLUSTER RMS Configuration and Administration Guide</i> .")	-	-
-	Start up RMS and check settings (See "Procedure 4" in "21.8.4.")	Start up RMS and check settings (See "Procedure 4" in "21.8.4.")

21.8.4 Setup procedure when the GFS Shared File System is to be used in an RMS cluster operation

This section explains the procedure for creating a GFS shared file system on a GDS volume for use by an RMS cluster operation.



See

For details, see the "*PRIMECLUSTER Configuration and Administration Guide*," "*PRIMECLUSTER RMS Configuration and Administration Guide*," and the "*PRIMECLUSTER Global Disk Service Configuration and Administration Guide*."

In addition to commands, a GDS volume can be also created by using the GDS Management View (GUI). For details to setup the GDS, refer to "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

Procedure 1. GFS Shared File System creation (node A (active node), node B (standby node))

Create a GFS shared file system on either of the nodes.

Use the `sfcnewfs` command to create a GFS shared file system.

Note that a management partition that is exclusively for the GFS Shared File System must be prepared by the GFS Shared File System on a GDS volume whose disk class does not include any GDS volumes that are to be used as switchover file systems.

When a GFS shared file system is created for the first time after installation, activation of the GFS Shared File System is necessary before file system creation.

- Create a management partition for a GFS shared file system on either of the nodes.

```
# sfcsetup -c /dev/sfdsk/gfs/rdsk/GFSctl
```

- Register node information in the management partition of the GFS Shared File System on each node.

```
sunny# sfcsetup -a /dev/sfdsk/gfs/rdsk/GFSctl
moony# sfcsetup -a /dev/sfdsk/gfs/rdsk/GFSctl
```

- Activate the GFS Shared File System on each node.

```
sunny# sfcfrmstart
moony# sfcfrmstart
```

- Create the GFS Shared File System on either of the nodes.

```
# sfcnewfs -H sunny,moony /dev/sfdsk/gfs/rdsk/vol0001
```

Procedure 2. Adding the relevant GFS shared file system to `/etc/vfstab` (node A (active node), node B (active node))

Add the mount information for the relevant GFS shared file system to `/etc/vfstab` on each node. At this time, specify “no” for “mount at boot” mount option in the `/etc/vfstab` entry.

```
/dev/sfdsk/gfs/dsk/vol0001 /dev/sfdsk/gfs/rdsk/vol0001 /sfcs sfcs - no rw
```

Procedure 3. Setting the definition of switchover for an RMS application in the RMS configuration (node A (active node), node B (standby node))

Set up the RMS configuration by using the userApplication Configuration Wizard.



See

For details on setup, refer to “*PRIMECLUSTER Configuration and Administration Guide*” and “*PRIMECLUSTER RMS Configuration and Administration Guide*.”



Note

For a GFS shared file system, the following measures are necessary in order for its GDS volumes to always be active.

- RMS applications must not set as an RMS target any mount point (“LocalFileSystems”) used by any GFS shared file system or any GDS class (AdditionalDiskClass under Gds:Global-Disk-Services) that contains any GDS volume used by a GFS shared file system or for the GFS Share File System management partition.
- `/usr/opt/reliant/bin/hvugdsetup` must not be executed to the GDS volumes.

Procedure 4. Confirming settings (node A (active node), node B (standby node))

After mounting the GFS shared file system on each node, start RMS and then confirm whether the settings in the `/etc/vfstab` file and the RMS application are correct, paying careful attention to the following points:

- If the GFS shared file system can be mounted on both nodes exactly as described in the /etc/vfstab file.
- If the RMS application switches over to a standby node at failover of the active node.



Note

If mounting of the GFS shared file system fails, the following may be the cause of the failure (apply the indicated solution):

- The type of file system specified in /etc/vfstab is incorrect.
Solution: Correct the /etc/vfstab file.
- The GDS volume is deactivated.
Solution: Activate the GDS volume.

21.8.5 Setup flow for addition of a file data partition to a GFS shared file system that is used in an RMS cluster operation

The setup flow for addition of a file data partition to a GFS shared file system that is used in an RMS cluster operation is shown below:

The flow of the setup procedure for addition of a file data partition to a GFS shared file system that is used in an RMS cluster operation

Common operation (to be executed on node A)	Node A (active node)	Node B (standby node)
-	Stop RMS. (Refer to "Procedure 1" in "21.8.6.")	Stop RMS. (Refer to "Procedure 1" in "21.8.6.")
-	Add a shared disk device as the GDS volume. (Refer to "Procedure 2" in "21.8.6.") Add a file data partition to the GFS Shared File System. (Refer to "Procedure 3" in "21.8.6.")	-
-	Start RMS. (Refer to "Procedure 4" in "21.8.6.")	Start RMS. (Refer to "Procedure 4" in "21.8.6.")

21.8.6 Setup procedure for addition of a file data partition to a GFS shared file system that is used in an RMS cluster operation

The setup flow for a file data partition addition to a GFS shared file system that is used in an RMS cluster operation is explained below:



See

For details on setting, see the "PRIMECLUSTER Configuration and Administration Guide," "PRIMECLUSTER RMS Configuration and Administration Guide," and the "PRIMECLUSTER Global Disk Service Configuration and Administration Guide."

Procedure 1. Stopping RMS (node A (active node), node B (standby node))

If RMS is activated, stop RMS on each node.

Procedure 2. Adding a file data partition to a GFS shared file system (node A (active node))

Add a file data partition to the relevant GFS shared file system from an active node with the `sfcadd(1M)` command.

```
sunny# sfcadd -D /dev/sfdsk/gfs/rdsk/vol0003 /dev/sfdsk/gfs/rdsk/vol0001
```

Procedure 3. Starting up RMS (node A (active node), node B (standby node))

Activate RMS on each node.

Chapter 22 File System Backing-up and Restoring

This chapter describes how to backup and restore data in a GFS shared file system.

22.1 Types of Backups and Restores

The following describes the types of backups and restores for GFS shared file systems.

- Backing up and restoring file by file
 - The GFS Shared File System backup and restoration tools `sfdump(1M)` and `sfcrestore(1M)` have the same functions as the UFS `ufsdump(1M)` and `ufsrestore(1M)` commands.
 - General purpose backup tools not dependent on the file system structure (ArcServe/OPEN, Solstice Backup, etc.)
 - Standard Solaris OE commands `cpio(1)` and `tar(1)`

- Backing up and restoring the entire file system
 - Standard Solaris OE command `dd(1M)`
 - support tool `sfcmbkup(1M)` for backing up and restoring by executing the `dd(1M)` command



Note

Commands that depend on the file system structure, such as UFS `ufsdump(1M)` and VxFS `vxdump(1M)`, cannot be used to backup a GFS shared file system. Use `sfdump(1M)` instead.

The following describes the the types of backups and restores of GFS shared file system data covered in this chapter.

- backing up and restoring data using `sfdump / sfcrestore(1M)` (22.2)
- backing up and restoring data using `sfcmbkup(1M)` (22.3)
- backing up and restoring data using Standard Solaris OE commands `cpio(1)`, `tar(1)` and `dd(1M)` (22.4, 22.5)

22.2 Backing Up and Restoring using sfdump(1M) and sfcstore(1M)

The sfdump(1M) and sfcstore(1M) commands are backup and restoration tools for GFS shared file systems. The commands have the same functions as the ufsdump(1M) and ufsrestore(1M) commands in UFS. The following are the main features of the sfdump(1M) and sfcstore(1M) commands:

- Multiple backups and restorations

The entire file system can be backed up. Incremental backups or backup as individual files can also be created. The file system can be restored from these backups.

- Multiple tape support

A file system can be backed up onto multiple tapes. Therefore, if the contents of a file system or file to be backed up cannot be stored on one tape, the contents can be backed up onto multiple tapes.

- Command line interface compatibility with ufsdump(1M) and ufsrestore(1M)

GFS shared file systems can be backed up and restored using the same command line interface as that of the ufsdump(1M) and ufsrestore(1M) commands in UFS.



Note

Note the following about compatibility between the ufsdump(1M)/ufsrestore(1M) and sfdump(1M)/sfcstore(1M) commands:

- Backup data created by the ufsdump(1M) command is not compatible with the data created by the sfdump(1M) command. The sfcstore(1M) command cannot restore the data backed up by the ufsdump(1M) command. In the same way, the ufsrestore(1M) command cannot restore the data backed up by the sfdump(1M) command.
- The sfdump(1M) command can backup GFS shared file systems. In addition, the ufsdump(1M) command cannot backup GFS shared file systems.



Note

Backup by using sfdump(1M) can only be performed from nodes defined as shared nodes of the GFS shared file system.

This section explains how to execute the sfdump(1M) and sfcstore(1M) command to backup and restore GFS shared file systems.



See

For details on sfdump(1M), see [Appendix F.2.5, "sfdump\(1M\)."](#) For details on sfcstore(1M), see [Appendix F.2.25, "sfcstore\(1M\)."](#)

22.2.1 Backing up using the sfdump(1M)

This section contains an example of backing up an entire GFS shared file system specified in the /etc/vfstab file shown below. Here, /dev/rmt/0 is the backup destination tape drive.

```
% cat /etc/vfstab
#device          device          mount FS  fsck mount  mount
#to mount        to fsck         point type pass  at boot options
...
/dev/sfdsk/gfs/rdisk/volume1 /dev/sfdsk/gfs/rdisk/volume1 /mnt  sfcs  -  no    -
...
```

The character special file in the representative partition of the GFS shared file system is /dev/rdisk/c0t10d0s0. The mount point is /mnt.

1. Set for superuser.

```
% su
```

2. Unmount the GFS shared file system to be backed up, so that the GFS shared file system is not updated.



Note

Even if the GFS shared file system is mounted, a backup can be performed. The backup must be performed while the GFS Shared File System is not being updated. If the file system is updated during the backup, the backup contents may be invalid.



Note

Note that sfdump fails with the following message output if unmounted of the GFS Shared File System is activated and the mounted of the node that performs backup is stopped:
DUMP: Warning - /dev/FJSVgdhd/rdsk/gdhd0s0 is mounted on other node(s) or requires a file system check. Stop the file system or run fsck.
DUMP: The ENTIRE dump is aborted.

In such cases, unmount the GFS Shared File System or perform the backup after mounting the file system on the node performing the backup. If the above message is output when the file system is unmounted, the file system integrity may be lost. Use fsck to restore the file system.



See

For details on repairing a file system, see [Section 21.2, "Checking a File System for Consistency and Repairing It."](#)

3. Define a tape drive to backup a GFS shared file system.

```
# sfdump 0ucf /dev/rmt/0 /dev/sfdsk/gfs/rdisk/volume1
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Tue May 23 11:20:14 2000
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/sfdsk/gfs/rdisk/volume1 (kaede:/mnt) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 282 blocks (141KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 250 blocks (125KB) on 1 volume at 1838 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Tue May 23 11:20:14 2000
```

Each character of 0ucf in the first parameter has the following meaning:

- 0
This is the backup level. The number 0 means a full backup. A number between 1 and 9 means an incremental backup. In this example, the file system is backed up as an incremental backup from the latest backup at a lower level than the specified backup level.
- u
The dump record file, /etc/sfcs_dumpdates, is updated. Specify this character for full backup or incremental backup.
- c
The contents are backed up onto tape.
- f
The write destination is specified. /dev/rmt/0 is specified as the write destination.

22.2.2 Restoration using sfcrestore(1M)

This section explains the procedure for fully restoring a GFS shared file system from a backup tape described in the previous section. The sfcrestore(1M) command can also restore a GFS shared file system backup to a different GFS shared file system. The restore procedure shown below applies when a backup and restore are performed in the same GFS Shared File System.



Note

If a different GFS shared file system is to be restored, note the capacity of the restoration destination. Restoration may fail because of insufficient capacity.

1. Set for superuser.

```
% su
```

2. If the destination GFS shared file system is mounted, unmount it.

3. Use the sfcnewfs(1M) command to re-create the GFS shared file system on the restoration destination. Use the same options and arguments as those used for creating this file system.

In addition, add the **-Z** option for instructing new creation. The following is shows an example of specifying "sfcnewfs -H kaede,ichou /dev/sfdsk/gfs/rdisk/volume1" for creating a file system before execution:

```
# sfcnewfs -Z -H kaede,ichou /dev/sfdsk/gfs/rdisk/volume1
```



Note

When a GFS Shared File System is created, the contents of the previously created GFS shared file system are lost.

- 4. Mount the GFS shared file system.
- 5. After the file system is mounted, change the current working directory to the mount point.

```
# cd /mnt
```

- 6. Insert the appropriate tape, and execute the `sfcrestore(1M)` command to restore the GFS shared file system from the tape.

```
# sfcrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump date: Wed May 24 13:28:08 2000
Dumped from: the epoch
Level 0 dump of /mnt on kaede: /dev/sfdsk/gfs/rdisk/volume1
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Warning: ./lost+found: File exists
Make node ./etc
Make node ./etc/default
...
extract file ./etc/fs/nfs/mount
extract file ./etc/fs/ufs/mount
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
```

Each character of `rvf` in the first parameter has the following meaning:

- **r**
All contents of the backup media are restored. Specify this character to restore the entire contents of the backup data.
- **v**
Details are displayed. The name of each file is displayed when it is restored.
- **f**
The read source is specified. `/dev/rmt/0` is specified as the read source.



Information

To restore an incremental backup, restore the level 0 backup tape according to step 6 and then repeat step 6 in ascending order based on the backup level.

- 7. Delete the `restoresymtable` file, which is a temporary file created by the `sfcrestore(1M)` command.

```
# rm /mnt/restoresymtable
```

- 8. To restore a GFS shared file system that uses the quota function, execute the `sfcquotadm(1M)`

22.2 Backing Up and Restoring using sfcdump(1M) and sfcrestore(1M)

command with the `-i` option specified to reflect the contents of an external quotas file to the internal quotas file. For information on the `sfcquotadm(1M)` command, see [Appendix F.2.22 "sfcquotadm\(1M\)."](#) This step is not required if a quota function is not used.

```
# sfcquotadm -i /mnt
```

9. Move to a different directory.

```
# cd /
```

10. Umount the newly restored GFS shared file system.

11. Execute the `sfcdump(1M)` command to fully backup the newly restored the GFS shared file system. Insert a tape and then execute the `sfcdump(1M)` command to backup the GFS shared file system.

```
# sfcdump 0ucf /dev/rmt/0 /mnt
```



Note

Backup the entire GFS shared file system immediately after it has been restored. This must be done because the assignment of the i-node numbers changes after restoration using the `sfcrestore(1M)` command. An incremental backup cannot correctly generated from a newly restored GFS shared file system.

12. Mount the restored GFS shared file system.

22.2.3 Notes on using differential backup

`sfcdump(1M)` provides the differential backup function that backs up only files updated after the last backup. This manual does not describe a specific procedure for differential backup.

Note the following when using differential backup.

22.2.3.1 The node that performs the differential backup

Differential backup is implemented by using the dump record file (`/etc/sfdfs_dumpdates`) information updated through `sfcdump` command execution with the `u` function modifier specified. Only dump record files on the node that executed `sfcdump` are updated, therefore, differential backups must always be performed from the same node.

If a differential backup is to be performed from another node, the dump record file must be copied from the node where backup was previously performed.

22.2.3.2 If a panic occurs on a node

A differential backup of a GFS shared file system may not properly backup files updated from a node that did not stop normally, for example due to a panic.

`sfcdump(1M)` determines whether to backup files during a differential backup by comparing the time information of files and the time when backup was last performed as obtained from the dump record file. If a node does not stop normally, the contents of files may be updated without the time information of the files being updated. Such files will not be backed up by a differential backup. They may not be backed up until the next full backup.

If a node using the GFS shared file system has failed to stop normally, the next backup should be a full backup.

22.3 Backup/Restoration using sfcmbkup

The sfcmbkup command is a support tool for backing up and restoring by executing the dd command. A GFS shared file system can be created with a multi-partition configuration. In this configuration, however, operation is complicated because the dd command must be executed several times to backup the GFS shared file system. In addition, each backup of the multiple partitions must be restored to its correct partition during restoration. If the backups are not restored to the correct partitions, the restore may fail.

The sfcmbkup(1M) command is a support tool that solves this problem. The sfcmbkup(1M) command can be executed to create shell scripts for backing up and restoring a GFS shared file system. These commands provide a safe backup and restoration process for a GFS shared file system.

This section shows how to execute sfcmbkup(1M) command to backup and restore a GFS shared file system.



See

For details on sfcmbkup(1M), see [Appendix F.2.12, "sfcmbkup\(1M\)."](#)

Superuser authority (root) is required to execute the sfcmbkup(1M) command and run shell scripts created by the sfcmbkup(1M) command.

Assume a GFS shared file system has the multi-partition configuration shown below.

```
# sfcinfo /dev/sfdsk/gfs/rdsk/volume1
FSID special size Type mount
1 /dev/sfdsk/gfs/dsk/volume1(3fa0081) 25986 META /mnt
1 /dev/sfdsk/gfs/dsk/volume1(3fa0081) 105751 DATA /mnt
1 /dev/sfdsk/gfs/dsk/volume2(3fa0082) 5120 LOG /mnt
1 /dev/sfdsk/gfs/dsk/volume3(3fa0083) 131736 DATA /mnt
```

To backup and restore this file system, specify the **representative partition** of the file system and the media device used to make the backup copy then create a shell script for the backup and restoration.

The following shows an example of creating a shell script to backup and restore the above file system using /dev/rmt/0 as the backup device.

```
# sfcmbkup /dev/sfdsk/gfs/rdsk/volume1 /dev/rmt/0
```

After executing the sfcmbkup(1M) command, the following two shell scripts are generated in the current working directory:

```
sfdfs.bkup    Shell script for performing backup
sfdfs.rstr    Shell script for restoring
```

These scripts are executed to back up or restore a file system.

The following shows an example of executing a shell script.

- Execute backup by using the sfdfs.bkup shell script

```
# ./sfdfs.bkup
```

- Example of restoring to same partition as backup source using shell script sfdfs.rstr

```
# ./sfdfs.rstr
```

Editing the variable SPECS of script sfcfs.rstr allows specified partitions to be restored to a disk other than the one used for the backup. For this purpose, place "!" before the edited special file name to indicate that volume information should not be checked.

```
SPECS="/dev/sfdsk/gfs/dsk/volume1 ! /dev/sfdsk/gfs/dsk/volume2 /dev/sfdsk/gfs/dsk/volume3"
```

**Note**

If a partition is restored to one different than that of the backup source, the partition configuration at the restore destination must be the same as that at the backup source.

If a partition is restored to one different than that of the backup source, the following operations are required after restoration:

- Changing information about partitions configuring the file system using sfcadm(1M)

The device size required for the backup can be estimated by using the sfcmbkup(1M) command before performing the backup. Prepare the required medium for the backup in accordance with the estimate.

Specify the representative partition to execute the sfcmbkup(1M) command

```
# sfcmbkup -e /dev/sfdsk/gfs/rdsk/volume1
special                size(KB)
/dev/sfdsk/gfs/rdsk/volume1    131760
/dev/sfdsk/gfs/rdsk/volume2    131760
/dev/sfdsk/gfs/rdsk/volume3    131760
```

22.4 Backing Up using Standard Solaris OE commands

22.4.1 Backing up file by file

To backup a single file or all files under a directory, use `cpio(1)`, `tar(1)` or `sfccpio(1)`.



Note

Extent attributes cannot be copied by using `cpio(1)` or `tar(1)`. Use `sfccpio(1)` when **extent attribute information** may be included in the file to be backed up.

Below are examples of backing up onto tape using `cpio (1)`, `tar (1)` and `sfccpio(1)`

- Backup method using `cpio(1)`

```
# cd /userdata
# find . -depth -print | cpio -oc > /dev/rmt/0
```

- Backup method using `tar(1)`

```
# cd /userdata
# tar cvf /dev/rmt/0 .
```

- Backup method using `sfccpio(1)`

```
# cd /userdata
# find . -depth -print | sfccpio -oc > /dev/rmt/0
```



Note

The tapes used for backups must be labeled before storage so that the backup method can be identified. `cron(1M)` can be used to start backups at specific times.



See

For details on `cpio(1)` and `tar(1)`, see the "*Solaris X Reference Manual Collection*" or the online manual.

For details on `sfccpio(1)`, see [Appendix F.3.2 "sfccpio\(1\)."](#)

22.4.2 Backing up an entire file system

To back up the GFS Shared File System, use `dd(1M)`. The following describes the backup procedure for using `dd(1M)`.

Entire file system backups are faster than file by file backups. However, a file system backed up with this method cannot be restored on a file by file basis.

Before starting a backup, use the following procedure to check the **partition** information for the desired GFS shared file system. Estimate the backup capacity required including the number of physical media required and then unmount the GFS shared file system.

- For a single partition configuration

Use `sfcinfo(1M)` to display partition information and check that the file system has only a single partition (same special file names) and that the capacity of the tape is sufficient. The size of the

partition can be estimated by adding 1 MB to the sum of the values displayed in the size field (in kilobytes).

Example: In the following example, a file system with a single partition configuration is backed up.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
FSID special                size Type mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    12626 META /mnt
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)   113992 DATA /mnt
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    5120 LOG /mnt

# dd if=/dev/sfdsk/gfs/rdsk/volume1 of=/dev/rmt/0 bs=1024k
```

- For a multi-partition configuration
dd(1M) can handle one partition at a time. For a multi-partition configuration, therefore, the partitions must be backed up one at a time. Use sfcinfo(1M) to check the partition configuration then backup each partition with the same procedure as used for a single partition configuration.

Example: In the following example, a file system with three partitions (log area separation + data area addition) is backed up.

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
FSID special                size Type mount
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)    25986 META /mnt
  1 /dev/sfdsk/gfs/dsk/volume1(3fa0081)   105751 DATA /mnt
  1 /dev/sfdsk/gfs/dsk/volume2(3fa0082)     5120 LOG /mnt
  1 /dev/sfdsk/gfs/dsk/volume3(3fa0083)   131736 DATA /mnt

# dd if=/dev/sfdsk/gfs/rdsk/volume1 of=/dev/rmt/0 bs=1024k -> Tape 1
Tape change
# dd if=/dev/sfdsk/gfs/rdsk/volume2 of=/dev/rmt/0 bs=1024k -> Tape 2
Tape change
# dd if=/dev/sfdsk/gfs/rdsk/volume3 of=/dev/rmt/0 bs=1024k -> Tape 3
```



Note

- All the partitions making up the desired GFS shared file system must be backed up. Avoid backing up and storing only some of the partitions.
- The tapes used for backups must be labeled so that the backed up partitions can be identified. They must then be stored as a set.



Note

dd(1M) does not support multi-tape volumes. If the file system is too large to fit on a single tape, it must be backed up as separate blocks. To do this, set **bs** (block length) and **count** (number of blocks) and increase **isseek** (input-side offset (number of blocks) by the value of count).

Example: In the following example, a partition is backed up in 1 GB units.

```
# dd if=/dev/sfdsk/gfs/rdisk/volume1 of=/dev/rmt/0 bs=1024k count=1024
  -> Tape 1
Tape change
# dd if=/dev/sfdsk/gfs/rdisk/volume1 of=/dev/rmt/0 bs=1024k count=1024 isseek=1024
  -> Tape 2
Tape change
# dd if=/dev/sfdsk/gfs/rdisk/volume1 of=/dev/rmt/0 bs=1024k count=1024 isseek=2048
  -> Tape 3
Tape change
# dd if=/dev/sfdsk/gfs/rdisk/volume1 of=/dev/rmt/0 bs=1024k count=1024 isseek=3072
  -> Tape 4
...

```

22.5 Restoring using Standard Solaris OE commands

The following describes the procedure for restoring a file system from the backup tapes prepared as described earlier.

Two restoration methods are available:

- Restoring file by file using the appropriate standard Solaris OE command
- Restoring an entire file system using dd(1M)

22.5.1 Restoring File by file

Use cpio(1), tar(1) or sfccpio(1) to restore files from the backup tape to the disk.



Note

Extent attributes can not be copied using cpio(1) or tar(1). Use sfccpio(1) when **extent attribute information** may be included in the file to be backed up.



See

For details on cpio(1) and tar(1), see the "*Solaris X Reference Manual Collection*" or the online manual.

For details on sfccpio(1), see [Appendix F.3.2, "sfccpio\(1\)."](#)

The following are examples of restoring from a tape.

- Restoring using cpio(1)

```
# cd /userdata  
# cpio -icdm < /dev/rmt/0
```

- Restoring using tar(1)

```
# cd /userdata  
# tar xvf /dev/rmt/0
```

- Restoring using sfccpio(1)

```
# cd /userdata  
# sfccpio -icdm < /dev/rmt/0
```

Restoring from a backup tape must be performed using the same command that was used to create the backup.

22.5.2 Restoring an entire file system

Use dd(1M) to restore each **partition** from the backup tape to the disk

Before restoring, unmount the desired GFS shared file system on all the nodes using the file system.

- For a single partition configuration

Example: In the following example, a file system with a single partition configuration is restored.

```
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/volume1 bs=1024k
```

- For a multi-partition configuration

Example: In the following example, a file system with three partitions (log area separation + data area addition) is restored.

```
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/volume1 bs=1024k <- Tape 1
Tape change
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/volume2 bs=1024k <- Tape 2
Tape change
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/volume3 bs=1024k <- Tape 3
```



Note

dd (1M) does not support multi-tape volumes. To restore a file system backed up as separate blocks, specify the same values for the **bs** and count parameters as for where specified for the backup. For **oseek**, specify the same value that was specified for **iseek**.

Example: In the following example, a file system that was backed up in 1 GB units is restored.

```
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/volume1 bs=1024k count=1024
<- Tape 1
Tape change
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/volume1 bs=1024k count=1024 oseek=1024
<- Tape 2
Tape change
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/volume1 bs=1024k count=1024 oseek=2048
<- Tape 3
Tape change
# dd if=/dev/rmt/0 of=/dev/sfdsk/gfs/rdsk/volume1 bs=1024k count=1024 oseek=3072
<- Tape 4
...
```


22.6 Backing up and Restoring User Limit Values of the quota function

When a file is backed up or restored, the user limit values of the **quota function** of the GFS share file system can be backed up or restored by backing up or restoring the quotas file for each GFS share file system. The GFS Shared File System manages the user limit values with the internal and external quota files. When backing up or restoring, `sfcquotadm(1M)` must be used to ensure that the contents of the files are consistent.



See

For details on `sfcquotadm(1M)`, see [Appendix F.2.22, "sfcquotadm\(1M\)."](#)

The following shows the procedure by which the user limit values of the quota function of the file system to be mounted in `/sfcs` are saved onto another file system using `cp(1)`:

```
# sfcquotadm -e /sfcs
# cp /sfcs/quotas /work/quotas
```

The following shows the procedure by which the user limit values are restored by restoring the quotas file `/work/quotas` previously backed up to the file system mounted in `/sfcs`:

```
# cp /work/quotas /sfcs/quotas
# sfcquotadm -i /sfcs
```



Note

When the quota function is disabled, it must be enabled for restoration.

22.7 High speed backup and restoration of a GFS shared file system

This section explains high speed backup and restoration of a GFS shared file system implemented by coordinating with GDS and using the `sfcproxyop(1M)` and `sfcproxyrestore(1M)` commands.



Optional product is required for this function.

22.7.1 Outline

The GFS Shared File system, by coordinating with **GDS** provides a snapshot function that allows users to create a replica (snapshot) of a GFS shared file system. Snapshots allow operations such as non-disruptive backups or data analysis to run concurrently with core business applications.

Furthermore, using the advanced copy feature of our company's ETERNUS3000/GR720/GR730/GR740 disk array, a backup of a GFS shared file system can be created instantly without imposing a load on the server or SAN.

The snapshot copy is instantly logically complete. File system access need only be stopped very briefly while the copy logically completes, minimizing the impact on core business applications.

Even if the GFS shared file system is mounted, a snapshot copy can be created. If a snapshot is created while an application is accessing the file system, the volume data may not be complete. To create a snapshot while running an application, the application or database must ensure data integrity.

The restored file system can be immediately used after the logical completion of restoration.

22.7.2 Implementation methods

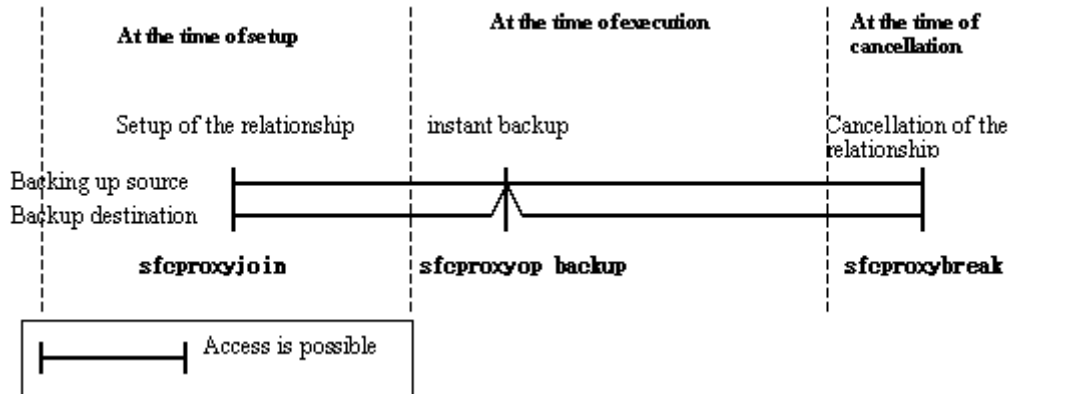
In the GFS Shared File System, the following two GR disk array features are employed for instant copy. The GFS Shared File System implements instant copy using OPC (One Point Copy) and EC (Equivalent Copy).

22.7.2.1 OPC (One Point Copy) employment - Implementation by instant copy

This implementation uses the **OPC** function in the disk array to create the instant copy. It is a requirement that the OPC function of our company's ETERNUS3000/GR720/GR730/GR740 disk array is enabled.

Issuing the backup command logically copies the contents of the file system to the backup destination. The backup is instantly logically complete. The contents of the backup destination are held until the next backup. Until the next backup, the backup destination can be mounted using `sfcproxyop(1M)` command and backups from this file system can be performed.

Figure OPC employment



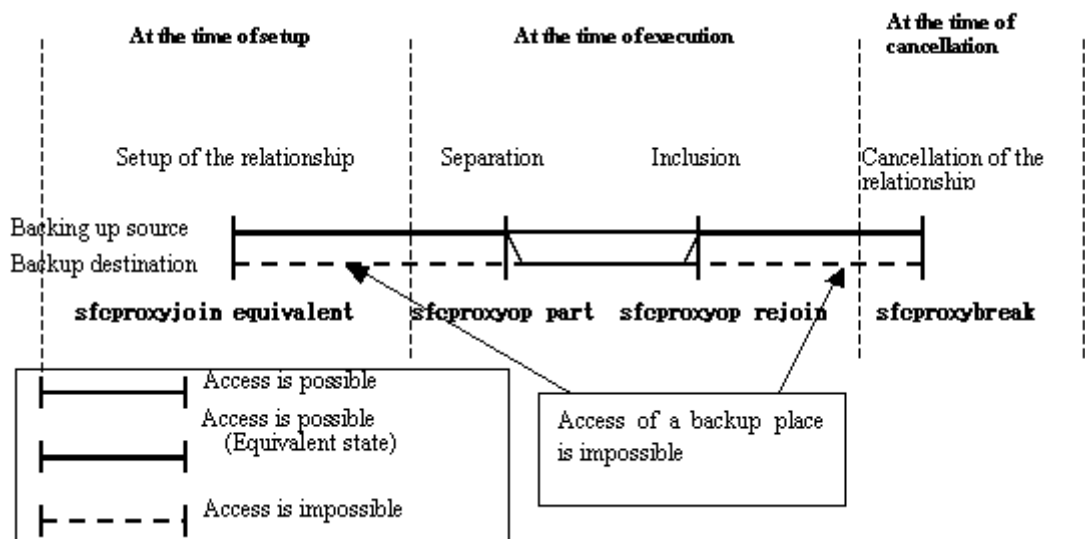
22.7.2.2 EC(Equivalent Copy) - Implementation by joining then separating

EC starts by setting up the relationship between the backup source and backup destination. While the backup source and destination are joined, the backup destination is maintained as an equivalent image of the backup source. A backup copy can be created instantly by separating the backup destination from the backup source. The physical copy must be complete before the backup destination can be separated.

The EC function can be implemented in the disk array or software. If the EC function is enabled in the disk array, the physical copy can be completed without placing a load on the server or the SAN.

After a backup is completed to tape or other media, the backup destination can be rejoined to the backup source in preparation for the next backup copy. While the backup destination is separated it can be mounted using the `sfcproxyop(1M)` command.

Figure EC employment



22.7.2.3 Selection of an employment pattern

Choose **OPC employment** or **EC employment** after carefully considering the characteristics of each.

EC requires that the physical copy to the backup destination be complete before the backup destination can be separated. The setup of the backup destination must be scheduled to allow completion of the physical copy before the backup copy is required. OPC allows a backup to be taken at an arbitrary time. It is unnecessary to schedule time to allow the physical copy to complete before the backup, however, the I/O performance of the backup source and destination are affected until the physical copy is complete.

OPC requires that the OPC function of our company's ETERNUS3000/GR720/GR730/GR740 disk array be enabled. EC will be used if OPC is not available.

22.7.3 Precondition

High speed backup and restoration requires the following:

- Software
 GDS is required. The file system must be created on GDS volumes that are configured for high speed backup and restoration.
- Hardware
 OPC requires that the OPC feature be enabled on our company's ETERNUS3000/GR720/GR730/GR740 disk array. EC does not require disk array features.

22.7.4 Contents of the backup image

A GFS shared file system records the partition information on the media. The partition information on the backup destination requires modification before the backup destination can be mounted. When the backup source is restored from the backup destination, the partition information on a backup source requires modification before the backup source can be mounted.

The GFS Shared File System provides commands to handle modification of the partition information. The commands are `sfcproxyop part` for an OPC backup, `sfcproxyop backup` for an EC backup, and `sfcproxyrestore` for restoring the backup source from the backup destination.

22.7.5 Mount Unmount of a backup place

The GFS shared file system created in the backup destination can not be mounted or unmounted using usual file system commands.

The command `sfcproxyop mount` mounts the backup destination. The command `sfcproxyop unmount` unmounts the backup destination.



For details on `sfcproxyop (1M)`, see [Appendix F.2.19, "sfcproxyop\(1M\)."](#)

Usage cautions (differences from a usual GFS shared file system)



Note

- Do not mount on more than one node at a time.
- Unmount is not performed automatically during system shutdown. Always perform a normal unmount using `sfcproxyop unmount` after use.

22.7.6 Setup of the backup relationship

It is necessary to perform the following steps to establish the backup relationship. Refer to preceding figures in this section **OPC employment** and **EC employment**.

(1) Lock the GDS volumes

Set the lock attribute for all the GDS volumes of the backup destination on all nodes but the node using the backup destination.

Stop all the GDS volumes of the backup destination on all nodes but the node using the backup destination.

The volume lock will prevent a node from accidentally starting a volume used by the backup destination.

(2) Setup of the backup relationship

Set up the relationship between all of the partitions of the GFS shared file system backup source and backup destination.

22.7.6.1 Lock the GDS volumes

The GDS `sdxattr(1M)` command sets the volume lock attribute which is used to deter a node from starting a GDS volume.



For details on the `sdxattr` command, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide."

(Example) Locking the backup destination volumes

The backup destination consists of the following GDS volumes:

`/dev/sfdsk/class0001/rdisk/volume0003` (class `class0001`, volume `volume0003`)

`/dev/sfdsk/class0001/rdisk/volume0004` (class `class0001`, volume `volume0004`)

The GDS volumes are accessible by `nodeA`, `nodeB`, and `nodeC`. The following commands are executed when `nodeC` performs the backup.

```
nodeA:
# sdxattr -V -c class0001 -v volume0003 -a lock=on
# sdxattr -V -c class0001 -v volume0004 -a lock=on
nodeB:
# sdxattr -V -c class0001 -v volume0003 -a lock=on
# sdxattr -V -c class0001 -v volume0004 -a lock=on
```

22.7.6.2 Setting up the backup relationship

The relationship between the backup destination and backup source is set up after the volumes of the backup destination are locked and stopped on all but the command node.

A relationship is set up between all partitions of a GFS shared file system that is the backup source to the partitions of the backup destination.

The corresponding partitions of the backup source and destination must relate as GDS master and proxy. It is necessary to configure the GDS master/proxy relationship. The main points are shown below. For details, see "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

- master and proxy should belong to different groups of the same disk class.
- The volume to be set up as master must not already be a master or a proxy.
- The volume to be set up as proxy must not already be a master or a proxy.

The following are requirements for the backup source and destination.

- The backup source and backup destination must have the same partition configuration including the same number partitions of the same size.
- When a file system is already created on the backup destination, the configuration of MDS nodes and sharing nodes of the backup destination must be the same as the backup source.
- When using OPC, the corresponding partitions for the backup source and backup destination must be in the same ETERNUS3000/GR720/GR730/GR740 disk array cabinet.

22.7 High speed backup and restoration of a GFS shared file system

- When using OPC, the partitions of the backup source destination are not mirrored by GDS.

As part of the setup of the backup source and backup destination relationship, a name is associated to identify this backup relationship. It will be referred to as the relationship name and is used by all operations to identify the specific backup relationship.

Configuration restriction:

- An employment pattern can not be mixed within partitions for the backup source and destination of a GFS shared file system. All of the volumes used for the backup location and destination must use either EC or OPC.

Before setting up the relationship, all the GDS volumes of the backup destination must be stopped. Use the `GDS sdxvolume(1M)` command to stop the GDS volumes on all nodes that share the volume.



See

For details on the `sdxvolume(1M)` command, see "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

(Example 1) Stopping the backup destination volumes

The backup destination consists of the following GDS volumes:

`/dev/sfdsk/class0001/rdsk/volume0003` (class class0001, volume volume0003)

`/dev/sfdsk/class0001/rdsk/volume0004` (class class0001, volume volume0004)

and the volumes are accessible by nodeA, nodeB, and nodeC.

The commands performed are as follows:

```
nodeA:
# sdxvolume -F -c class0001 -v volume0003
# sdxvolume -F -c class0001 -v volume0004
nodeB:
# sdxvolume -F -c class0001 -v volume0003
# sdxvolume -F -c class0001 -v volume0004
nodeC:
# sdxvolume -F -c class0001 -v volume0003
# sdxvolume -F -c class0001 -v volume0004
```



Note

The relationship can not be set up when any volume of the backup source has an error status. If a volume of the backup source is stopped on any node, it will display an error status. For details on recovery from an error state refer to "Appendix F.1 Resolving Problems in the "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide*."

OPC sets up the relationship using the `sfcproxyjoin` separate command and EC uses the `sfcproxyjoin` equivalent command.



See

For details on `sfcproxyjoin(1M)` command, see [Appendix F.2.18. "sfcproxyjoin\(1M\)."](#)

(Example 2) Setup of the relationship for OPC

The file system of the backup source contains the following two partitions:

/dev/sfdsk/class0001/rdisk/volume0001

/dev/sfdsk/class0001/rdisk/volume0002

The backup destination contains the following two partitions.

/dev/sfdsk/class0001/rdisk/volume0003

/dev/sfdsk/class0001/rdisk/volume0004

The following commands relate the partitions of the backup source and the backup destination for OPC.

```
# sfcproxyjoin separate -r backup1_for_volume0001 ¥  
/dev/sfdsk/class0001/rdisk/volume0001=/dev/sfdsk/class0001/rdisk/volume0003 ¥  
/dev/sfdsk/class0001/rdisk/volume0002=/dev/sfdsk/class0001/rdisk/volume0004
```

(Example 3) Setup of relationship for EC

The backup source contains the following two partitions.

/dev/sfdsk/class0001/rdisk/volume0001

/dev/sfdsk/class0001/rdisk/volume0002

The backup destination contains the following two partitions.

/dev/sfdsk/class0001/rdisk/volume0003

/dev/sfdsk/class0001/rdisk/volume0004

The following command relates the partitions of the backup source and the backup destination for EC.

```
# sfcproxyjoin equivalent -r backup1_for_volume0001 ¥  
/dev/sfdsk/class0001/rdisk/volume0001=/dev/sfdsk/class0001/rdisk/volume0003 ¥  
/dev/sfdsk/class0001/rdisk/volume0002=/dev/sfdsk/class0001/rdisk/volume0004
```

22.7.7 Operation at execution time

22.7.7.1 Backup by the instant copy

OPC uses the `sfcproxyop backup` command to create the backup copy.



See

For details on `sfcproxyop (1M)`, see [Appendix F.2.19, "sfcproxyop\(1M\)."](#)

(Example)

```
# sfcproxyop backup backup1_for_volume0001
```

After the `sfcproxyop backup` command has been executed, the backup destination can be used to perform offline or online backups. Offline backups can be created using commands such as `sfdump(1M)`, `dd(1M)`, and `sfcmbkup(1M)`. Online backups using commands such as `tar(1)` and `sfcpcio(1)` can be created after mounting the backup destination using the `sfcproxyop mount` command.

22.7.7.2 Separation of a backup place

EC uses the `sfcproxyop part` command to create the backup copy.



See

For details on `sfcproxyop (1M)`, see [Appendix F.2.19, "sfcproxyop\(1M\)."](#)

(Example)

```
# sfcproxyop part backup1_for_volume0001
```

After the `sfcproxyop part` command has been executed, the backup destination can be used to perform offline or online backups. Offline backups can be created using commands such as `sfcdump(1M)`, `dd(1M)`, and `sfcmbkup(1M)`. Online backups using commands such as `tar(1)` and `sfccpio(1)` can be created after mounting the backup destination using the `sfcproxyop mount` command.

22.7.7.3 Rejoining the backup destination to the backup source

In EC employment the `sfcproxyop rejoin` command is used to reestablish the relationship between the backup source and a separated backup destination



See

For details on `sfcproxyop (1M)`, see [Appendix F.2.19. "sfcproxyop\(1M\)."](#)

(Example)

```
# sfcproxyop rejoin backup1_for_volume0001
```

22.7.7.4 Mount of the backup destination

After the backup destination has been created, either with `sfcproxyop backup` for OPC or `sfcproxyop part` for EC, the backup destination file system can be mounted using the `sfcproxyop mount` command.



See

For details on `sfcproxyop (1M)`, see [Appendix F.2.19. "sfcproxyop\(1M\)."](#)

(Example)

```
# sfcproxyop mount backup1_for_volume0001 /mnt
```

22.7.7.5 Unmount of the backup destination

Use the `sfcproxyop umount` command to unmount a file system mounted with the `sfcproxyop mount` command.



See

For details on `sfcproxyop (1M)`, see [Appendix F.2.19. "sfcproxyop\(1M\)."](#)

(Example)

```
# sfcproxyop umount backup1_for_volume0001 /mnt
```

22.7.8 Operation at the time of cancellation

22.7.8.1 Canceling the backup relationship

The `sfcproxybreak(1M)` command is used to cancel the backup source to backup destination relationship.



For details on `sfcproxybreak(1M)`, see [Appendix F.2.16, "sfcproxybreak\(1M\)."](#)

(Example)

```
# sfcproxybreak backup1_for_volume0001
```

22.7.8.2 Release of the volume locks

The GDS volumes of the backup destination were locked on all but one node at the time the relationship was set up.

The GDS command `sdxattr(1M)` is used on each node to cancel the volume lock for that node.



For details on the `sdxattr` command, see "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide.*"

(Example) Releasing the volume lock on the backup destination

The following commands are executed when the backup destination contains the following two GDS volumes.

```
/dev/sfdsk/class0001/rdisk/volume0003 (class class0001, volume volume0003)
```

```
/dev/sfdsk/class0001/rdisk/volume0004 (class class0001, volume volume0004)
```

The following commands release the locks setup on nodeA and nodeB.

```
nodeA:  
# sdxattr -V -c class0001 -v volume0003 -a lock=off  
# sdxattr -V -c class0001 -v volume0004 -a lock=off  
nodeB:  
# sdxattr -V -c class0001 -v volume0003 -a lock=off  
# sdxattr -V -c class0001 -v volume0004 -a lock=off
```

22.7.9 Restoring the backup source

22.7.9.1 Restoring the backup source from the backup destination

A file system can be restored by instant copy from the backup destination to the backup source. The following explanation assumes that the backup destination contains valid data from a previous backup.

The `sfcproxyrestore(1M)` command is used to restore the backup source from the backup destination.



See

For details on `sfcproxyrestore(1M)`, see [Appendix F.2.20, "sfcproxyrestore\(1M\)."](#)

(Example)

```
# sfcproxyrestore backup1_for_volume0001
```

The `sfcproxyrestore(1M)` command restores the backup destination to the backup source. In addition, the backup relationship between the backup source and backup destination is restored. After the physical copy is complete, a new backup can be created. In an OPC employment, the backup copy is created using the `sfcproxyop part` command.

(Example)

```
# sfcproxyop part backup1_for_volume0001
```

22.7.9.2 Restoring the backup source from secondary media, such as tape

The procedure for restoring from a secondary backup, such as tape, depends on the method used to create the backup.

[When restoring per partition]

When a partition including the partition information is backed up from the backup destination by a command such as `dd(1M)`, the partition information of the backup destination is included in the backup. Restore the backup to the backup destination location then use the appropriate high speed copy to restore to the backup source location. For details, see the procedure in "[Restoring the backup source from the backup destination.](#)"

[When restoring per file]

The partition information is not included in backups created using `tar(1)`, `sfccpio(1)`, `sfcrestore(1M)`, etc., therefore, the backup can be restored directly to the backup source after the backup source has been re-created and mounted.

22.7.10 Displaying the copy state

The `sfcproxyinfo(1M)` command displays the relationship and copy state information for the backup sources and backup destinations.



See

For details on `sfcproxyinfo(1M)`, see [Appendix F.2.17, "sfcproxyinfo\(1M\)."](#)

The following information is displayed.

- Relationship name
- OPC **employment** or EC **employment**
- Equivalent state
- Error state
- Physical copy state
- The correspondence between partitions and the copy state between the partitions

(Example 1) Status of an instant copy using OPC employment after some of the partitions have completed the physical copy

```
# sfcproxyinfo backup1_for_volume0001

relation name: backup1_for_volume0001
type: OPC
status: SEPARATE, COPY
master          proxy          m_status      p_status
/dev/sfdsk/class0001/rdisk/volume0001 /dev/sfdsk/class0001/rdisk/volume0004 DONE COPY
/dev/sfdsk/class0001/rdisk/volume0002 /dev/sfdsk/class0001/rdisk/volume0005 DONE DONE
/dev/sfdsk/class0001/rdisk/volume0003 /dev/sfdsk/class0001/rdisk/volume0006 DONE COPY
```

(Example 2) Status following a `sfcproxyjoin` or `sfcproxyop` rejoin using EC employment

```
# sfcproxyinfo backup1_for_volume0001

relation name: backup1_for_volume0001
type: EC
status: JOIN, COPY
master          proxy          m_status      p_status
/dev/sfdsk/class0001/rdisk/volume0001 /dev/sfdsk/class0001/rdisk/volume0004 DONE COPY
/dev/sfdsk/class0001/rdisk/volume0002 /dev/sfdsk/class0001/rdisk/volume0005 DONE DONE
/dev/sfdsk/class0001/rdisk/volume0003 /dev/sfdsk/class0001/rdisk/volume0006 DONE DONE
```

22.7.11 Actions following an error

After a copy process fails, the contents of the GDS volume of the backup source or the backup destination maybe invalid. Cancel the backup relationship, correct the problem causing the error, and then restore the backup relationship. The following procedure explains the process.

22.7.11.1 Retaining Informational and canceling the backup relationship

The backup relationship is canceled by using the `sfcproxybreak(1M)` command. The following explains how to collect the configuration information before canceling the backup relationship.

The following information is needed to restore the backup relationship.

- (1) Relationship name
- (2) **OPC employment** or **EC employment**
- (3) Correspondence between partitions

The backup relationship information can be saved by directing the output of the `sfcproxyinfo -f` command to a file. The information will be needed to recreate the backup relationship after the problem causing the copy failure is corrected.

The following is an example of saving the configuration information by using the `sfcproxyinfo(1M)` command.

```
# sfcproxyinfo -f backup1_for_volume0001 &gt; /var/tmp/relation_backup
```



For details, see Section 22.7.8.1, "[Canceling the backup relationship.](#)" The relationship can be canceled with the volume locked on other nodes.

22.7.11.2 Recovery from an INVALID state



For details on recovery from an INVALID state refer to "Appendix F.1 Resolving Problems in the *PRIMECLUSTER Global Disk Services Configuration and Administration Guide.*"

22.7.11.3 Recreate the backup relationship

The configuration information that was saved by the process in Section 22.7.11.1 is used to recreate the backup relationship.



For details on the setting method of correlation, see "[22.7.6.2 A setup of correlation.](#)"

The following example uses the file `/var/tmp/relation_backup` to recreate the backup relationship between the backup source and the backup destination.

```
# sfcproxyjoin equivalent -r backup1_for_volume0001 -f /var/tmp/relation_backup
```

22.7.12 Notes regarding file system changes

22.7.12.1 Partition addition (sfcadd)

A partition can not be added to a file system that has an existing backup relationship.
Cancel the backup relationship before adding a partition.

22.7.12.2 File system deletion (sfcadm -D)

A file system with an associated backup relationship can not be deleted.
Cancel the backup relationship before deleting the file system.

Chapter 23 Performance and Tuning

This chapter describes how to use a variety of utilities to optimize and effectively use the GFS Shared File System.

23.1 Update Log

For high speed file system recovery, the GFS Shared File System writes to the disk an **update log**, which keeps a record of changes in the file system structure. Keeping an update log may give the impression that more I/O operations occur than in conventional file systems, but this is not true. As a result of reducing file meta-data I/O operations, a file system with this log has file system performance equal to or greater than that of file systems without the log. The reduction in operations occurs because changes are written as a batch and the reflection of update data is delayed. To receive these benefits, pay attention the following regarding the characteristics of the update log and actual file system access:

- Is the size of the **update log area** sufficient?
- Is the size of the update log area too large?
- Should the update log area be separated and placed in another **partition**?

The following sections consider each of these questions.

To change the settings for the update log area, use `sfcadm(1M)` to resize the area.

23.1.1 Is the size of the update log area sufficient?

The GFS Shared File System delays writing from the **update log buffer** in memory to the **update log area** on the disk as much as possible. This delayed writing operation batches pieces of update log information about the same meta-data into a single block of information, minimizing the number of write operations of the updated log information. For efficiency of the delayed writing of the log, a sufficient update log area must be reserved.

In general, reserve a 1 megabytes update log area for every 100 megabytes of a file system. (This is the default setting of `sfcnewfs(1M)`.)

The statistics command specific to the GFS Shared File System, `sfcstat(1M)`, can be used to determine the average number of updates processed resulting from the updating of meta-data on the disk.

If the file system performance is low and the average is close to 1, try increasing the update log area.

23.1.2 Is the size of the update log area too large?

If a system failure occurs, the GFS Shared File Systems performs high speed file recovery based on the information in the **update log area**. More specifically, it reads the **update log** information and reflects the contents of the update log in the **meta-data**. High speed file recovery depends on the size of the update log area.

If the time allowed for file system recovery is limited, the size of the update log area must be considered.

As a rough guide, assume that five seconds is required for each megabyte of the update log area.

With the standard setting of `sfcnewfs(1M)`, the size of the update log area is limited to 50 megabytes, regardless of the file system size.

23.1.3 Should the update log area be separated and placed in another partition?

The GFS Shared File System ensures that writing to the **update log area** does not degrade file system performance. Depending on the operating state, however, I/O bottlenecks may occur.

If I/O bottlenecks occur, create a partition for the update log on a disk with a relatively low I/O load. File system performance can be increased by separating the I/O operations related to the file system from those related to the update log area.

To check whether writing to the update log area causes I/O bottlenecks, use the statistics command specific to the GFS Shared File System, `sfcstat(1M)`. Determine the number of I/O operations issued to the update log area per second and the average wait time. This command can also be used to determine the number of I/O operations related to the entire file system.

23.2 Tuning Parameters

The **MDS**(Meta-Data Server) of the GFS Shared File System allows tuning of the following items:

- Amount of **extent-based** management information to be cached in memory(SFCFS_NCACHE_EXTENT)
- Amount of directory information to be cached in memory (SFCFS_NCACHE_DIRECTORY)
- Number of **i-nodes** on the disk to be cached in memory (SFCFS_NCACHE_INODE)

These values can be specified using `sfcadm(1M)`.

For each file system, an area with a size equal to the specified value x 1 kilobytes is reserved for the extent-based management information cache and for the directory information cache, and an area with a size equal to the specified value x 4 kilobytes is reserved for i-nodes cache. All of these areas are in the **MDS** process space.

By the default, two megabytes (2,048) of memory is allocated as extent management information cache, ten megabytes (10,240) of memory is allocated as directory information cache and eight megabytes (8,192) of memory is allocated as an i node cache.

If the size of a directory exceeds directory information cache size, the performance of the file creation to the directory or reference will fall. When a large directory may be created, it is strongly recommended that the amount of directory cache be set appropriately.

Use the statistics command specific to the GFS Shared File System, **sfcstat(1M)**, to determine the cache hit ratios of the above items and perform tuning if required.

(Example)

```
# sfcadm -o SFCFS_NCACHE_EXTENT=4096 /dev/sfdsk/gfs/rdisk/volume1
# sfcadm -o SFCFS_NCACHE_DIRECTORY=20480 /dev/sfdsk/gfs/rdisk/volume1
# sfcadm -o SFCFS_NCACHE_INODE=4096 /dev/sfdsk/gfs/rdisk/volume1
```

Chapter 24 Migration to the GFS Shared File System

This chapter describes how to migrate from existing file systems such as the UFS and GFS Local File Systems to GFS Shared File Systems.

24.1 Migration from a UFS file system

Note the following when migrating from an existing UFS file system to a GFS shared file system:



Note

- UFS **quota function** user limit values must be reset
- tar(1) must be used to migrate UFS files with an ACL
- When a UFS file is moved to a GFS shared file system, all holes in the file definition will have an area assigned in the GFS shared file system. The file no longer has any holes.

File system migration first requires that the files and directories on the UFS file system be saved in a separate location (for example, on a backup device or another file system). The new GFS shared file system can then be configured and the files and directories expanded. Use the following procedure to accomplish this:

1. Check that the UFS file system is not in use.
2. Save all the files on the UFS file system to a backup device or another file system using the appropriate command such as tar(1) and cpio(1).
3. Create the GFS shared file system, as described in [Section 19.2, "Creation"](#) and [Section 20.2, "Flow from file system creation to mount/unmount."](#)
4. Restore the saved files to the newly created GFS shared file system.

The following shows an example of migration on the same partition.

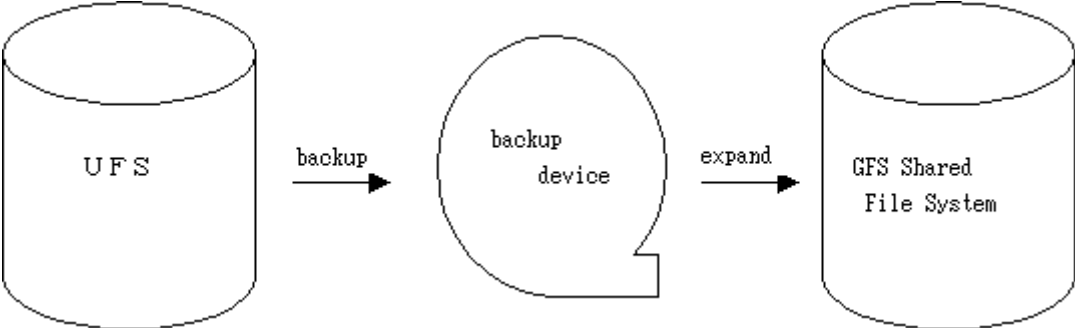
```
# cd /ufsmnt <- /ufsmnt is the mount point for the UFS file system from which to migrate.
# tar cvpf - . > /data/backup.tar <-Save files.
# cd /
# umount /ufsmnt
```

Perform the GDS setup and the setup of a management partition.

(For details on GDS setup, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide")

```
# sfcnewfs -H mikan,mango /dev/FJSVgdhd/rdisk/gdhd0s1 <- Configure the GFS shared file system.
# cd /sfdfs <- Move to the mount point of /sfdfs.
# tar xvf /data/backup.tar <- Restore the backup file to the GFS shared file system.
```

Figure Migration from the UFS File System



24.2 Migration from a GFS local file system

For file system migration, move files and directories from the GFS local file system to another location such as a backup device or another file system. After creating the new GFS shared file system, the files should be restored as shown below.

1. Check that this GFS local file system is not in use.
2. Move all the files on the GFS local file system to a backup device or another file system by executing `tar(1)`, `cpio(1)` and `sfxcpio(1)`.
3. Create the GFS shared file system on a disk, as described in ["19.2 Creation"](#), ["20.2 Flow from file system creation to mount/umount."](#)
4. Restore the saved files to the newly created GFS shared file system.



Note

- The user regulation values for the **quota function** on the GFS local file system need to be set.
- When the **ACL** function is used on a GFS local file system, the ACL information can be moved with `tar(1)`, `cpio(1)`, `sfxcpio(1)` and `sfxdump(1M)`.
- When the GFS Local File System extent property is set, the extent information can be moved with `sfxcpio(1)` or `sfxdump(1M)`. When the data is moved with `sfxcpio(1)`, restore it with `sfccpio(1)`. When it is moved with `sfxdump(1M)`, restore it with `sfxrestore(1M)`.
- When moving files on a GFS local file system to a GFS shared file system with `tar(1)`, `cpio(1)` and `sfxcpio(1)`, the files are defragmented because the additional memory that would be required if the files were found to be fragmented during restoration. Use `sfxdump(1M)` and `sfxrestore(1M)` if you wish to perform backup and restoration with the fragmented state maintained.

Appendix E List of Messages

This appendix explains the GFS Shared File System messages.

If a message has an explanation and response identical to another message, the description is omitted. In this case, see the explanation and response for the last message.

E.1 AC Messages of GFS Shared File System

This section explains AC messages of the GFS Shared File System that appear during operation. Messages are output to the console at each node and stored in `/var/adm/messages`.

E.1.1 Panic messages

This subsection explains the AC (kernel) panic message of the GFS Shared File System.

E.1.1.1 PANIC: `sfcfs:0015: mount_point: bad v_count == 0`

E.1.1.2 PANIC: `sfcfs:0016: mount_point: bad v_count == 0`

Explanation

An error in i-node management (`v_count`) of a file system (`mount_point`) was detected.

Response

Obtain a crash dumping, and contact local Customer Support.

E.1.1.3 PANIC: `sfcfs:0057: page_create == NULL`

Explanation

An error in i-node management (page assignment) of a file system was detected.

Response

Obtain a crash dumping, and contact local Customer Support.

E.1.1.4 PANIC: `sfcfs:0058: bn == SFCFS_HOLE`

E.1.1.5 PANIC: `sfcfs:0059: bn == SFCFS_HOLE`

Explanation

An error in i-node management (area management) of a file system was detected.

Response

Obtain a crash dumping, and contact local Customer Support.

E.1.1.6 PANIC: `sfcfs:0062: bn == SFCFS_HOLE and rw == S_WRITE`

Explanation

An error in i-node management (area management) of a file system was detected.

Response

Obtain a crash dumping, and contact local Customer Support.

E.1.1.7 PANIC: `sfcfs:0063: inode not in hash queue`

Explanation

An error in i-node management (hash management) of a file system was detected.

Response

Obtain a crash dumping, and contact local Customer Support.

E.1.1.8 PANIC: sfcfs:0064: *mount_point*: writes still outstanding, so panic to protect fs from writing by this node (*wrcnt=cnt*)

Explanation

The system was panicked because I/O by the indicated file system (*mount_point*) has not finished for a regular period of time.

Response

Obtain a crash dumping, and contact local Customer Support.

E.1.1.9 PANIC: sfcfs:0170: *mount_point*: could not reply to MDS (*proc=pnum*, *ino=inum*)

E.1.1.10 PANIC: sfcfs:0170: *mount_point*: could not reply to MDS (*proc=pnum*)

Explanation

An error was detected in the reply of AC procedure to MDS.

Response

Obtain a crash dumping and *fjsnap*, and contact local Customer Support.

E.1.2 Warning messages

This subsection explains the AC (kernel) warning message of the GFS Shared File System.

E.1.2.1 WARNING: **sfdfs:0005: *mount_point*: daemon create failed**

Explanation

A daemon could not be created during an attempt to mount a file system (*mount_point*).

Response

The memory load is thought as a cause. Please examine the increase of a real memory.

E.1.2.2 WARNING: **sfdfs:0006: *mount_point*: too many file systems.**

Explanation

The file system (*mount_point*) management table is full.

Response

Unmount unnecessary GFS Shared File Systems for the related node.

E.1.2.3 WARNING: **sfdfs:0070: *mount_point*: super block read error. dev = *dnum***

Explanation

Failed to read a super block on the device (device number is *dnum*) when mounting the file system (*mount_point*). Mount processing fails.

Response

Respond according to [Appendix G.2.1, "Action for I/O errors,"](#) in this manual.

E.1.2.4 WARNING: **sfdfs:0074: *mount_point*: primary and secondary super blocks are different. dev = *dnum***

Explanation

An inconsistency was detected between the contents of the primary super block and the contents of the secondary super block on the device (device number is *dnum*). Mounting *mount_point* fails.

Response

Display partition information by **fstyp_sfdfs(1M)** with -v option, and confirm to the file system composition errorless it. Check whether the configuration of the specified file system is normal. If it is abnormal, make the file system again. If it is normal, execute the **sfcdm(1M)** with no option to set up the information on file system configuration again, then retry mounting after recover the file system with **fsck_sfdfs(1M)**.

E.1.2.5 WARNING: **sfdfs:0075: *mount_point*: partition configuration read error. dev = *dnum***

Explanation

Failed to read partition configuration data on the device (device number is *dnum*) when mounting the file system (*mount_point*). Mount processing fails.

Response

Respond according to [Appendix G.2.1, "Action for I/O errors,"](#) in this manual.

E.1.2.6 WARNING: sfcfs:0078: *mount_point*: primary and secondary partition configuration are different. dev = *dnum***Explanation**

An inconsistency was detected between the primary partition configuration data and secondary partition configuration data on the device (device number is *dnum*). Mounting *mount_point* fails.

Response

Display partition information by **fstyp_sfcfs(1M)** with -v option, and confirm to the file system composition errorless it. Check whether the configuration of the specified file system is normal. If it is abnormal, make the file system again. If it is normal, execute the **sfcadm(1M)** with no option to set up the information on file system configuration again, then retry mounting after recover the file system with **fsck_sfcfs(1M)**.

E.1.2.7 WARNING: sfcfs:0080: *mount_point*: node information not found for *node***Explanation**

Information for the indicated *node* was not found in the file system (*mount_point*).

Response

Register correct node information using **sfcnode(1M)**.

E.1.2.8 WARNING: sfcfs:0081: *mount_point*: super blocks differ. dev *dnum* to *dnum***Explanation**

An inconsistency was found in the super block information in the indicated file system (*mount_point*) in a multiple partition configuration. Mounting fails.

Response

Display partition information by **fstyp_sfcfs(1M)** with -v option, and confirm to the file system composition errorless it. Check whether the configuration of the specified file system is normal. If it is abnormal, make the file system again. If it is normal, execute the **sfcadm(1M)** with no option to set up the information on file system configuration again, then retry mounting after recover the file system with **fsck_sfcfs(1M)**.

E.1.2.9 WARNING: sfcfs:0090: *mount_point*: reply timeout (proc=*pnum*, xid=*xid*)**Explanation**

A reply time-out occurred during communication with the meta-data server. The file system (*mount_point*) cannot be used.

Response

Unmounting that includes the file system of the node from which the message was issued. Determine the cause of the communication time-out. When an unmount processing fails, restart the node.

E.1.2.10 WARNING: sfcfs:0091: *mount_point*: connection failed to ip=*iaddr*, port=*pnum*, err=*ecode*

E.1.2.11 WARNING: sfcfs:0092: *mount_point*: connection failed to ip=*iaddr*, port=*pnum*, err=*ecode*

Explanation

A TCP/IP connection could not be established with the indicated IP address (*iaddr*) and indicated port number (*pnum*). Mounting the file system (*mount_point*) fails.

Response

Check the network, node, and IP address settings.

E.1.2.12 WARNING: sfcfs:0093: *mount_point*: server not found

Explanation

The server to be connected could not be found during meta-data server failure recovery. The file system (*mount_point*) cannot be used.

Response

Unmount the file system of the node from which the message was issued. The file system management data may be illegal. Check whether configuration data is valid using **sfcinfo(1M)**.

E.1.2.13 WARNING: sfcfs:0094: *mount_point*: server not specified

Explanation

No meta-data server was specified in a mount system call. Mounting the file system (*mount_point*) fails.

Response

Contact local Customer Support.

E.1.2.14 WARNING: sfcfs:0095: *mount_point*: protocol error (msg type *mtype*)

E.1.2.15 WARNING: sfcfs:0096: *mount_point*: protocol error (xid == 0)

E.1.2.16 WARNING: sfcfs:0097: *mount_point*: protocol error (not found corresponding request. list=*r*len, xid=*xid*)

E.1.2.17 WARNING: sfcfs:0098: *mount_point*: protocol error (reply stat *rstat*)

E.1.2.18 WARNING: sfcfs:0099: *mount_point*: protocol error (accept stat *astat*)

Explanation

An illegal message was received during communication with the meta-data server. The file system (*mount_point*) cannot be used.

Response

Unmount the file system of the node from which the message was issued. Check the network, node, and IP address settings.

E.1.2.19 WARNING: sfcfs:0100: *mount_point*: no response from server

Explanation

Communication with the meta-data server was lost. The file system (*mount_point*) cannot be used.

Response

Unmount the file system of the node from which the message was issued. Check the network, node, and IP address settings.

E.1.2.20 WARNING: sfcfs:0101: *mount_point*: message send failed. proc=*pnum*, err=*ecode*

Explanation

An error occurred during communication with the meta-data server of *mount_point*.

Response

Operation can be continued if there's no other error message.

E.1.2.21 WARNING: sfcfs:0103: *mount_point*: rpc daemon not exists (proc=*pnum*)

Explanation

Communication failed because a communication request was issued when no communication daemon for the indicated file system (*mount_point*) existed.

Response

Processing is not affected.

E.1.2.22 WARNING: sfcfs:0104: *mount_point*: unknown request *pnum*

Explanation

The indicated file system (*mount_point*) received an unexpected communication request.

Response

Processing is not affected.

E.1.2.23 WARNING: sfcfs:0105: *mount_point*: rpc daemon receive failed. err = *ecode*

Explanation

A receive error occurred on the communication path to the meta-data server of the indicated file system (*mount_point*). The file system cannot be used.

Response

Unmount the file system of the node from which the message was issued. Check the network, node, and IP address settings.

E.1.2.24 WARNING: sfcfs:0106: *mount_point*: No messages are available to be received

Explanation

A receive error occurred on the communication path to the meta-data server of the indicated file system (*mount_point*).

Response

The GFS Shared File System attempts automatic recovery. Unless another error message is issued, the file system can be used the way it is.

E.1.2.25 WARNING: sfcfs:0107: *mount_point*: request timeout (proc=*pnum*)

Explanation

An error occurred during communication with the meta-data server of the indicated file system (*mount_point*).

Response

The GFS Shared File System attempts automatic recovery. Unless another error message is issued, the file system can be used the way it is.

E.1.2.26 WARNING: sfcfs:0108: *mount_point*: mount failed. (*pname* procedure failed) sfcfs_error = *ecode*

Explanation

An error occurred during communication with MDS of the file system (*mount_point*). Mount processing fails.

Response

Unmount the file system at the node at which this message appeared. Confirm the setting of the network, node, and IP address.

E.1.2.27 WARNING: sfcfs:0109: *mount_point*: UNMOUNT procedure failed. sfcfs_error = *err*

Explanation

An error occurred during communication with MDS of the file system (*mount_point*). No process is affected.

Response

Processing is not affected.

E.1.2.28 WARNING: sfcfs:0110: invalid parameter length *len*

Explanation

The length (*len*) of a parameter in a mount system call is invalid. Mounting fails.

Response

Contact local Customer Support.

E.1.2.29 WARNING: sfcfs:0111: *mount_point*: invalid recover

Explanation

A system call for meta-data server failure recovery contains an invalid parameter. The file system (*mount_point*) cannot be used.

Response

Unmount the indicated file system, and then mount it again.

E.1.2.30 WARNING: sfcfs:0112: *mount_point*: recover failed for secondary. (error in rpcd ecode)**Explanation**

An error occurred in a system call for meta-data server failure recovery of *mount_point*.

Response

This message indicates a failure in secondary server recovery processing. There is no problem in using the file system. The file system can be used the way it is.

E.1.2.31 WARNING: sfcfs:0113: *mount_point*: recover failed for primary. (error in rpcd ecode)**Explanation**

An error occurred in a system call for meta-data server failure recovery of *mount_point*.

Response

Recovery processing of the primary meta-data server failed. The GFS Shared File System attempts automatic recovery. Unless another error occurs, the file system can be used the way it is.

E.1.2.32 WARNING: sfcfs:0114: *mount_point*: recover failed. (nprocs = *nprocs*)**E.1.2.33 WARNING: sfcfs:0115: *mount_point*: recover failed. (MOUNT procedure err=ecode)****Explanation**

An error occurred in a system call for meta-data server failure recovery of the file system (*mount_point*).

Response

The GFS Shared File System attempts automatic recovery. Unless another error message is issued, the file system can be used the way it is.

E.1.2.34 WARNING: sfcfs:0116: *mount_point*: recover failed. (invalid MOUNT response, sfi_file: *file_handle*, mt_file: *file_handle*)**E.1.2.35 WARNING: sfcfs:0117: *mount_point*: recover failed. (invalid MOUNT response, sfi_blksize = *blksize*, mt_blksize = *blksize*)****E.1.2.36 WARNING: sfcfs:0118: *mount_point*: recover failed. (invalid MOUNT response, sfi_tsize = *tsize*, mt_tsize = *tsize*)****E.1.2.37 WARNING: sfcfs:0119: *mount_point*: recover failed. (invalid MOUNT response, sfi_flags = *flags*, mt_flags = *flags*)****E.1.2.38 WARNING: sfcfs:0120: *mount_point*: recover failed. (invalid MOUNT**

response, sfi_extpower = *extpower*, mt_extpower = *extpower*)

E.1.2.39 WARNING: sfcfs:0121: *mount_point*: recover failed. (invalid MOUNT response, sfi_extlevel = *extlevel*, mt_extlevel = *extlevel*)

E.1.2.40 WARNING: sfcfs:0122: *mount_point*: recover failed. (invalid MOUNT response, sfi_volinfo_len = *len*, mt_volinfo_len = *len*)

E.1.2.41 WARNING: sfcfs:0123: *mount_point*: recover failed. (invalid MOUNT response, volinfo[*valid*] mismatch)

E.1.2.42 WARNING: sfcfs:0124: *mount_point*: recover failed. (BARSYNC(*flags*) time-out)

E.1.2.43 WARNING: sfcfs:0125: *mount_point*: recover failed. (bs_cmnd(*flags*) != *expected_flags*)

E.1.2.44 WARNING: sfcfs:0126: *mount_point*: recover failed. (resend error *ecode*)

E.1.2.45 WARNING: sfcfs:0127: *mount_point*: recover failed. (resend timeout, rest=*nprocs*)

E.1.2.46 WARNING: sfcfs:0128: *mount_point*: recover failed. (resend error *ecode*)

E.1.2.47 WARNING: sfcfs:0129: *mount_point*: recover failed. (UMOUNT procedure) err= *ecode*

E.1.2.48 WARNING: sfcfs:0130: *mount_point*: recover failed. (RECOVER procedure) ino=*inum*, err=*ecode*

E.1.2.49 WARNING: sfcfs:0131: *mount_point*: recover failed. (RECOVER procedure) ino=*inum*, sfcfs_err=*ecode*

E.1.2.50 WARNING: sfcfs:0132: *mount_point*: recover failed. (resend error *ecode*)

E.1.2.51 WARNING: sfcfs:0133: *mount_point*: recover failed. (resend timeout, rest=*nprocs*)

E.1.2.52 WARNING: sfcfs:0134: *mount_point*: recover failed. (BARSYNC(*flags*) procedure) err= *ecode*

E.1.2.53 WARNING: sfcfs:0135: *mount_point*: recover failed. (FSCTL(*flags*) procedure) err= *ecode***Explanation**

An error occurred in a system call for meta-data server failure recovery. The indicated file system (*mount_point*) cannot be used from the related node.

Response

Unmount the indicated file system, and check the file system configuration and the network, node, and IP address settings.

E.1.2.54 WARNING: sfcfs:0136: *mount_point*: recover failed. drvname is not found. (major= *0xnum*)**Explanation**

An inconsistency was detected during a recovery processing on the file system (*mount_point*).The inconsistency will be resolved and the processing will continue.

Response

No response is required.

E.1.2.55 WARNING: sfcfs:0139: *mount_point*: volinfo is different. (invalid VOLINFO, volinfo[*num*] mismatch)**E.1.2.56 WARNING: sfcfs:0139: *mount_point*: volinfo is different. (invalid VOLINFO, sfi_volinfo_len = *num*, vi_volinfo_len = *num*)****E.1.2.57 WARNING: sfcfs:0139: *mount_point*: volinfo is different. (invalid VOLINFO, volid = *num*, sfi_volinfo_val[*num*].vli_volid = *num*)****Explanation**

Detected a mismatch in control table volinfo. An error preventing the continuation of processing is detected, therefore, the file system (*mount_point*) is shut down.

Response

Unmount the file system in all nodes. Confirm the setting of the file system configuration, network, node, and IP address.

E.1.2.58 WARNING: sfcfs:0140: *mount_point*: cannot access: too many I/O errors. dev = *dnum***Explanation**

The file system (*mount_point*) is shut down because of a super block read error on the device (device number is *dnum*.) Access to the file system will cause an error return.

Response

Respond according to [Appendix G.2.1, "Action for I/O errors,"](#) in this manual.

E.1.2.59 WARNING: sfcfs:0200: *mount_point*: daemon create failed

E.1.2.60 WARNING: sfcfs:0201: *mount_point*: daemon create failed

Explanation

Daemon creation during mounting of the file system (*mount_point*) is incomplete.

Response

The memory load is thought as a cause. Please examine the increase of a real memory.

E.1.2.61 WARNING: sfcfs:0204: *mount_point*: l=*inum* mds down rreload garbage collect

E.1.2.62 WARNING: sfcfs:0205: *mount_point*: l=*inum* mds down rarchive garbage collect

Explanation

Error occurs in MDS communications for the file (*inum*) in the file system (*mount_point*).

Response

Operation can be continued if there's no other error message.

E.1.2.63 WARNING: sfcfs:0221: *mount_point*: recover failed. sfcfs_hsm_recover error=*ecode*

Explanation

An error has occurred in a system call for MDS down recovery. The file system (*mount_point*) cannot be used from the node.

Response

Unmount the file system in all nodes, and check the settings of the file system configuration/network, node and IP address.

E.1.2.64 WARNING: sfcfs:0250: *mount_point*: mpathd daemon create failed

Explanation

Activation of the daemon for the file system (*mount_point*) to control multiple hosts is incomplete. Mounting is going to fail.

Response

The memory load is thought as a cause. Please examine the increase of a real memory.

E.1.2.65 WARNING: sfcfs:0252: address family not specified

Explanation

Invalid network information has been specified. Mounting is going to fail.

Response

Check the network settings.

E.1.2.66 WARNING: sfcfs:0253: address family (0xnum) not supported

Explanation

Address family of the specified network is not supported. Mounting is going to fail.

Response

Change the network settings to use applicable network address family.

E.1.2.67 WARNING: sfcfs:0254: unknown address family (0xnum)

Explanation

Address family of the specified network is not supported. Mounting is going to fail.

Response

Change the network settings to use the correct network address family.

E.1.2.68 WARNING: sfcfs:0255: server address check failed

Explanation

The self-node TCP/IP setup data cannot be obtained. Mounting is going to fail.

Response

Mount the file system again.

E.1.2.69 WARNING: sfcfs:0256: *mount_point*: no mpath heartbeat response from server MDS=(*addr=hostname*, *port=num*)

Explanation

There's no heartbeat response from MDS on *hostname*.

Response

In the node that output messages, check whether the file system is capable of continuing operation. When it is not capable of continuing, check the network connection and meta data server conditions. After restoring, mount the file system in the node that output messages again.

E.1.2.70 WARNING: sfcfs:0259: *mount_point*: mpath connection wasn't established MDS=(*addr=hostname*, *port=num*), *error=error*

Explanation

Re-connection to the MDS on *hostname* is incomplete.

Response

In the node that output messages, check whether the file system is capable of continuing operation. When it is not capable of continuing, check the network connection and meta data server conditions. After restoring, restart the file system in the node that output messages again.

**E.1.2.71 WARNING: sfcfs:0260: *mount_point*: mpath connect was failed
MDS=(addr=*hostname*, port=*num*), error=*error***

Explanation

Connection to the MDS on *hostname* is incomplete.

Response

Check the network connection and meta data server conditions. After restoring, mount the file system in the node that output messages again.

**E.1.2.72 WARNING: sfcfs:0261: *mount_point*: mpath connection was broken
MDS=(addr=*hostname*, port=*num*), error=*error***

Explanation

There's no heartbeat response from the MDS on *hostname*.

Response

In the node that output messages, check whether the file system is capable of continuing operation. When it is not capable of continuing, check the network connection and meta data server conditions. After restoring, mount the file system in the node that output messages again.

**E.1.2.73 WARNING: sfcfs:0262: *mount_point*: bad mpath seqno ac(*num1*)
mds(*num2*)**

Explanation

An inconsistency has arisen in the sequence number management between MDS and AC.

Response

In the node that output messages, check whether the file system is capable of continuing operation. When it is not capable of continuing, check the network connection and meta data server conditions. After restoring, mount the file system in the node that output messages again.

E.1.2.74 WARNING: sfcfs:0263 invalid version number *version*

Explanation

Invalid version information specified for argument of mount system call. An inconsistency occurs between the file system management commands and AC versions. Mounting is going to fail.

Response

Check that a package is applied properly to all the nodes.

E.1.2.75 WARNING: sfcfs:0264: *mount_point*: switchover failed. error=*ecode*

Explanation

Host exchange in the multiple LAN function of the file system (*mount_point*) is incomplete.

Response

In the node that output messages, check whether the file system is capable of continuing operation. When it is not capable of continuing, check the network connection and meta data server conditions. After restoring, mount the file system in the node that output messages again.

E.1.3 Information messages

This section explains AC (kernel) information messages of the GFS Shared File System.

E.1.3.1 NOTICE: **sfcfs:0001: *mount_point*: file system full**

Explanation

There is no more file system (*mount_point*) data area.
(For a non-privileged user, this message is also output when space drops below the minfree limit.)

Response

Expand the file system, delete unnecessary files, or move files to a file system that has enough space.
To expand the file system, use **sfcadd(1M)**.

E.1.3.2 NOTICE: **sfcfs:0050: setting sfcfs_ninode to max value of *max_value***

Explanation

The value set for **sfcfs_ninode** was changed to *max_value* because it exceeded the specified value.

Response

No response is required because the system makes the correction and continues processing. After specifying the **sfcfs_ninode** value in **/etc/system**, be sure to check it.

E.1.3.3 NOTICE: **sfcfs:0051: freeing wrong file system by umount *fsid***

Explanation

File system management detected release of the wrong file system (*fsid*.)

Response

No response is required because the system makes the correction and continues processing.

E.1.3.4 NOTICE: **sfcfs:0071: *mount_point*: primary super block read failed, but secondary read. dev = *dnum* block = *blknum* size = *size***

Explanation

The primary super block on the device (device number is *dnum*) could not be read. Processing was continued because the secondary super block was read correctly.

Response

File system operation can be continued. However, because a block error occurred, promptly take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.1.3.5 NOTICE: **sfcfs:0072: *mount_point* : secondary super block read failed, but primary read. dev = *dnum* block = *blknum* size = *size***

Explanation

The secondary super block on the device (device number is *dnum*) could not be read. Processing was continued because the primary super block was read correctly.

Response

File system operation can be continued. However, because a block error occurred, promptly take action

according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.1.3.6 NOTICE: sfcfs:0076: *mount_point*: primary partition configuration read failed, but secondary read. dev = *dnum* block = *blknum* size = *size*

Explanation

Primary partition configuration data on the device (device number is *dnum*) could not be read. Processing was continued because the secondary partition configuration data was read correctly.

Response

File system operation can be continued. However, because a block error occurred, promptly take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.1.3.7 NOTICE: sfcfs:0077: *mount_point*: secondary partition configuration read failed, but primary read. dev = *dnum* block = *blknum* size = *size*

Explanation

Secondary partition configuration data on the device (device number is *dnum*) could not be read. Processing was continued because the primary partition configuration data was read correctly.

Response

File system operation can be continued. However, because a block error occurred, promptly take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.1.3.8 NOTICE: sfcfs:0079: *mount_point*: nodename mismatch. *node_name* <-> *node_name*

Explanation

A node with an existing host id and nonexistent node name was registered in the file system (*mount_point*).

Response

Unless message sfcfs:0080 is issued, no response is required.

E.1.3.9 NOTICE: sfcfs:0084: *mount_point*: cannot mmap. because the file is currently being opened on other node. ino=*inum*, flags=*flags*, prot=*prot*, pid=*pid*

Explanation

The specified file (*inum*) on the file system (*mount_point*) to **mmap(2)** is currently being **open(2)** on an other node.

The file cannot be mapped to maintain the consistency of the data.

Response

Please wait for the termination of the process running on the *node1* indicated in the message "sfcfs_mds:0112", and execute again.

Some processes work correctly even if this message and the message "sfcfs_mds:0112" were issued. In this case, these messages can be ignored.

E.1.3.10 NOTICE: sfcfs:0137: *mount_point*: recover failed. (ADDVOL(1) procedure) err= *ecode*, devname= *special***Explanation**

An error is detected during recovery for the file system (*mount_point*).

Response

No response is required.

E.1.3.11 NOTICE: sfcfs:0138: *mount_point*: addvol failed. (ADDVOL(0xnum) procedure) err = *ecode***Explanation**

An error is detected during online addition of partition(s) on *mount_point*.

Response

Respond according to a message displayed by **sfcadd(1M)**. When **sfcadd(1M)** terminates abnormally, check the environment and re-execute the **sfcadd(1M)**.

E.1.3.12 NOTICE: sfcfs:0141: *mount_point*: cannot open. because the file is currently being mapped on other node. ino=*inum*, flags=*flags*, pid=*pid***Explanation**

The file to be opened is under **mmap(2)** on another node. To ensure data consistency, the file cannot be opened.

Response

Wait until the application on the node appearing on *node1* of message sfcfs_mds:0112 has terminated, then re-execute.

E.1.3.13 NOTICE: sfcfs:0142: *mount_point*: over file hard limit (pid *pid*, uid *uid*)**Explanation**

The number of the files which user *unum* owns on *mount_point* was reached to the hard limit value of the number of files set up by the quota function. User *unum* cannot own a file any more.

Response

In order for user *unum* to enable create a new file, please delete unnecessary files which user *unum* owns on *mount_point* until the number of files is less than the soft limit value and hard limits value set by quota function.

E.1.3.14 NOTICE: sfcfs:0143: *mount_point*: Warning: too many files (pid *pid*, uid *uid*)**Explanation**

The number of files which user *unum* owns on *mount_point* exceeds the soft limit value set by quota function.

Response

In order for user *unum* to enable create a new file, please delete unnecessary files which user *unum*

owns on *mount_point* until the number of files is less than the soft limit value and hard limits value set by quota function.

E.1.3.15 NOTICE: sfcfs:0144: *mount_point*: over file and time limit (pid *pid*, uid *uid*)

Explanation

The number of the files which user *unum* owns on *mount_point* was not less than the soft limit value in the time limit set up by the quota function, after it exceeding the soft limit value. User *unum* cannot create a new file until it is less than a soft limit value.

Response

In order for user *unum* to enable create a new file, please delete unnecessary files which user *unum* owns on *mount_point* until the number of files is less than the soft limit value and hard limits value set by quota function.

E.1.3.16 NOTICE: sfcfs:0145: *mount_point*: over hard disk limit (pid *pid*, uid *uid*, inum *inum*)

Explanation

The total of the number of use blocks of a file which user *unum* owns on *mount_point* reached to the hard limit value of the number of blocks set up by the quota function. User *unum* cannot use a block any more.

Response

In order for user *unum* to enable allocate a new block, please delete unnecessary files which user *unum* owns on *mount_point* until the number of use blocks is less than the soft limit value and hard limits value set by quota function.

E.1.3.17 NOTICE: sfcfs:0146: *mount_point*: Warning: over disk limit (pid *pid*, uid *uid*, inum *inum*)

Explanation

The total of the number of use blocks of a file which user *unum* owns on *mount_point* reached to the soft limit value of the number of blocks set up by the quota function.

Response

In order for user *unum* to enable allocate a new block, please delete unnecessary files which user *unum* owns on *mount_point* until the number of use blocks is less than the soft limit value and hard limits value set by quota function.

E.1.3.18 NOTICE: sfcfs:0147: *mount_point*: over disk and time limit (pid *pid*, uid *uid*, inum *inum*)

Explanation

The total of the number of use blocks of a file which user *unum* owns on *mount_point* was not less than the soft limit value in the time limit set up by the quota function, after it exceeding the soft limit value. User *unum* cannot get a new block until it is less than a soft limit value.

Response

In order for user *unum* to enable allocate a new block, please delete unnecessary files which user *unum* owns on *mount_point* until the number of use blocks is less than the soft limit value and hard limits value set by quota function.

E.1.3.19 NOTICE: sfcfs:00251: *mount_point*: spath duplicate. name=*hostname*

Explanation

The same data (*hostname*) is repeatedly specified in the host specification for the file system (*mount_point*.) Operation continues by excluding invalid entries.

Response

Check that the multiple LAN specification is done properly.

E.1.3.20 NOTICE: sfcfs:00257: *mount_point*: mpath connection was closed for switch MDS=(*addr=hostname, port=num*)

Explanation

Communication path to MDS is closed temporarily by a change command for the file system (*mount_point*.)

Response

No countermeasures need to be applied.

E.1.3.21 NOTICE: sfcfs:00258: *mount_point*: mpath connection was re-established

Explanation

Re-connection to MDS for the file system (*mount_point*) is completed.

Response

No countermeasures need to be applied.

E.2 GFS Shared File System Driver messages

This section explains the GFS Shared File System driver messages. Messages are output to the console at each node and stored in `/var/adm/messages`.

E.2.1 Information messages

E.2.1.1 NOTICE: sfcfsdrv:0001: *mount_point*: cancel migrate session (*ecode*)

E.2.1.2 NOTICE: sfcfsdrv:0002: *mount_point*: cancel reload session (*ecode*)

E.2.1.3 NOTICE: sfcfsdrv:0003: *mount_point*: cancel hsmlock session (*ecode*)

E.2.1.4 NOTICE: sfcfsdrv:0004: *mount_point*: cancel async request (*ecode*)

E.2.1.5 NOTICE: sfcfsdrv:0005: *mount_point*: cancel sync request (*ecode*)

E.2.1.6 NOTICE: sfcfsdrv:0006: *mount_point*: cancel reload (*ecode*)

E.2.1.7 NOTICE: sfcfsdrv:0007: *mount_point*: lost session(*mnum*, *inum*) in RARCHIVE

E.2.1.8 NOTICE: sfcfsdrv:0008: *mount_point*: lost session(*mnum*, *inum*) in SVCREG

E.2.1.9 NOTICE: sfcfsdrv:0009: *mount_point*: lost session(*mnum*, *inum*) in RRELOAD

E.2.1.10 NOTICE: sfcfsdrv:0010: *mount_point*: lost session(*mnum*, *inum*) in HSMUNLOCK

E.2.1.11 NOTICE: sfcfsdrv:0011: *mount_point*: petaterm (*ecode*)

E.2.1.12 NOTICE: sfcfsdrv:0012: *mount_point*: getfsinfo (*ecode*)

Explanation

An error occurred in the file system (*mount_point*) driver.

Response

Operation can be continued provided no other error message is output.

E.3 GFS Shared File System Daemon messages

This section explains the GFS Shared File System daemon messages.

E.3.1 Panic messages

This section explains the panic message of each GFS Shared File System daemon.

E.3.1.1 sfcprmd daemon

E.3.1.1.1. sfcfsdrv:0015: Semaphore(*semid=id*) access error, panic for keep data integrity of GFS and continue cluster operation.

Explanation

Operation of a semaphore (*id*) went wrong and it changed into the state where the management continuation of the GFS Shared File System on this node cannot be carried out. Although the panic of this node is carried out for data protection, GFS Shared File System management is continued by other nodes.

Response

Extract a material for investigation by fjsnap and contact local Customer Support.

E.3.1.1.2. sfcfsdrv:0015: sfcprmd(*pid=pid*) was down, panic for keep data integrity of GFS and continue cluster operation.

Explanation

sfcprmd daemon was downed and it changed into the state where the management continuation of the GFS Shared File System on this node cannot be carried out. Although the panic of this node is carried out for data protection, GFS Shared File System management is continued by other nodes.

Response

Extract a material for investigation by fjsnap and contact local Customer Support.

E.3.1.1.3. sfcfsdrv:0015: *daemon(pid=pid)* was down, panic for keep data integrity of GFS and continue cluster operation.

Explanation

The daemon shown by *daemon* was downed and it changed into the state where the management continuation of the GFS Shared File System on this node cannot be carried out. Although the panic of this node is carried out for data protection, GFS Shared File System management is continued by other nodes.

Response

Extract a material for investigation by fjsnap and contact local Customer Support.

E.3.1.1.4. sfcfsdrv:0015: Panic for keep data integrity of GFS and continue cluster operation.

Explanation

Abnormalities occurred and it changed into the state where the management continuation of the GFS Shared File System on this node cannot be carried out. Although the panic of this node is carried out for data protection, GFS Shared File System management is continued by other nodes.

Response

Extract a material for investigation by fjsnap and contact local Customer Support.

E.3.2 Error messages

This section explains the error messages of each GFS Shared File System daemon.

E.3.2.1 sfcfrmd daemon

E.3.2.1.1. ERROR:sfcfrmd:0001: not superuser

Explanation

sfcfrmd daemon is not performed by super user

Response

Please start sfcfrmd daemon by **sfcfrmstart(1M)** command as a super user.

E.3.2.1.2. ERROR:sfcfrmd:0002: associated with a terminal device

Explanation

The sfcfrmd daemon is direct performed.

Response

Please start sfcfrmd daemon by **sfcfrmstart(1M)** command as a super user.

E.3.2.1.3. ERROR:sfcfrmd:0003: cannot read *pathname*, or something wrong

Explanation

Since there was injustice or file reading of *pathname* went wrong, sfcfrmd daemon starting went wrong.

Response

Please check whether the file of *pathname* is set up correctly.

E.3.2.1.4. ERROR:sfcfrmd:0004: cannot get node version *string*

Explanation

The sfcfrmd daemon failed to start, because a working file *string* in **/var/opt/FJSVsfcs** was broken.

Response

Make sure that **/var/opt** directory has a space by **df(1M)** command and re-start the node.

E.3.2.1.5. ERROR:sfcfrmd:0005: something wrong in *pathname*

Explanation

The sfcfrmd daemon failed to start, because **/etc/services** has no entry of **sfcsrm** or *pathname* is broken.

Response

Make sure that each **/etc/services** file has the entry "**sfcsrm 9200/tcp** in all the nodes in the cluster. If not, add it. For details of this file, see the **services(4)** of the "*Solaris X Reference Manual Collection.*"

If the setting above is right, *pathname* seems broken. Collect the diagnostic data with **fjsnap** and contact local Customer Support.

E.3.2.1.6. ERROR:sfcfrmd:0006: cannot allocate memory space

Explanation

Since acquisition of a memory went wrong, sfcfrmd daemon starting went wrong.

Response

Please check the state of a system, and wait for the end of other processes, or increase swap space and reboot the node.

E.3.2.1.7. ERROR:sfcfrmd:0007: cannot start service

Explanation

Failed to start the sfcfrmd daemon.

Response

Obtain a core dumping of sfcfrmd daemon by **gcore(1)** in all the nodes (for example, "**/usr/bin/ps -e | grep sfcfrmd**" to get the process id and "**/usr/bin/gcore process_id**"). In addition to this, collect the diagnostic data with fjsnap and contact local Customer Support.

E.3.2.1.8. ERROR:sfcfrmd:0008: I/O error or data corruption of management partition: *pathname*

Explanation

sfcfrmd daemon failed in reading and writing for the management partition (*pathname*.)

Response

Please check whether a setup as a management partition domain of the GFS Shared File System is right, or a disk operates normally.

E.3.2.1.9. ERROR:sfcfrmd:0009: something wrong in cluster settings

Explanation

Since a cluster was not started, sfcfrmd daemon starting went wrong.

Response

Please start a cluster.

E.3.2.1.10. ERROR:sfcfrmd:0010: cannot get host name

Explanation

The CIP address registered in the management partition changed.

Response

By remaking and restoring the management partition, back the configuration of the partition to a previous one before a value of the CIP address changed. For details to remake and restore, see [Section 16.5, "Backup and restore of the management partition information."](#)

E.3.2.1.11. ERROR:sfcfrmd:0011: cannot connect to sfcprmd

Explanation

The sfcfrmd daemon failed to connect the sfcprmd daemon.

Response

Make sure that the sfcprmd daemon is running by **ps(1)** command (for example, "**/usr/bin/ps -e | grep sfcprmd**"). If the sfcprmd daemon was stopped by the unload dependency script, restart sfcprmd by the load script ("**/opt/SMAW/SMAWcf/dep/start.d/S81sfcs load**"). Otherwise, re-start the node.

If this message appeared when sfcprmd daemon is running, collect the investigation material using the fjsnap command and contact local Customer Support.

E.3.2.1.12. ERROR:sfcfrmd:0012: cannot connect to sfchnsd

Explanation

The sfcfrmd daemon failed to connect the sfchnsd daemon.

Response

Make sure that the sfchnsd daemon is running by **ps(1)** command (for example, `/usr/bin/ps -e | grep sfchnsd`). If the sfchnsd daemon was stopped by the unload dependency script, restart sfchnsd by the load script (`/opt/SMAW/SMAWcf/dep/start.d/S81sfdfs load`). Otherwise, re-start the node.

If this message appeared when sfchnsd daemon is running, collect the investigation material using the fjsnap command and contact local Customer Support.

E.3.2.1.13. ERROR:sfcfrmd:0013: cannot activate sfchnsd

Explanation

sfchnsd daemon is not activated.

Response

Obtain a core dumping of sfchnsd daemon by **gcore(1)** (for example, `/usr/bin/ps -e | grep sfchnsd` to get the process id and `/usr/bin/gcore process_id`). In addition to this, collect the diagnostic data with fjsnap and contact local Customer Support.

E.3.2.1.14. ERROR:sfcfrmd:0014: cannot get service port

Explanation

A service port is not able to be acquired.

Response

Make sure that each `/etc/services` file has the entry `sfcfsrm 9200/tcp` in all the nodes in the cluster. If not, add it. For details of this file, see the **services(4)** of the *Solaris X Reference Manual Collection*.

E.3.2.2 MDS(sfcfsmsg) daemon

E.3.2.2.1. ERROR: sfcfs_mds:0001: can't execute MDS for *mount_point*, errcode=ecode

E.3.2.2.2. ERROR: sfcfs_mds:0002: can't execute MDS for *special*

E.3.2.2.3. ERROR: sfcfs_mds:0003: can't execute MDS

Explanation

The meta-data server could not be started.

Response

Contact local Customer Support.

E.3.2.2.4. ERROR: sfcfs_mds:0004: can't execute primary MDS for *mount_point*, errcode=ecode

Explanation

The primary meta-data server for the file system (*mount_point*) could not be started.

Response

Contact local Customer Support.

E.3.2.2.5. ERROR: sfcfs_mds:0005: can't execute secondary MDS for *mount_point*, errcode=ecode

Explanation

The secondary meta-data server for the file system (*mount_point*) could not be started.

Response

Contact local Customer Support.

E.3.2.2.6. ERROR: sfcfs_mds:0006: can't take over MDS for *mount_point*, errcode=ecode

Explanation

The meta-data server for the file system (*mount_point*) switch processing failed.

Response

Contact local Customer Support.

E.3.2.2.7. ERROR: sfcfs_mds:0007: primary MDS for *mount_point* terminated, errcode=ecode

Explanation

The primary meta-data server for the file system (*mount_point*) terminated abnormally.

Response

Contact local Customer Support.

E.3.2.2.8. ERROR: sfcfs_mds:0008: secondary MDS for *mount_point* terminated, errcode=ecode

Explanation

The secondary meta-data server for the file system (*mount_point*) terminated abnormally.

Response

Contact local Customer Support.

E.3.2.2.9. ERROR: sfcfs_mds:0021: *mount_point*: illegal operation: op= code

Explanation

An illegal operation was detected during processing in a file system (*mount_point*.)

Response

Obtain a core dumping, and contact local Customer Support.

E.3.2.2.10. ERROR: sfcfs_mds:0021: can't allocate memory in primary MDS for *mount_point*, errcode=ecode

Explanation

Cannot allocate memory in primary MDS for the file system (*mount_point*.) Memory of system is insufficient.

Response

Increase the swap space or real memory.

E.3.2.2.11. ERROR: sfcfs_mds:0022: can't allocate memory in secondary MDS for *mount_point*, errcode=ecode

Explanation

Cannot allocate memory in secondary MDS for the file system (*mount_point*.) Memory of system is insufficient.

Response

Increase the swap space or real memory.

E.3.2.2.12. ERROR: sfcfs_mds:0023: can't allocate memory in taking over MDS for *mount_point*, errcode=ecode

Explanation

Cannot allocate memory in taking over of MDS for the file system (*mount_point*.) Memory of system is insufficient.

Response

Increase the swap space or real memory.

E.3.2.2.13. ERROR: sfcfs_mds:0025: *mount_point*: invalid slot status code

Explanation

An inconsistency occurred during processing in the file system (*mount_point*.)

Response

Obtain a core dumping, and contact local Customer Support.

E.3.2.2.14. ERROR: sfcfs_mds:0031: MDS for *mount_point* terminated because logreplay failed

Explanation

The meta-data server for the file system (*mount_point*) log replay failed.

Response

Contact local Customer Support.

E.3.2.2.15. ERROR: sfcfs_mds:0052: v_count != 0 ino_pointer inum

Explanation

An error was detected in a released i-node (*inum*) in a file system.

Response

Obtain a core dump and contact local Customer Support.

E.3.2.2.16. ERROR: sfcfs_mds:0053: mount_point: illegal transaction buffer. bufno = number

E.3.2.2.17. ERROR: sfcfs_mds:0054: mount_point: illegal transaction buffer. bufno = number

E.3.2.2.18. ERROR: sfcfs_mds:0055: mount_point: buffer overflow. size = size

Explanation

An error detected in update log management of file system (*mount_point*.)

Response

Obtain a core dump and contact local Customer Support.

E.3.2.2.19. ERROR: sfcfs_mds:0056: mount_point: hold count is invalid. type = tnum number = num

Explanation

An error was detected in the file system data reference count.

Response

Obtain a core dump and contact local Customer Support.

E.3.2.3 sfcfsd daemon

E.3.2.3.1. ERROR: sfcfsd: 0001: gfs monitor daemon for mountpoint(*mount_point*) abort: *detail*

E.3.2.3.2. ERROR: sfcfsd: 0002: gfs monitor daemon for fsid(*fsid*) abort: *detail*

E.3.2.3.3. ERROR: sfcfsd: 0003: gfs monitor daemon abort: *detail*

Explanation

The GFS Shared File System monitoring daemon ended abnormally. The error explained in *detail* occurred.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.3.4. ERROR: sfcfsd: 0004: can't execute primary MDS: mountpoint(*mount_point*)

Explanation

Starting up the primary MDS for the file system (*mount_point*) has failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.3.5. ERROR: sfcfsd: 0005: can't execute secondary MDS: mountpoint(*mount_point*)

Explanation

Start up of the secondary MDS for the file system (*mount_point*) has failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.3.6. ERROR: sfcfsd: 0006: can't take over MDS: mountpoint(*mount_point*)

Explanation

Switchover of the MDS for the file system (*mount_point*) failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.3.7. ERROR: sfcfsd: 0007: primary MDS down: mountpoint(*mount_point*)

Explanation

The primary MDS for the file system (*mount_point*) has failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.3.8. ERROR: sfcfsd: 0008: secondary MDS down: mountpoint(*mount_point*)

Explanation

The secondary MDS for the file system (*mount_point*) has failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.3.9. ERROR: sfcfsd: 0009: can't mount to primary MDS. gfs monitor daemon stop: mountpoint(*mount_point*)

Explanation

Mounting on the primary MDS for the file system (*mount_point*) failed. The GFS Shared File System monitoring daemon terminates.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.3.10. ERROR: sfcfsd: 0010: can't mount to secondary MDS: mountpoint(*mount_point*)

Explanation

Mounting on the secondary MDS for the file system (*mount_point*) failed.

Response

There is no problem with the operation of the file system. Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.3.11. ERROR: sfcfsd: 0011: blockade: mountpoint(*mount_point*)

Explanation

The file system (*mountpoint*) has been closed.

Response

The target mount point cannot be used and only unmounting can be performed. Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.4 sfcfs_mount command

This section explains the sfcfs_mount command messages that is called sfcfsd daemon.

E.3.2.4.1. ERROR: sfcfs_mount: 0001: *mount_point*: not superusr

Explanation

sfcfs_mount is not performed by super user.

Response

Do not execute the module directly.

E.3.2.4.2. ERROR: sfcfs_mount: 0002: *block_special* no such device

E.3.2.4.3. ERROR: sfcfs_mount: 0003: *mount_point* not a directory or a component of *block_special* is not a directory

E.3.2.4.4. ERROR: sfcfs_mount: 0004: *block_special* or *mount_point*, no such file or directory

E.3.2.4.5. ERROR: sfcfs_mount: 0005: *block_special* is not this fstype.

Explanation

Find setting error in the */etc/vfstab*. Therefore, mount the file system will be failed.

Response

Correct the entry in */etc/vfstab*. Moreover, try mount again.

E.3.2.4.6. ERROR: sfcfs_mount: 0006: *block_special* is already mounted, *mount_point* is busy, or the allowable number of mount points has been exceeded

Explanation

Some process is used *mount_point* or already mounted at *mount_point*. Therefore, mount the file system will be failed.

Response

Please check the process that is used *mountpoint* by **fuser(1M)**. Or check the file system that is used *mount_point* by **mount(1M)**.

E.3.2.4.7. ERROR: sfcfs_mount: 0007: *block_special* not a block device

Explanation

Find setting error in the */etc/vfstab*. Therefore, mount the file system will be failed.

Response

Correct the entry in */etc/vfstab*. Moreover, try mount again.

E.3.2.4.8. ERROR: sfcfs_mount: 0008: *block_special* write-protected

Explanation

block_special will be set write-protected Therefore, mount the file system will be failed.

Response

Change *block_special* setting to read and write mode.

E.3.2.4.9. ERROR: sfcfs_mount: 0010: large files may be present on *block_special*, and it was attempted to be mounted nolargefiles**Explanation**

Files that have 2 gigabytes over size are included in the file system, when the mount option nolargefiles specified Therefore, mount the file system will be failed.

Response

Change *block_special* setting to read and write mode.

E.3.2.4.10. ERROR: sfcfs_mount: 0011: *mount_point*: invalid argument**Explanation**

Internal error occurred. Therefore, mount the file system will be failed.

Response

Please install same version level and same patch ID of GFS Shared File System in all nodes. If there are same, collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.4.11. ERROR: sfcfs_mount: 0013: *mount_point*: CFS protocol error**Explanation**

Internal error occurred during communication with the meta-data server of *mount_point*. Therefore, mount the file system will be failed.

Response

Please install same version level and same patch ID of GFS Shared File System in all nodes. If there are same, collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.4.12. ERROR: sfcfs_mount: 0014: *mount_point*: response from MDS and information of AC are mismatch**Explanation**

Internal error found in response from meta-data server. Therefore, mount the file system will be failed.

Response

Please install same version level and same patch ID of GFS Shared File System in all nodes. If there are same, collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.4.13. ERROR: sfcfs_mount: 0015: *mount_point*: cannot connect to Secondary-MDS**E.3.2.4.14. ERROR: sfcfs_mount: 0016: *mount_point*: cannot connect to Primary-MDS****Explanation**

Access Client cannot connect to meta-data server. Therefore, mount the file system will be failed.

Response

Check meta-data server down. In this case, try mount again.

E.3.2.4.15. ERROR: sfcfs_mount: 0019: cannot mount *block_special*: *errmsg* [errno=no]**Explanation**

Internal error occurred. Therefore, mount the file system will be failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.4.16. ERROR: sfcfs_mount: 0033: Cannot get port number of sfcfs-*n* for [P-MDS|S-MDS], not found in /etc/services

E.3.2.4.17. ERROR: sfcfs_mount: 0034: Cannot get port number sfcfs-*n* for [P-MDS|S-MDS], not found in /etc/services

Explanation

The port number sfcfs-*n* in **/etc/services** is not found. Therefore, mount the file system will be failed.

Response

Restore definition of sfcfs-*n* in **/etc/services**.

E.3.2.4.18. ERROR: sfcfs_mount: 0035: Length of host address *z* for host must be *n*

Explanation

Internal error occurred. Therefore, mount the file system will be failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.4.19. ERROR: sfcfs_mount: 0036: Cannot get host address of *host* for [P-MDS|S-MDS], not found in /etc/hosts

Explanation

Cannot get the host address of *host*. Therefore, mount the file system will be failed.

Response

Check host name (*host*) definition. Or change definition host name of the file system by **sfcadm(1M)**.

E.3.2.5 sfcpcnd daemon

E.3.2.5.1. ERROR: sfcpcnd:0001: Permission denied. Abort.

Explanation

Execution was attempted with other than super user permission.

Response

Do not execute the module directly.

E.3.2.5.2. ERROR: sfcpcnd:0002: Initialize fail. Abort.

Explanation

sfcpcnd daemon failed booting.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.5.3. ERROR: sfcpcnd:0003: Cannot create the semaphore. Abort.

Explanation

Creation of the semaphore failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.5.4. ERROR: sfcpcnd:0004: Fatal error. Abort.

Explanation

sfcpcnd daemon terminated abnormally.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.6 sfcprmd daemon

E.3.2.6.1. ERROR: sfcprmd:0001: Permission denied. Abort.

Explanation

Execution was attempted with other than super user permission.

Response

Do not execute the module directly.

E.3.2.6.2. ERROR: sfcprmd:0002: Initialize failed. Abort.

Explanation

sfcprmd daemon failed booting.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.6.3. ERROR: sfcprmd:0003: Cannot open /dev/null. Abort.

Explanation

Opening /dev/null failed.

Response

Check the system environment.

E.3.2.6.4. ERROR: sfcprmd:0004: Cannot access to the /var/opt/FJSVsfdfs/.sfcprmd_uds. Abort.

Explanation

Accessing /var/opt/FJSVsfdfs/.sfcprmd_uds failed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.2.6.5. ERROR: sfcprmd:0005: Memory not enough. Abort.

Explanation

There is insufficient memory to allow sfcprmd daemon to boot.

Response

Increase the swap space or the real memory.

E.3.2.6.6. ERROR: sfcprmd:0007: Cannot access to the semaphore of sfcpcnd. Abort.

Explanation

The semaphore created by sfcpcnd daemon cannot be accessed.

Response

Collect the diagnostic data with fjsnap and then contact local Customer Support.

E.3.3 Warning messages

This section explains the warning message of each daemon of the GFS Shared File System.

E.3.3.1 MDS (sfcfsmg) daemon

E.3.3.1.1. WARNING: sfcfs_mds:0005: *mount_point*: daemon create failed

Explanation

A daemon could not be created when mount the file system (*mount_point*). Memory of system is insufficient.

Response

Increase the swap space or the real memory.

E.3.3.1.2. WARNING: sfcfs_mds:0006: *mount_point*: too many file systems.

Explanation

The file system management table is full (*mount_point*).

Response

In the all nodes, unmount unnecessary GFS Shared File Systems.

E.3.3.1.3. WARNING: sfcfs_mds:0021: can't allocate memory in primary MDS for *mount_point*

Explanation

Memory could not be allocated on the primary meta-data server for the file system (*mount_point*). Memory of system is insufficient.

Response

Increase the swap space or the real memory.

E.3.3.1.4. WARNING: sfcfs_mds:0022: can't allocate memory in secondary MDS for *mount_point*

Explanation

Memory could not be allocated on the secondary meta-data server for the file system (*mount_point*). Memory of system is insufficient.

Response

Increase the swap space or the real memory.

E.3.3.1.5. WARNING: sfcfs_mds:0022: *mount_point*: bad dir ino *inum* at offset *off*: *info*

Explanation

An error was detected in a directory entry of the file system (*mount_point*).

Response

Unmount the file system in all node, and execute **fsck_sfcs(1M)** with **-o nolog** specified.

E.3.3.1.6. WARNING: sfcfs_mds:0023: can't allocate memory in taking over MDS for *mount_point*

E.3.3.1.7. WARNING: sfcfs_mds:0023: can't allocate memory in taking over MDS for *mount_point*, *errcode=ecode*

Explanation

Memory could not be allocated during meta-data server for the file system (*mount_point*) switch processing. Memory of system is insufficient.

Response

Increase the swap space or the real memory.

E.3.3.1.8. WARNING: sfcfs_mds:0023: *mount_point*: target directory link count < 2 ino = *inum*

Explanation

An error was detected in directory management data of the file system (*mount_point*).

Response

Unmount the file system in all node, and execute **fsck_sfcfs(1M)** with **-o nolog** specified.

E.3.3.1.9. WARNING: sfcfs_mds:0024: can't allocate memory for *buf_name* (size *size*, *errcode ecode*) in logreplay: *mount_point*

Explanation

Memory could not be allocated during update log replay for the file system (*mount_point*.) Memory of system is insufficient.

Response

Increase the swap space or the real memory.

E.3.3.1.10. WARNING: sfcfs_mds:0024: *mount_point*: invalid offset *off* ino = *inum*

Explanation

An error was detected in a directory entry of the file system (*mount_point*).

Response

Unmount the file system in all node, and execute **fsck_sfcfs(1M)** with **-o nolog** specified.

E.3.3.1.11. WARNING: sfcfs_mds:0027: *mount_point*: bad dir, ino = *inum*

Explanation

An error was detected in an entry in a directory of the file system (*mount_point*).

Response

Unmount the file system, and execute **fsck_sfcfs(1M)** with **-o nolog** specified.

E.3.3.1.12. WARNING: sfcfs_mds:0030: *mount_point*: meta data read error. type = *number* dev = *dnum* block = *block_num* size = *size*

E.3.3.1.13. WARNING: sfcfs_mds:0031: *mount_point*: meta data read error. type = *number* dev = *dnum* block = *block_num* size = *size*

Explanation

An error occurred during an attempt to read the meta-data area of the file system (*mount_point*) from a device (device number is *dnum*.)

Response

Take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.3.3.1.14. WARNING: sfcfs_mds:0032: *mount_point*: bitmap read error. type = *number* dev = *dnum* block = *block_num* size = *size***Explanation**

An error occurred during an attempt to read the bitmap area of the file system (*mount_point*) from a device (device number is *dnum*.)

Response

Take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.3.3.1.15. WARNING: sfcfs_mds:0033: *mount_point*: bitmap read error. type = *number* dev = *dnum* block = *block_num* size = *size***Explanation**

An error occurred during an attempt to read the bitmap area of the file system (*mount_point*) from a device (device number is *dnum*.)

Response

Take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.3.3.1.16. WARNING: sfcfs_mds:0034: *mount_point*: bitmap read error. type = *number* dev = *dnum* block = *block_num* size = *size***Explanation**

An error occurred during an attempt to read the bitmap area of the file system (*mount_point*) from a device (device number is *dnum*.) The released area was made invalid.

Response

Take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.3.3.1.17. WARNING: sfcfs_mds:0035: *mount_point*: bitmap write error. information was lost. type = *number* dev = *dnum* block = *block_num* size = *size***Explanation**

An error occurred during an attempt to write to the bitmap area of the file system (*mount_point*) on a device (device number is *dnum*.) Part of the file system (*mount_point*) information was lost.

Response

Take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.3.3.1.18. WARNING: sfcfs_mds:0036: *mount_point*: bitmap write error. type = *number* dev = *dnum* block = *block_num* size = *size***Explanation**

An error occurred during an attempt to write to the bitmap area of the file system (*mount_point*) on a device (device number is *dnum*.)

Response

Unless message sfcfs_mds:0035 or sfcfs_mds:0039 is issued, recovery by retry processing was successful, so no response is required.

**E.3.3.1.19. WARNING: sfcfs_mds:0037: *mount_point*: meta data write error. type = *number*
dev = *dnum* block = *block_num* size = *size***

Explanation

An error occurred during an attempt to write to the meta-data area of the file system (*mount_point*) on the device (device number is *dnum*.)

Response

Unless message sfcfs_mds:0040 is issued, recovery by retry processing was successful, so no response is required.

**E.3.3.1.20. WARNING: sfcfs_mds:0038: *mount_point*: fail to update super block at first time.
dev = *dnum* block = *block_num* size = *size***

Explanation

Super blocks of the file system (*mount_point*) could not be updated during mount processing. Mounting fails.

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

**E.3.3.1.21. WARNING: sfcfs_mds:0039: *mount_point*: bitmap write error. information was
lost. type = *number* dev = *dnum* block = *block_num* size = *size***

Explanation

An error occurred during an attempt to write to the bitmap area of the file system (*mount_point*) on a device (device number is *dnum*.) Part of the file system (*mount_point*) information was lost.

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

**E.3.3.1.22. WARNING: sfcfs_mds:0040: *mount_point*: meta data write error. information was
lost. type = *number* dev = *dnum* block = *block_num* size = *size***

Explanation

An error occurred during an attempt to write to the meta-data area of the file system (*mount_point*) on a device (device number is *dnum*.) Part of the file system (*mount_point*) information was lost.

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.3.1.23. WARNING: sfcfs_mds:0042: *mount_point*: LOG daemon is dead. dev = *dnum*

Explanation

The log daemon stopped processing because it could not recover from a failure to write to the update log area of the file system (*mount_point*) on a device (device number is *dnum*.)

Response

Because of an I/O error, the update log daemon was not able to operate and terminated. The file system access performance is degraded because operation is continued without the update log available. Back up the file system, and take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

**E.3.3.1.24. WARNING: sfcfs_mds:0043: *mount_point*: super block write error. dev = *dnum*
block = *block_num* size = *size*****Explanation**

An error occurred during an attempt to write a super block of the file system (*mount_point*) to a device (device number is *dnum*.)

Response

Unless message sfcfs_mds:0044 is issued, recovery was performed by retry processing, so no response is required.

E.3.3.1.25. WARNING: sfcfs_mds:0044: *mount_point*: super block write error. information was lost. dev = *dnum* block = *block_num* size = *size***Explanation**

An error occurred during an attempt to write a super block of the file system (*mount_point*) to a device (device number is *dnum*.)

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.3.1.26. WARNING: sfcfs_mds:0046: *mount_point*: cannot access: too many I/O errors. dev = *dnum***Explanation**

A file system was shut down because of a super block write error. If this file system (*mount_point*) is accessed subsequently, an error is returned.

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.3.1.27. WARNING: sfcfs_mds:0063: *mount_point*: cannot delete extent. ino = *inum* len = *len* width = *width* length = *len***Explanation**

An error in extent-based management of the file (*inum*) on the file system (*mount_point*) was detected.

Response

Unmount the file system in all nodes, and execute **fsck_sfcfs(1M)** with **-o nolog** specified.

E.3.3.1.28. WARNING: sfcfs_mds:0064: *mount_point*: no extent exists. ino = *inum* osize = *len* len = *len***Explanation**

An error in extent-based management of the file (*inum*) on the file system (*mount_point*) was detected.

Response

Unmount the file system, and execute **fsck_sfcfs(1M)** with **-o nolog** specified.

E.3.3.1.29. WARNING: sfcfs_mds:0065: *mount_point*: cannot delete extent of directory. ino = *inum* len = *len* width = *width* length = *len***Explanation**

An error in extent-based management of the file (*inum*) on the file system (*mount_point*) was detected.

Response

In all the nodes, unmount the indicated file system, and execute **fsck_sfcfs(1M)** with **-o nolog** specified.

E.3.3.1.30. WARNING: sfcfs_mds:0070: *mount_point*: super block read error. dev = *dnum*

Explanation

A super block could not be read from device (device number is *dnum*) when the mount the file system (*mount_point*) started.

Response

Unmount the indicated file system, and execute **fsck_sfcfs(1M)** with **-o nolog** specified.

E.3.3.1.31. WARNING: sfcfs_mds:0074: *mount_point*: primary and secondary super blocks are different. dev = *dnum*

Explanation

An inconsistency was detected between the contents of the primary super block and the contents of the secondary super block on the device (device number is *dnum*.)

Response

Check whether the specified file system configuration is normal. If it is normal, set the file system configuration data again using **sfcadm(1M)**, perform file system recovery using **fsck_sfcfs(1M)**, and mount file system again.

E.3.3.1.32. WARNING: sfcfs_mds:0075: *mount_point*: partition configuration read error. dev = *dnum*

Explanation

Partition configuration data could not be read when mount the file system (*mount_point*) started.

Response

Take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.3.3.1.33. WARNING: sfcfs_mds:0078: *mount_point*: primary and secondary super blocks are different. dev = *dnum*

Explanation

An inconsistency was detected between the contents of the primary super block and the contents of the secondary super block on the device (device number is *dnum*.)

Response

In the case that the file system is restored and not adjusted file system configuration by **sfcadm(1M)**, set the file system configuration data again using **sfcadm(1M)**, perform file system recovery using **fsck_sfcfs(1M)**, and mount file system again.

E.3.3.1.34. WARNING: sfcfs_mds:0080: *mount_point*: node information not found for *node*

Explanation

Information for *node* was not found in the file system (*mount_point*).

Response

Register correct node information using **sfcnode(1M)**.

E.3.3.1.35. WARNING: sfcfs_mds:0081: *mount_point*: super blocks differ. dev *dnum* to *dnum***Explanation**

An inconsistency was found in the super block information in the indicated file system (*mount_point*) in a multiple partition configuration.

Response

Check whether the specified file system configuration is normal. If it is normal, set the file system configuration data again using **sfcadm(1M)**, perform file system recovery using **fsck_sfcfs(1M)**, and mount the file system again.

E.3.3.1.36. WARNING: sfcfs_mds:0082: *mount_point*: partition configuration differ. dev *dnum* to dev *devlist_num***Explanation**

An inconsistency was found in the super block information in the indicated file system (*mount_point*) in a multiple partition configuration.

Response

In the case that the file system is restored and not adjusted file system configuration by **sfcadm(1M)**, set the file system configuration data again using **sfcadm(1M)**, perform file system recovery using **fsck_sfcfs(1M)**, and mount the file system again.

E.3.3.1.37. WARNING: sfcfs_mds:0112: mmap can't access between other nodes. mnt=*mount_point*, ino=*inum*, nodename(*node1*, *node2*), mode(*mode1*, *mode2*)**Explanation**

open(2) and **mmap(2)** that cannot be maintain the consistency of the data was issued to the same file.

Response

Please wait for the termination of the process running on the "*node1*" indicated this message, and execute again.

To determine the process using the file (*inum*), act according to the following sequence:

- 1) determine the filename of the *i-number*, using **find(1)** with **-inum** option.
- 2) determine the process ID using the filename, using **fuser(1M)**.
- 3) determine the command name of the process ID, using **ps(1)**.

E.3.3.1.38. WARNING: sfcfs_mds:0131: *mount_point*: super block write error. dev = *dnum* block = *block_num* size = *size***Explanation**

An error occurred during an attempt to write a super block to a device (device number is *dnum*.)

Response

Take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.3.3.1.39. WARNING: sfcfs_mds:0133: *mount_point*: volume information write error. dev = *dnum* block = *block_num* size = *size***Explanation**

An error occurred during an attempt to write a super block to a device (device number is *dnum*.)

Response

Take action according to the instructions in [Appendix G.2.1, "Action for I/O errors."](#)

E.3.3.1.40. WARNING: sfcfs_mds:0133: read error on log partition (errno = *eno* errcode = *ecode*): *mount_point*

Explanation

An error occurred during an attempt to read the log partition of the file system (*mount_point*.)

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.3.1.41. WARNING: sfcfs_mds:0133: *mount_point*: metamap write error. dev = *dnum* block = *blknum* size = *size*

Explanation

An error occurred during an attempt to write the meta partition of the file system (*mount_point*.)

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.3.1.42. WARNING: sfcfs_mds:0134: read error on meta partition (errno = *eno* errcode = *ecode*): *mount_point*

Explanation

An error occurred during an attempt to read the meta partition of the file system (*mount_point*.)

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.3.1.43. WARNING: sfcfs_mds:0135: write error on log partition (errno = *eno* errcode = *ecode*): *mount_point*

Explanation

An error occurred during an attempt to write the log partition of the file system (*mount_point*.)

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.3.1.44. WARNING: sfcfs_mds:0136: write error on meta partition (errno = *eno* errcode = *ecode*): *mount_point*

Explanation

An error occurred during an attempt to write the meta partition of the file system (*mount_point*.)

Response

Take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.3.1.45. WARNING: sfcfs_mds:0141: can't receive heartbeat procedure from *hostname*, *mount_point* = *mount_point*

Explanation

Communication with the AC was lost. The file system (*mount_point*) cannot be used from the node with the indicated host name.

Response

Unmount the indicated file system on the node having the host name in the message. Check the network, node, and IP address settings.

E.3.3.1.46. WARNING: sfcfs_mds:0160: *mount_point*: Bad file system. Remaking file system is necessary

Explanation

An error of the file system (*mount_point*) was detected. The mount operation will be failed to care the file system data.

Response

Collect the diagnostic data with `fjsnap` and then contact local Customer Support.

E.3.3.1.47. WARNING: sfcfs_mds:0160: *mount_point*: illegal reserve extent was found, vol = *vol* block = *block_num* size = *size*

Explanation

An error in extent-based management on the file system (*mount_point*) was detected.

Response

In all the nodes, unmount the file system in all node, and execute `fsck_sfcfs(1M)` with `-o nolog` specified.

E.3.3.1.48. WARNING: sfcfs_mds:0161: *mount_point*: illegal free extent was found, vol = *vol* block = *block_num* size = *size*

Explanation

An error in extent-based management on the file system (*mount_point*) was detected.

Response

In all the nodes, unmount the file system in all node, and execute `fsck_sfcfs(1M)` with `-o nolog` specified.

E.3.3.2 sfcfs_mount command

This section explains the sfcfs_mount command messages that is called sfcfsd daemon.

E.3.3.2.1. WARNING: sfcfs_mount: 0017: *mount_point*: communicate to MDS from AC failed.

Explanation

Access Client cannot communicate to meta-data server. So, retry again.

Response

No response is required.

E.3.3.2.2. WARNING: sfcfs_mount: 0018: *mount_point*: not enough memory

Explanation

Cannot get enough kernel memory. Therefore, mount the file system (*mount_point*) will be failed.

Response

The memory load is thought as a cause. Please examine the increase of a real memory.

E.3.3.2.3. WARNING: sfcfs_mount: 0037: *mount_point*: communicate to MDS from AC failed.

Explanation

Access Client cannot communicate to meta-data server. So, retry again.

Response

No response is required.

E.3.3.2.4. WARNING: sfcfs_mount: 0028: *mount_point*: quota option is not match for MDS information, so retry mount with the opposite option now.

Explanation

Cannot validate the quota function of the file system (*mount_point*), so going to invalidate the quota function.

Response

Validate the quota function with **sfcquotaon(1M)**.

E.3.3.2.5. WARNING: sfcfs_mount: 0029: *mount_point*: retry mount failed too.

Explanation

Access Client cannot communicate to meta-data server. So, retry again.

Response

No response is required.

E.3.4 Information messages

This section explains the information messages of each GFS Shared File System daemon.

E.3.4.1 sfcfrmd daemon

E.3.4.1.1. NOTICE:sfcfrmd:0001: serve as master

Explanation

sfcfrmd daemon was started as a master.

Response

None.

E.3.4.1.2. NOTICE:sfcfrmd:0002: serve as slave

Explanation

sfcfrmd daemon was started as a slave.

Response

None.

E.3.4.1.3. NOTICE:sfcfrmd:0003: service start

Explanation

Service of sfcfrmd daemon was started.

Response

None.

E.3.4.1.4. NOTICE:sfcfrmd:0004: service stop

Explanation

Service of sfcfrmd daemon was completed.

Response

None.

E.3.4.2 MDS (sfcfsmsg) daemon

E.3.4.2.1. NOTICE: sfcfs_mds:0001: *mount_point*: file system full

Explanation

There is no more space in file data area of the file system (*mount_point*).
(For an ordinary user, this message is also output when space drops below the **minfree** limit.)

Response

Expand the file system, delete unnecessary files, or move files to a file system that has enough space.
To expand the file system, use **sfcadd(1M)**.

E.3.4.2.2. NOTICE: sfcfs_mds:0002: *mount_point*: out of inodes

Explanation

There is no more space in I-node area of the file system (*mount_point*).

Response

Expand the file system, delete unnecessary files or move files to a file system that has enough space.
To expand the file system, use **sfcadd(1M)**.

E.3.4.2.3. NOTICE: sfcfs_mds:0003: *mount_point*: file system meta data full (Vdata) type = *number*

Explanation

There is no more space in V-data area of the file system (*mount_point*).

Response

Expand the file system, delete unnecessary files or directories, or move files to a file system that has enough space. To expand the file system, use **sfcadd(1M)**.

E.3.4.2.4. NOTICE: sfcfs_mds:0004: *mount_point*: too much fragment (extent)

Explanation

There is no more area for managing unused space in the file system (*mount_point*) because of area fragmentation. An area released by subsequent area release processing may become an invalid area.

Response

Back up the file system, and then re-create and restore it to unfragment the fragmented areas in the file system.

E.3.4.2.5. NOTICE: sfcfs_mds:0010: illegal mode (recovered): *mount_point* / *inum* mode = *file_mode*

E.3.4.2.6. NOTICE: sfcfs_mds:0011: illegal free inode (recovered): *mount_point* / *inum* had size *size*

E.3.4.2.7. NOTICE: sfcfs_mds:0012: illegal mode (recovered): *mount_point* / *inum* mode = *file_mode*

E.3.4.2.8. NOTICE: sfcfs_mds:0013: illegal inode (recovered): *mount_point* / *inum*

E.3.4.2.9. NOTICE: sfcfs_mds:0014: illegal inode (recovered): *mount_point* / *inum* new size = *size* blocks = *blocks*

Explanation

An error was detected in the i-node in a file system (*mount_point*), but the i-node recovered automatically from the error.

Response

No response is required because the system makes the correction and continues processing.

E.3.4.2.10. NOTICE: sfcfs_mds:0020: *mount_point*: bad namelen == 0 parent ino = *inum***E.3.4.2.11. NOTICE: sfcfs_mds:0026: *mount_point*: bad namelen == 0 parent ino = *inum*****Explanation**

An illegal path name request was made in a directory operation for the file system (*mount_point*.)

Response

No response is required because the system rejected the illegal request and continues processing. If this error occurs frequently, identify the target directory using parent inode number (*inum*) and the operation making the illegal request, and take appropriate action.

E.3.4.2.12. NOTICE: sfcfs_mds:0041: *mount_point*: I/O error on log: dev = *dnum* block = *block_num* size = *size* error = *enum***Explanation**

The system could not write an update log of the file system (*mount_point*) on the device (device number is *dnum*.)

Response

Unless message sfcfs_mds:0042 is issued, recovery by retry processing was successful, so no response is required.

E.3.4.2.13. NOTICE: sfcfs_mds:0050: setting sfxfs_ninode to max value of *max_inode***Explanation**

A value that exceeded the specified range was originally specified in **sfcfs_ninode**, but this value was changed appropriately.

Response

No response is required because the system corrects the value and continues processing. If the value of **sfcfs_ninode** was specified in **/etc/system**, review the value.

E.3.4.2.14. NOTICE: sfcfs_mds:0051: freeing wrong file system by umount *fsid***Explanation**

The file system management detected an attempt to release the wrong file system.

Response

No response is required because the system corrects the error and continues processing.

E.3.4.2.15. NOTICE: sfcfs_mds:0060: *mount_point*: read only mount, but modified (recovered):**Explanation**

An error was detected in the file system (*mount_point*) but was resolved automatically.

Response

No response is required because the system continues processing after correcting the error automatically.

E.3.4.2.16. NOTICE: sfcfs_mds:0071: *mount_point*: primary super block read failed, but secondary read. dev = *dnum* block = *block_num* size = *size*

Explanation

The primary super block of the file system (*mount_point*) on the device (device number is *dnum*) could not be read. Processing was continued because the secondary super block was read correctly.

Response

File system operation can be continued. However, because a block error occurred, promptly take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.4.2.17. NOTICE: sfcfs_mds:0072: *mount_point*: secondary super block read failed, but primary read. dev = *dnum* block = *block_num* size = *size*

Explanation

The secondary super block of the file system (*mount_point*) on the device (device number is *dnum*) could not be read. Processing was continued because the primary super block was read correctly.

Response

File system manipulation can be continued. However, because a block error occurred, promptly take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.4.2.18. NOTICE: sfcfs_mds:0073: *mount_point*: VTOC read error. dev = *dnum*

Explanation

The size of the specified partition on the device (device number is *dnum*) could not be obtained. Processing was continued because the primary super block was read correctly.

Response

Unmount the file system in all nodes, and make the super blocks consistent using **sfcadm(1M)**.

E.3.4.2.19. NOTICE: sfcfs_mds:0076: *mount_point*: primary partition configuration read failed, but secondary read. dev = *dnum* block = *block_num* size = *size*

Explanation

Primary partition configuration data of the file system (*mount_point*) on the device (device number is *dnum*) could not be read. Processing was continued because the secondary partition configuration data was read correctly.

Response

File system operation can be continued. However, because a block error occurred, promptly take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.4.2.20. NOTICE: sfcfs_mds:0077: *mount_point*: secondary partition configuration read failed, but primary read. dev = *dnum* block = *block_num* size = *size*

Explanation

Secondary partition configuration data of the file system (*mount_point*) on the device (device number is *dnum*) could not be read. Processing was continued because the primary partition configuration data was read correctly.

Response

File system operation can be continued. However, because a block error occurred, promptly take action according to the instructions in [Appendix G.2.1, " Action for I/O errors."](#)

E.3.4.2.21. NOTICE: sfcfs_mds:0079: *mount_point*: nodename mismatch. *node* <-> *node*

Explanation

A node with an existing hostid and nonexistent node name was registered in the file system (*mount_point*).

Response

Unless sfcfs_mds:0080 is issued, no response is required.

E.3.4.2.22. NOTICE: sfcfs_mds:0125: *mount_point*: recover this file system by excuting update log replay. special=*special*

Explanation

The file system (*mount_point*) is recovered using update log.

Response

No response is required.

E.3.4.2.23. NOTICE: sfcfs_mds:0151: illegal free inode (recovered) *mount_point*/*inum* had blocks *blocks*

Explanation

An error was detected i-node managements of the file system (*mount_point*) but was resolved automatically.

Response

No response is required because the system makes the correction and continues processing.

E.3.4.3 sfcprmd daemon

E.3.4.3.1. INFO: sfcprmd:0006: *daemon* was respawn fail.

Explanation

daemon of GFS Shared File System finished abnormally.

Response

Collect the diagnostic data with fjsnap and contact local Customer Support.

E.4 Command messages for File System Common Management

This section explains the messages that are displayed when file system common management command is used.

E.4.1 df_sfcfs command

E.4.1.1 df: can not get hostid

Explanation

The host ID cannot be referenced.

Response

Contact local Customer Support.

E.4.1.2 df: cannot open *block_special*

Explanation

Special file *block_special* in the specified file system could not be opened.

Response

Check whether the special file *block_special* exists in the system.

If the logical volume of the GDS is used, information is displayed only when the logical volume of GDS is activated. If the logical volume is not activated, retry after activating it.

If the GFS management view is used for operation, retry after activating the logical volume.

E.4.1.3 df: cannot find mnttab entry for *special*

Explanation

The entry for the specified special file indicated by *special* was not found in the */etc/mnttab* file.

Response

Check whether the specified special file name indicated by *special* is correct.

E.4.1.4 df: cannot find mount point for *mount_point*

Explanation

The entry for the specified mount point (*mount_point*) was not found in the */etc/mnttab* or */etc/vfstab* file.

Response

Check whether the specified mount point name indicated by *mount_point* is correct. If *mount_point* is true, Correct the entry in */etc/vfstab*. Moreover, try mount again.

E.4.1.5 df: '-g' and '-o i' are mutually exclusive

Explanation

The **-g** and **-o i** options are mutually exclusive.

Response

Specify either option, and re-execute the command.

E.4.1.6 df: invalid magic word 0xmagic

Explanation

The partition configuration data of a special file configuring the specified file system is invalid.

Response

Check whether the specified file system is the GFS Shared File System.

E.4.1.7 ioctl: ...

Explanation

`ioctl(2)` for special file *special* in the specified file system could not be executed.

Response

Contact local Customer Support.

E.4.1.8 df: llseek:

Explanation

`llseek(2)` for special file *special* in the specified file system could not be executed.

Response

Contact local Customer Support.

E.4.1.9 df: *mount_point*:

Explanation

`statvfs(2)` for mount point (*mount_point*) could not be executed.

Response

Check whether the GFS Shared File System is mounted at the mount point indicated by *mount_point* and whether the specified mount point name is correct.

E.4.1.10 df: *mount_point* not an sfcfs file system

Explanation

The GFS Shared File System is not mounted at the specified mount point indicated by *mount_point*.

Response

Check whether the specified mount point name indicated by *mount_point* is correct.

E.4.1.11 df: '-o v' and '-o i' are mutually exclusive

Explanation

The `-o v` and `-o i` options are mutually exclusive.

Response

Specify either option, and re-execute the command.

E.4.1.12 df: /etc/mnttab:

Explanation

The `/etc/mnttab` file could not be opened and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.4.1.13 df: out of memory!

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.4.1.14 df: premature EOF on *special*

Explanation

An unexpected EOF was detected during an attempt to read special file *special*.

Response

Contact local Customer Support.

E.4.1.15 df: read error on *special*: bno = *block_number*

Explanation

A read error occurred in block *block_number* in special file *special*.

Response

Contact local Customer Support.

E.4.1.16 df: read super-block error

E.4.1.17 df: read:

Explanation

The super block of a special file in the specified file system could not be read.

Response

Contact local Customer Support.

E.4.1.18 df: file system configuration information not found.

Explanation

The file system configuration information cannot be gotten.

Response

Restore the file system configuration information. For detail, see [chapter 16, "Management Partition."](#)

E.4.1.19 df: *special*:

Explanation

Special file *special* could not be opened.

Response

Check whether special file *special* exists in the system and whether the specified special file name is correct.

E.4.2 fsck_sfdfs command

For the **fsck_sfdfs(1M)** error messages, see [Section 21.2.1, "Action to be taken if fsck terminates abnormally."](#)

E.4.3 mkfs_sfdfs command

E.4.3.1 mkfs_sfdfs: ERROR: {Node|Partiton|Meta|Log|Data} is not same

Explanation

The specified configuration is not the same as the configuration of the previously created file system.

Response

When specifying the **-o force** option, specify the same configuration as that of the previously created file system and re-execute the command.

E.4.3.2 mkfs_sfdfs: ERROR: Argument out of range: *value*

Explanation

The value (*value*) specified for an option was too large.

Response

Specify a correct value within the range allowed for each option, and re-execute the command.

E.4.3.3 mkfs_sfdfs: ERROR: Bad numeric arg: "*string*"

Explanation

The nonnumeric character string (*string*) was specified for an option parameter for which a numeric value must be specified.

Response

Specify a correct parameter value.

E.4.3.4 mkfs_sfdfs: ERROR: Cannot acquire lock: *type(num)*

Explanation

Cannot get the lock for file system. Other command or daemon is operating with lock.

Response

Please wait to complete operating the command or daemon of a GFS Shard file system in all nodes, and re-execute **mkfs_sfdfs(1M)** again.

E.4.3.5 mkfs_sfdfs: ERROR: Cannot find {any|free} port of sfdfs in *file*

Explanation

A failure occurred because all communication port numbers have been used.

Response

Add a new communication port number for sfdfs to **/etc/services**.

E.4.3.6 mkfs_sfcfs: ERROR: can't find *hostname* as hostname

Explanation

Command argument *hostname* is wrong, or a file system recovery manager is not active on *hostname*.

Response

Check command argument *hostname*. If a file system recovery manager is not active on *hostname*, execute **sfcfrmstart(1M)** command.

E.4.3.7 mkfs_sfcfs: ERROR: Cannot look up *node*

Explanation

The specified valid node (*node*) could not be found in the domain in which **mkfs_sfcfs(1M)** was executed.

Response

Specify a correct host name for *node*, and re-execute the command.

E.4.3.8 mkfs_sfcfs: ERROR: Cannot look up *raw_device_file* in *node*

Explanation

The specified raw device file *raw_device_file* could not be found.

Response

Specify the name of a special file that can be shared by the specified hosts, and re-execute the command.

E.4.3.9 mkfs_sfcfs: ERROR: Cannot open *file*: *errmsg*

Explanation

File *file* could not be opened.

Response

Take action according to detailed information indicated by *errmsg*. For details, see the description of **open(2)**.

E.4.3.10 mkfs_sfcfs: ERROR: Data area is too small, can't mkfs

Explanation

The file data area is not sufficient in size, and therefore, a file system cannot be created.

Response

Review parameters so that a sufficiently large file data area can be reserved.

E.4.3.11 mkfs_sfcfs: ERROR: File system is corrupted, run fsck manually

Explanation

The file system cannot be created because it is not terminated normally in the middle of partition addition.

Response

Repair the file system using **fsck_sfdfs(1M)**. Complete partition addition using **sfcadd(1M)**, then re-execute **mkfs_sfdfs(1M)**.

E.4.3.12 mkfs_sfdfs: ERROR: Illegal host in the specified file system.

Explanation

The host that executed **mkfs_sfdfs(1M)** does not belong to the host group of the specified file system.

Response

Execute **mkfs_sfdfs(1M)** using the host belonging to the host group name of the specified file system.

E.4.3.13 mkfs_sfdfs: ERROR: Illegal option: *option*

Explanation

An illegal option was specified.

Response

Specify a correct option and argument.

E.4.3.14 mkfs_sfdfs: ERROR: Invalid magic word *0xmagic*

Explanation

Part of the management data of the specified GFS Shared File System was destroyed.

Response

Restore the management data using **fsck_sfdfs(1M)**, or delete it correctly using **sfcadm(1M)**.

E.4.3.15 mkfs_sfdfs: ERROR: LOG partition small, need over *size* mega bytes

Explanation

A file system cannot be created because the update log partition is too small.

Response

Check the parameter, and allocate *size* megabytes or more for the update log partition.

E.4.3.16 mkfs_sfdfs: ERROR: Master partition is small, log area space not enough

Explanation

The update log area cannot be created because the representative partition is too small.

Response

Check the specified parameter, allocate enough space for the representative partition, and re-execute the command.

E.4.3.17 mkfs_sfdfs: ERROR: Master partition is small, meta area not enough

Explanation

The meta-data area cannot be created because the representative partition is too small.

Response

Check the specified parameter, allocate enough space for the representative partition, and re-execute

the command.

E.4.3.18 mkfs_sfcfs: ERROR: Metasz small, can't mkfs

Explanation

Creation of a file system stops because the meta-data area is too small for managing the specified data area and file.

Response

Specify a larger value for the **metasz** parameter to increase the size of the meta-data area. Alternatively, check the data partition and **nbpi** and other parameters, and reduce the size of the meta-data area.

E.4.3.19 mkfs_sfcfs: ERROR: No response from *node*

Explanation

Communication for the host specified for *node* has not been enabled.

Response

When communication for the host has been enabled, re-execute the command.

E.4.3.20 mkfs_sfcfs: ERROR: Not super user

Explanation

The user executing this command is not a super user.

Response

Re-execute the command as a super user.

E.4.3.21 mkfs_sfcfs: ERROR: Number of file system is over *count*

Explanation

The new GFS Shared File System could not be added because the maximum number of file systems in the same domain indicated by *count* was exceeded.

Response

If possible, delete an unnecessary GFS Shared File System, and re-execute the command. For details of how to delete a file system, see the description of [sfcadm\(1M\)](#) and [sfcinfo\(1M\)](#).

Extend the capacity of an existing GFS Shared File System to correct the error. For details, see the description of [sfcadd\(1M\)](#).

E.4.3.22 mkfs_sfcfs: ERROR: Number of hostname is over *num*

Explanation

The number of specified sharing hosts exceeds the maximum *num* for GFS Shared File System.

Response

Specify no more than four hosts that will share the file system.

E.4.3.23 mkfs_sfcfs: ERROR: Number of mdsnode is over *num*

Explanation

The MDS operational information exceeds the limit of *num* that can be specified.

Response

Limit the MDS operational information to num.

E.4.3.24 mkfs_sfcfs: ERROR: Number of partition is over *count* in maxvol

Explanation

The number of specified partitions exceeded the upper limit *count* specified for the **maxvol** option.

Response

Specify a larger value for the **maxvol** option, or decrease the number of partitions configuring the file system by combining partitions into a larger partition.

E.4.3.25 mkfs_sfcfs: ERROR: Same device is specified more than once

Explanation

The specified partition has already been specified.

Response

Change the specification so that a partition is specified only as a representative, update log, or data partition.

E.4.3.26 mkfs_sfcfs: ERROR: Same hostname *string* is specified more than once

Explanation

The specified host name indicated by *string* has already been specified.

Response

Specify a unique host name for the **-o node** option.

E.4.3.27 mkfs_sfcfs: ERROR: failed to check service is stopped : *errmsg*

Explanation

Mount state acquisition of a special file went wrong.

Response

Please perform management according to the detailed information *errmsg*.

E.4.3.28 mkfs_sfcfs: ERROR: fstat(2) failed *special*: *errmsg*

Explanation

The status of a special file could not be obtained.

Response

Take action according to detailed information indicated by *reason*. For details, see the description of **fstat(2)**.

E.4.3.29 mkfs_sfcfs: ERROR: ioctl(2) {DKIOCINFO|DKIOCGVTOC} error: *errmsg*

Explanation

Device information could not be obtained.

Response

Check the system environment to see whether the specified special file is ready to use, and re-execute the command.

E.4.3.30 mkfs_sfcfs: ERROR: lseek(2) failed raw_device_file: errmsg

Explanation

`lseek(2)` could not be executed for `raw_device_file`.

Response

Take action according to detailed information indicated by `reason`. For details, see the description of `lseek(2)`.

E.4.3.31 mkfs_sfcfs: ERROR: malloc(3C) failed: errmsg

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.4.3.32 mkfs_sfcfs: ERROR: raw_device_file is already used

Explanation

The specified character special device is used with the other GFS Shared File System.

Response

Re-execute the command after taking one of the following actions:

- Use an unused character special device.
- Delete the file system if it is not being used.
- Specify **-o force** to re-create a file system with exactly the same configuration.

E.4.3.33 mkfs_sfcfs: ERROR: raw_device is already used as transit device.

Explanation

Since it is in other file systems in the middle of additional, specified partition `raw_device` cannot be used.

Response

By `sfcinfo(1M)`, check that the partition to specify is not in the middle of additional, and re-perform `sfcadd(1M)`.

E.4.3.34 mkfs_sfcfs: ERROR: raw_device_file is mounted, can't mkfs

Explanation

The specified partition indicated by `raw_device_file` cannot be changed because the file system is mounted.

Response

Unmount the file system, delete the file system using `sfcadm(1M)` and , and re-execute the command.

E.4.3.35 **mkfs_sfcfs: ERROR: *raw_device_file* is not a sfcfs file system**

Explanation

The specified character special device indicated by *raw_device_file* is not the GFS Shared File System.

Response

Specify the character special file of the GFS Shared File System.

Execute the command without specifying the **-o force** option, or correct the */etc/vfstab* setting.

E.4.3.36 **mkfs_sfcfs: ERROR: *raw_device_file* is not special device, can't mkfs**

Explanation

The specified *raw_device_file* is not a character special file.

Response

Specify a character special file.

E.4.3.37 **mkfs_sfcfs: ERROR: read(2) error at *raw_device_file*: *errmsg***

Explanation

read(2) for *raw_device_file* failed.

Response

Take action according to detailed information indicated by *errmsg*. For details, see the description of **read(2)**.

E.4.3.38 **mkfs_sfcfs: ERROR: write(2) error at *special*: *errmsg***

Explanation

I/O for *special* failed.

Response

Take action according to detailed information indicated by *errmsg*. For details, see the description of **write(2)**.

E.4.3.39 **reset to time: bogus optimization preference reset to *opt***

Explanation

The value *opt* specified for the **opt** option is illegal. The file system is created using the default value indicated by **time**.

Response

If there is a problem when **opt** option is **time**, delete the file system by **sfcadm(1M)** command and set **opt = s** and then re-execute the command.

E.4.3.40 ***value*: bogus logsz reset to system set value**

Explanation

The value *value* specified for the **logsz** option is outside the valid range. The file system is created using a calculated size.

Response

If there is a problem, specify a correct parameter value, and re-execute the command.

E.4.3.41 *value*: bogus maxdsz reset

Explanation

The value *value* specified for the **maxdsz** option is outside the valid range. The file system is created using a calculated size.

Response

If there is a problem, specify a correct parameter value, and re-execute the command.

E.4.3.42 *value*: bogus maxnode reset to 16

Explanation

The value *value* specified for the **maxnode** option is outside the valid range. The file system is created using the default value (16).

Response

If there is a problem, specify a correct parameter value, and re-execute the command.

E.4.3.43 *value*: bogus maxvol reset to 16

Explanation

The value *value* specified for the **maxvol** option is outside the valid range. The file system is created using the default value (16).

Response

If there is a problem, specify a correct parameter value, and re-execute the command.

E.4.3.44 *value*: bogus metasz reset

Explanation

The value *value* specified for the **metasz** option is outside the valid range. The file system is created using a calculated size.

Response

If there is a problem, specify a correct parameter value, and re-execute the command.

E.4.3.45 *value*: bogus mminfree reset to 10%

Explanation

The value *value* specified for the **mfree** option is outside the valid range. The file system is created using the default value (10%).

Response

If there is a problem, specify a correct parameter value, and re-execute the command.

E.4.3.46 *value*: bogus nblkpext reset to 1

Explanation

The value *value* specified for the **nblkpext** option is outside the valid range. The file system is created using the default value (1).

Response

If there is a problem, specify a correct parameter value, and re-execute the command.

E.4.3.47 *value*: bogus nbpi reset to 8192

Explanation

The value *value* specified for the **nbpi** option is outside the valid range. The file system is created using the default value (8192).

Response

If there is a problem, specify a correct parameter value, and re-execute the command.

E.4.4 mount_sfdfs command

E.4.4.1 ERROR: mount: 0001: Permission denied

Explanation

Execution was attempted with other than super user permission.

Response

Retry execution with super user permission.

E.4.4.2 ERROR: mount: 0002: Cannot open vfstab file

Explanation

/etc/vfstab could not be opened.

Response

Make sure that */etc/vfstab* exists. If not, restore it from a backup, or see [Section 20.2.3, "Setting */etc/vfstab*"](#) and redefine it.

E.4.4.3 ERROR: mount: 0003: Both *special* and *mount_point* need to be specified in vfstab

Explanation

Special file *special*, or mount point *mount_point* is not described in */etc/vfstab*.

Response

Check the status of */etc/vfstab*.

E.4.4.4 ERROR: mount: 0004: {*special* | *mount_point*}: Cannot connect to cluster foundation

Explanation

Connection to the cluster control facility failed.

Response

Confirm that the cluster control facility has started.

E.4.4.5 ERROR: mount: 0005: {*special* | *mount_point*}: Cannot allocate memory

Explanation

Memory acquisition failed. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.4.4.6 ERROR: mount: 0007: {*special* | *mount_point*}: Cannot get lock: config

Explanation

The lock to change the file system (*mount_point*) configuration information cannot be obtained. In an

optional process within the cluster configuration, the lock to change the file system configuration information is being obtained.

Response

Execution must be performed after waiting for the lock to change the domain configuration information to be released.

E.4.4.7 ERROR: mount: 0008: {*special* | *mount_point*}: Read error on management partition to get fsid

Explanation

Reading of the management partition to obtain the file system (*mount_point*) id has failed.

Response

After confirming the state of `sfcfrmd` daemon by using the like of the `ps(1)` command, contact local Customer Support.

E.4.4.8 ERROR: mount: 0009: {*special* | *mount_point*}: Cannot get lock: *fsid*

Explanation

Exclusive lock for `fsid` within the domain cannot be acquired. The exclusive lock for `fsid` has been acquired by another process running on a node in the cluster.

Response

Wait for the exclusive lock for `fsid` to be released, and then retry.

E.4.4.9 ERROR: mount: 0010: {*special* | *mount_point*}: Read error on management partition to get node information

Explanation

Reading of the management partition to obtain the cluster configuration node information has failed.

Response

After confirming the state of `sfcfrmd` daemon by using the likes of the `ps(1)` command, contact local Customer Support.

E.4.4.10 ERROR: mount: 0011: {*special* | *mount_point*}: Node information for this node was not found

Explanation

The node is not registered as a member to share GFS Shared File System.

Response

Show information of the node by `sfcinfo(1M)` command with `-an` option. If the host name of the node does not appear, add the host name by `sfcnode(1M)` command. If the host name of the node is wrong, change the host name by `sfcadd(1M)` command.

E.4.4.11 ERROR: mount: 0012: Cannot mount *special*

Explanation

Mount processing for *special* failed.

Response

Check the status of the specified special file and mount point, and then retry mount processing.

E.4.4.12 ERROR: mount: 0013: {*special* | *mount_point*}: sfcfrmd may not exist

Explanation

A communications channel with sfcfrmd daemon could not be established.

Response

After confirming the state of sfcfrmd daemon by using the likes of the **ps(1)** command, contact local Customer Support.

E.4.4.13 ERROR: mount: 0018: *special* is already mounted, *mount_point* is busy, or the allowable number or mount points has been exceeded

Explanation

Special file *special* may have already been mounted, or the mount point *mount_point* may be in use.

Response

Please check the process that is used *mountpoint* by **fuser(1M)**. Or check the file system that is used *mount_point* by **mount(1M)**.

E.4.4.14 ERROR: mount: 0019: {*special* | *mount_point*}: Mount demand processing to sfcfrmd went wrong

Explanation

Mount processing request to sfcfrmd daemon has failed.

Response

After confirming the state of sfcfrmd daemon by using the likes of the **ps(1)** command, contact local Customer Support.

E.4.4.15 ERROR: mount: 0020: {*special* | *mount_point*}: Time out occurred in mount demand processing to sfcfrmd

Explanation

Timeout occurred for a mount processing request to sfcfrmd daemon.

Response

After confirming the state of sfcfrmd daemon by using the likes of the **ps(1)** command, contact local Customer Support.

E.4.4.16 ERROR: mount: 0025: The description form of vfstab is wrong

Explanation

The description format of */etc/vfstab* is incorrect.

Response

Confirm the contents of */etc/vfstab*.

E.4.4.17 ERROR: mount: 0026: {*special* | *mount_point*}: filesystem may not be created

Explanation

The specified file system was not registered in the management partition.

Response

After confirming that the file system has already been created, re-execute the **mount_sfcs(1M)** command.

E.4.4.18 ERROR: mount: 0029: {*special* | *mount_point*}: self-node is already processing mount or umount

Explanation

In the self-node, mount or umount processing has already been performed.

Response

Please set time a little and execute a command again.

E.4.4.19 ERROR: mount: 0030: {*special* | *mount_point*}: mount_point need to be representative partition

Explanation

The special block device specified as a mount point was not a representation partition.

Response

Please specify a representation partition name and execute a command again.

E.4.4.20 WARNING: mount: 0027: mount_option *mntopts* is ignored, used *vfs_mntopts* in vfstab

Explanation

Although mount option *mntopts* has been specified as the **mount_sfcs(1M)** command parameter, mount processing was performed with mount option *vfs_mntopts*, described by */etc/vfstab*.

Response

To specify the mount option and perform mount processing, describe the mount option for the mount point in question in */etc/vfstab*.

E.4.4.21 WARNING: mount: 0028: mount_option *mntopts* is ignored

Explanation

Although mount option *mntopts* has been adopted as the **mount_sfcfs(1M)** command parameter, it was ignored and mount processing was performed with no parameters.

Response

To specify the mount option and perform mount processing, describe the mount option for the mount point in question in **/etc/vfstab**.

E.4.5 umount_sfdfs command

E.4.5.1 ERROR: umount: 0001: Permission denied

Explanation

Execution was attempted with other than super user permission.

Response

Retry execution with super user permission.

E.4.5.2 ERROR: umount: 0002: Cannot open mnttab file

Explanation

The `/etc/mnttab` file could not be opened and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.4.5.3 ERROR: umount: 0003: Both *special* and *mount_point* need to be specified in mnttab

Explanation

It needs to be described by `/etc/mnttab` when both the special file *special* and *mount_point* are specified as an operand.

Response

Check the arguments.

E.4.5.4 ERROR: umount: 0004: {*special* | *mount_point*} not in mnttab

Explanation

Although a special file or {*special* | *mount_point*} has been specified for the mount point, neither is described in `/etc/mnttab`.

Response

Check the arguments.

E.4.5.5 ERROR: umount: 0005: {*special* | *mount_point*}: Cannot allocate memory

Explanation

Memory acquisition failed. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.4.5.6 ERROR: umount: 0006: {*special* | *mount_point*}: Cannot connect to cluster foundation

Explanation

Connection to the cluster control facility failed.

Response

Confirm that the cluster control facility has already been started.

E.4.5.7 ERROR: umount: 0007: {*special* | *mount_point*}: Read error on management partition to get fsid

Explanation

Reading of the management partition to obtain the file system id failed.

Response

After confirming the state of sfcfrmd daemon by using the likes of the **ps(1)** command, contact local Customer Support.

E.4.5.8 ERROR: umount: 0008: {*special* | *mount_point*}: Cannot get lock: *fsid*

Explanation

Exclusive lock for the file system id within the domain cannot be obtained. Exclusive lock for the file system id (*fsid*) within the domain is being obtained with an optional process within the cluster configuration.

Response

Try to re-execute after the release of exclusive lock of *fsid* within the domain.

E.4.5.9 ERROR: umount: 0009: {*special* | *mount_point*}: Read error on management partition to get node information

Explanation

Reading of the management partition to obtain the cluster configuration node information has failed.

Response

After confirming the state of sfcfrmd daemon by using the likes of the **ps(1)** command, contact local Customer Support.

E.4.5.10 ERROR: umount: 0010: {*special* | *mount_point*}: Node information for this node was not found

Explanation

Although the management partition containing the cluster node information was read successfully, no self-node information was found to exist.

Response

Check the cluster configuration, and then contact local Customer Support.

E.4.5.11 ERROR: umount: 0011: {*special* | *mount_point*}: sfcfrmd may not exist

Explanation

A communication path to sfcfrmd daemon could not be established.

Response

After confirming the state of sfcfrmd daemon by using the likes of the **ps(1)** command, contact local Customer Support.

E.4.5.12 ERROR: umount: 0012: *special* not mounted

Explanation

Special had already been unmounted.

Response

None.

E.4.5.13 ERROR: umount: 0013: *special*: mount is in progress

Explanation

Mount processing had already been performed to *special*.

Response

Either wait a short while and retry the unmount processing, or perform the unmount processing in forced mode.

E.4.5.14 ERROR: umount: 0014: *special*: fail over is in progress

Explanation

Unmount processing was attempted as failover processing was being performed for *special*.

Response

Either wait a short while and retry the unmount processing, or perform the unmount processing in forced mode.

E.4.5.15 ERROR: umount: 0015: *special*: umount is in progress

Explanation

Unmount processing had already been performed to *special*.

Response

None.

E.4.5.16 ERROR: umount: 0016: {*special* | *mount_point*} busy

Explanation

Unmount processing for {*special* | *mount_point*} failed.

Response

After confirming the state of the special file *special* or mount point *mount_point*, contact local Customer

Support.

E.4.5.17 ERROR: umount: 0017: *device*: Umount demand processing to sfcfrmd went wrong

Explanation

The unmount processing request for sfcfrmd daemon failed.

Response

After confirming the state of sfcfrmd daemon by using the likes of the **ps(1)** command, contact local Customer Support.

E.4.5.18 ERROR: umount: 0018: *device*: Time out occurred in umount demand processing to sfcfrmd

Explanation

A timeout occurred for the sfcfrmd daemon unmount processing request.

Response

After confirming the state of sfcfrmd daemon by using the likes of the **ps(1)** command, contact local Customer Support.

E.4.5.19 ERROR: umount: 0022: *special* is not block device

Explanation

As the argument of the **umount_sfcfs(1M)** command, a block device was not specified.

Response

Specify a block device.

E.4.5.20 ERROR: umount: 0027: {*special* | *mount_point*}: self-node is already processing mount or umount

Explanation

In the self-node, mount or umount processing has already been performed.

Response

Please set time a little and execute a command again.

E.4.5.21 ERROR: umount: 0028: {*special* | *mount_point*}: *mount_point* need to be representative partition

Explanation

The special block device specified as a mount point was not a representation partition.

Response

Please specify a representation partition name and execute a command again.

E.4.5.22 WARNING: umount: 0026: {*special / mount_point*}: Cannot get lock: fsid, but processing is continued compulsorily

Explanation

Although exclusive lock could not be acquired for fsid within the domain, processing was performed forcibly.

Response

None.

E.5 Management Command messages Specific to GFS Shared File System

This section explains the GFS Shared File System management command messages.

E.5.1 sfcadd command

E.5.1.1 sfcadd:/etc/mnttab cannot open

Explanation

The `/etc/mnttab` file could not be opened and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.1.2 sfcadd: Add another partition is in progress

Explanation

The addition of another partition is in progress.

Response

Re-execute `sfcadd(1M)` in 10 minutes.

E.5.1.3 sfcadd: Add transit device, unknown error(*ecode*)

Explanation

An invalid error was detected when an attempt was made to add an entry to the management partition, assuming that the specified partition to be in the process of addition.

Response

Contact local Customer Support.

E.5.1.4 sfcadd: An AC of shared node detected disk I/O error

Explanation

An AC of a node sharing the file system detected a disk I/O error.

Response

Check the partition specified by an argument for normal access from all shared nodes. If a problem does not exist, re-execute `sfcadd(1M)`.

E.5.1.5 sfcadd: An AC of shared node was shut down

Explanation

The AC of either one of the nodes that share the file system was shut down.

Response

Detailed information is reported by error message `sfcmsg` daemon or AC. Apply the countermeasures described in that message. After taking appropriate action, perform mounting, and then re-execute

sfcadd(1M).

E.5.1.6 sfcadd: Another file system use specified device.

Explanation

The specified partition is being used by another file system.

Response

Execute **sfcinfo(1M)** to confirm a partition not used by another system, then specify the partition and execute **sfcadd(1M)**.

E.5.1.7 sfcadd: Not specified all transit partition names with D option

Explanation

All the partitions that remain in the management partition for which addition is in progress are not specified as added devices.

Response

According to the contents of the message that is output after this message, specify *devicename* of all devices that are being added, and then re-execute **sfcadd(1M)**.

E.5.1.8 sfcadd: argument error

Explanation

The specified command format is wrong.

Response

Check the command specification format, and re-execute the command.

E.5.1.9 sfcadd: Cannot add partition, because special is mounted as readonly

Explanation

As the specified file system is mounted in read only mode, **sfcadd(1M)** cannot be executed.

Response

Execute **sfcadd(1M)** after unmounting, and then perform mounting in other than read only mode.

E.5.1.10 sfcadd: cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.1.11 sfcadd: Cannot acquire lock: *type(num)*

Explanation

Cannot get the lock for file system. Other command or daemon is operating with lock.

Response

Please wait to complete operating the command or daemon of a GFS Shard file system in all nodes, and re-execute **sfcadd(1M)** again.

E.5.1.12 sfcadd: Cannot get host information

Explanation

It failed in the acquisition of host information because the host name and the device name specified by the argument were wrong or the **sfcfsrm** demon did not operate.

Response

Specify correctly again and execute **sfcadd(1M)** again when the host name the device name specified by the argument is wrong.

Confirm whether the file system control facility is operating normally while all hosts that share the target file system are activated, and then re-execute **sfcadd(1M)**.

E.5.1.13 sfcadd: Cannot connect to FsRM

Explanation

The file system control facility is not operating.

Response

Confirm whether the file system control facility is running.

E.5.1.14 sfcadd: '-D' and '-L' are mutually exclusive

Explanation

The **-D** and **-L** options were specified together.

Response

Remove **-L** option.

E.5.1.15 sfcadd: Fail to read disk

Explanation

MDS failed to read a disk at a node.

Response

For details, see a MDS message on a console.

E.5.1.16 sfcadd: file system (*partition-name*) is corrupted. To Fix: **fsck -F sfcfs *partition-name***

Explanation

sfcadd(1M) cannot be executed because the file system in the specified partition has not terminated normally.

Response

Repair the file system using **fsck_sfcfs(1M)**, then re-execute **sfcadd(1M)**.

E.5.1.17 sfcadd: FSID(*fsid*) is invalid.

Explanation

The file system ID number (*fsid*) of the specified file system is not registered in file system configuration information.

Response

Execute **sfcinfo(1M)** to check a correct file system ID number of the file system.

E.5.1.18 sfcadd: Illegal host in the specified file system.

Explanation

The host executing **sfcadd(1M)** does not belong to the host group of the specified file system.

Response

Execute **sfcadd(1M)** on the host sharing the object file system.

E.5.1.19 sfcadd: Internal error(*ecode*) strings

Explanation

An internal error occurred.

Response

Contact local Customer Support.

E.5.1.20 sfcadd: invalid magic word *0xmagic*

Explanation

The partition configuration data of the specified partition is invalid.

Response

Check whether the specified partition belongs to an sfcs file system.

E.5.1.21 sfcadd: ioctl({DKIOCINFO | DKIOCGVTOC}) error

Explanation

Device information could not be obtained.

Response

Check the system environment to see whether device information is correct.

E.5.1.22 sfcadd: I/O error

Explanation

Special file I/O processing failed.

Response

This disk cannot be used, use another disk.

E.5.1.23 sfcadd: major number get error

Explanation

The major number could not be obtained.

Response

Check whether the system environment is correct.

E.5.1.24 sfcadd: No response.

Explanation

A timeout is detected during communication with other nodes during the addition.

Response

Re-execute **sfcadd(1M)**.

E.5.1.25 sfcadd: Internal error occurred

Explanation

Connection with the file system control facility is disabled.

Response

Confirm whether the file system control facility is running normally.

E.5.1.26 sfcadd: not super user

Explanation

The command cannot be executed without super user authority.

Response

Execute the command with super user authority.

E.5.1.27 sfcadd: now locked, can't sfcadd

Explanation

sfcadd(1M) cannot be executed because another command is being executed.

Response

Upon termination of another command, re-execute the command.

E.5.1.28 sfcadd: number of partitions is over the maximum

Explanation

The number of specified partitions exceeded the upper limit.

Response

The maximum number of partitions allowed in a file system is the value specified for the **maxvol** parameter in **mkfs_sfcfs(1M)**. Check whether the number of specified partitions is correct.

E.5.1.29 sfcadd: read error (*block_no*)

E.5.1.30 sfcadd: read length fewer than expected.

Explanation

Special file I/O processing failed.

Response

Check whether the specified special file is correct.

E.5.1.31 sfcadd: same device is specified more than once

Explanation

The same partition already exists in the specified partition.

Response

Check whether the specified partition name is correct.

E.5.1.32 sfcadd: seek error (*block_no*)

Explanation

Special file I/O processing failed.

Response

Check whether the specified special file is correct.

E.5.1.33 sfcadd: file system configuration table sync error

Explanation

Distribution to another node that shares the file system failed.

Response

Confirm whether FsRM is running normally, and then re-execute the command.

E.5.1.34 sfcadd: sharing node is not consistent

Explanation

Sharing-node information is inconsistent.

Response

Check whether sharing-node information is consistent.

E.5.1.35 sfcadd: *special* is already defined in file system configuration table

Explanation

The specified partition name (*special*) has already been registered in the file system configuration information.

Response

Confirm the information within the management partition. Delete file system information that includes duplicated partitions and then re-execute.

E.5.1.36 sfcadd: *special* cannot open

Explanation

A special file (*special*) could not be opened.

Response

Check whether the specified special file is correct.

E.5.1.37 sfcadd: *special* fstat64 error

Explanation

The status of a special file (*special*) could not be obtained.

Response

Check whether the specified special file is correct.

E.5.1.38 sfcadd: *special* already used as sfcfs file system

Explanation

The partition to be added (*special*) is already being used in the file system.

Response

Check whether the specified partition is correct.

E.5.1.39 sfcadd: *special* is illegal device

Explanation

An illegal partition (*special*) that is not part of the file system configuration was specified.

Response

Specify all partitions configuring the file system.

E.5.1.40 sfcadd: *special* is mounted from other node

Explanation

The specified partition (*special*) is in a mount state only on other nodes, and cannot perform **sfcadd(1M)**.

Response

Re-execute the command, after the file system unmounts in all nodes or in the node that's mounted it.

E.5.1.41 sfcadd: *special* is mounted, can't sfcadd

E.5.1.42 sfcadd: *special* is mounted from own other node, can't sfcadd

E.5.1.43 sfcadd: *special* is other sfcfs file system mounted, can't sfcadd

Explanation

sfcadd(1M) could not be executed because the specified partition (*special*) has already been mounted.

Response

Unmount the file system, and re-execute the command.

E.5.1.44 sfcadd: *special* is not character special device

Explanation

The specified file (*special*) is not a character special file.

Response

Specify a character special file.

E.5.1.45 sfcadd: *special* is not found in file system configuration table

Explanation

The specified partition (*special*) is not available in the management partition.

Response

Check whether the specified partition is correct.

E.5.1.46 sfcadd: *special* is not master partition

Explanation

The specified partition name (*special*) does not indicate the representative partition.

Response

Check whether the specified partition is the representative partition.

E.5.1.47 sfcadd: *special* is not sfcfs file system

Explanation

The specified partition (*special*) does not belong to an sfcfs file system.

Response

Check whether the specified partition belongs to an sfcfs file system.

E.5.1.48 sfcadd: *special* is not shared device

Explanation

The specified partition (*special*) is not shared by nodes.

Response

Check whether the system environment is correct.

E.5.1.49 sfcadd: system error(*function_name*)

Explanation

A system error occurred.

Response

Check the system environment.

E.5.1.50 sfcadd: too many data partition

Explanation

Too many special files were specified.

Response

Check whether the number of specified partitions is correct, and re-execute the command.

E.5.1.51 sfcadd: Transit device is exist.

Explanation

Failed to delete a transit partition from the file system configuration data.

Response

Execute **sfcinfo(1M)** to check the file system for remaining transit devices. If devices exist, re-execute **sfcadd(1M)**.

E.5.1.52 sfcadd: unknown error(*ecode*)

Explanation

An illegal error was detected.

Response

Contact local Customer Support.

E.5.1.53 sfcadd: WARNING: *special* is already added

Explanation

The partition (*special*) has already been added.

Response

A response is not required

E.5.1.54 sfcadd: write error (*block_no*)

E.5.1.55 sfcadd: write length fewer than expected.

Explanation

Special file I/O processing failed.

Response

Check whether the specified special file is correct.

E.5.2 sfcadm and sfcnode commands

E.5.2.1 {sfcadm|sfcnode}:all node entry delete, can't sfcnode

Explanation

The command cannot be executed because information about all nodes sharing the file system will be deleted.

Response

Reduce the specified host information, and re-execute the command.

E.5.2.2 {sfcadm|sfcnode}:too many host count

Explanation

Too many host names were specified.

Response

Check whether the number of specified host names is correct, and re-execute the command.

E.5.2.3 {sfcadm|sfcnode}:too many partition count

Explanation

Too many special files were specified.

Response

Check whether the number of specified partitions is correct, and re-execute the command.

E.5.2.4 {sfcadm|sfcnode}:argument error

Explanation

The specified command format is wrong.

Response

Check the command specification format, and re-execute the command.

E.5.2.5 {sfcadm|sfcnode}:cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.2.6 {sfcadm|sfcnode}:cannot get host information

Explanation

The acquisition of host information failed.

Response

Confirm that the file system control facility is running normally and that the target host has not failed.

E.5.2.7 {sfcadm|sfcnode}:Cannot connect to FsRM

Explanation

The file system control facility is not running.

Response

Confirm that file system control facility is running.

E.5.2.8 {sfcadm|sfcnode}:*filename* cannot open

Explanation

Relevant file (*filename*) does not exist.

Response

Confirm if *filename* is available.

Note that, if the management cannot be found, restore it from a backup or create a new one.

E.5.2.9 {sfcadm|sfcnode}:FSID(*fsid_num*) is invalid

Explanation

File system identification numbers (*fsid_num*) are not registered in the management partition.

Response

Confirm whether the relevant FSID number is registered by using **sfcinfo(1M)**.

E.5.2.10 {sfcadm|sfcnode}:hostid(*hostid*) not found in volume information

Explanation

The specified host ID (*hostid*) was not found in the partition configuration data.

Response

Check whether the specified host ID exists in the partition information.

E.5.2.11 {sfcadm|sfcnode}:invalid hostname

Explanation

An invalid host name was specified.

Response

Check the specified host name.

E.5.2.12 {sfcadm|sfcnode}:hostname(*hostname*) is not found

Explanation

The specified host name (*hostname*) was not found in the node information.

Response

Check whether the specified host name exists in the node information.

E.5.2.13 {sfcadm|sfcnode}:*hostname* is not shared

Explanation

The specified host (*hostname*) does not share the device.

Response

Check the system environment.

E.5.2.14 {sfcadm|sfcnode}: *hostname* registered as MDS, can't sfcnode -d

Explanation

The specified host (*hostname*) could not delete, because it is registered as the meta-data server.

Response

Change the meta-data server host by **sfcadm -g**.

E.5.2.15 {sfcadm|sfcnode}: invalid devicename

Explanation

An invalid partition name was specified.

Response

Check whether the specified partition name is correct.

E.5.2.16 {sfcadm|sfcnode}: invalid magic word *magic_no*

Explanation

The partition configuration data of the specified partition is invalid.

Response

Check whether the specified partition belongs to an sfcs file system.

E.5.2.17 {sfcadm|sfcnode}: ioctl({DKIOCINFO | DKIOCGVTOC}) error

Explanation

Device information could not be obtained.

Response

Check the system environment to see whether device information is correct.

E.5.2.18 {sfcadm|sfcnode}:I/O error

E.5.2.19 {sfcadm|sfcnode}:read error (*block_no*)

E.5.2.20 {sfcadm|sfcnode}:seek error (*block_no*)

E.5.2.21 {sfcadm|sfcnode}:write error (*block_no*)

Explanation

Special file I/O processing failed.

Response

Check whether the specified special file is correct.

E.5.2.22 {sfcadm|sfcnode}:log partition size too small

Explanation

The specified log size exceeds the size of the entire log partition.

Response

Check the size of the log partition.

E.5.2.23 {sfcadm|sfcnode}:log partition is not separated

Explanation

The size could not be changed because the log partition is not separated.

Response

Check whether the specified partition is correct.

E.5.2.24 {sfcadm|sfcnode}:logsize is illegal size

Explanation

An illegal log size was specified.

Response

Check whether the specified log size is correct.

E.5.2.25 {sfcadm|sfcnode}:major number get error

Explanation

The major number could not be obtained.

Response

Check whether the system environment is correct.

E.5.2.26 {sfcadm|sfcnode}:invalid MDS hostname

Explanation

The specification of MDS contains an error.

Response

Confirm the management partition and specified MDS information.

E.5.2.27 {sfcadm|sfcnode}:node full, can't {sfcadm | sfcnode}

Explanation

The number of nodes sharing the file system exceeded the upper limit.

Response

The maximum number of nodes sharing a file system is the value specified for the **maxnode** parameter in **mkfs_sfcfs(1M)**. If the maximum number will be exceeded, the file system must be reorganized.

E.5.2.28 {sfcadm|sfcnode}: node name length is too long

Explanation

The specified node name is too long.

Response

Check whether the specified node name is correct. To specify multiple node names, delimit each name by a comma.

E.5.2.29 {sfcadm|sfcnode}:file system configuration information is full of entry

Explanation

No more file system information could be added.

Response

Delete unnecessary information using **sfcadm(1M)**, and re-execute the command.

E.5.2.30 {sfcadm|sfcnode}:not enough element of sfcfs partition

Explanation

Not all the partitions belonging to the file system were specified.

Response

Specify all partitions belonging to the file system.

E.5.2.31 {sfcadm|sfcnode}:this node entry not found

Explanation

Local node information was not found in the information for nodes sharing the file system.

Response

Check whether local node information exists.

E.5.2.32 {sfcadm|sfcnode}:not super user

Explanation

The command cannot be executed without super user authority.

Response

Execute the command with super user authority.

E.5.2.33 {sfcadm|sfcnode}:not support command

Explanation

The executed command is not supported.

Response

Check the executed command name.

E.5.2.34 {sfcadm|sfcnode}: ERROR: Cannot acquire lock: *type(num)*

Explanation

Cannot get the lock for file system. Other command or daemon is operating with lock.

Response

Please wait to complete operating the command or daemon of a GFS Shard file system in all nodes, and re-execute sfcadm(1M) or sfcnode(1M) again.

E.5.2.35 {sfcadm|sfcnode}:please specify MDS setting

Explanation

A meta-data server's host name must be specified.

Response

Specify a meta-data server's host name, and re-execute the command.

E.5.2.36 {sfcadm|sfcnode}:please specify PORTNAME setting

Explanation

A service port name must be specified.

Response

Specify a service port name, and re-execute the command.

E.5.2.37 {sfcadm|sfcnode}:same device is specified more than once

Explanation

A partition that is the same as the one specified was found.

Response

Check whether the specified partition name is correct.

E.5.2.38 {sfcadm|sfcnode}:invalid service portname

Explanation

An invalid service port name was specified.

Response

Check the specified service port name.

E.5.2.39 {sfcadm|sfcnode}: file system configuration information sync error

Explanation

Distribution to another node sharing the file system failed.

Response

Check that a file system watch mechanism works normally, and re-execute the command.

E.5.2.40 {sfcadm|sfcnode}:*special* cannot open

Explanation

A special file (*special*) could not be opened.

Response

Check whether the specified special file is correct.

E.5.2.41 {sfcadm|sfcnode}:*special* fstat64 error

Explanation

The status of a special file (*special*) could not be obtained.

Response

Check whether the specified special file is correct.

E.5.2.42 {sfcadm|sfcnode}:*special* is illegal device

Explanation

An invalid partition (*special*) that is not part of the file system configuration was specified.

Response

Specify all partitions configuring the file system.

E.5.2.43 {sfcadm|sfcnode}:*special* is mounted, can't {sfcadm | sfcnode}

Explanation

The command could not be executed because the specified partition (*special*) has already been mounted.

Response

Unmount the file system, and re-execute the command.

E.5.2.44 {sfcadm|sfcnode}:*special* is not character special device

Explanation

The specified file (*special*) is not a character special file.

Response

Specify a character special file.

E.5.2.45 {sfcadm|sfcnode}:*special* is not sfcfs file system configuration table

Explanation

The specified partition (*special*) does not exist in the management partition.

Response

Reexamine whether the specified partition is correct.

E.5.2.46 {sfcadm|sfcnode}:*special* is not shared device

Explanation

The specified partition (*special*) is not shared by nodes.

Response

Check whether the system environment is correct.

E.5.2.47 {sfcadm|sfcnode}:system error(*function_name*)

Explanation

A system error occurred.

Response

Contact local Customer Support.

E.5.2.48 {sfcadm|sfcnode}:unknown error(*ecode*)

Explanation

An invalid error was detected.

Response

Contact local Customer Support.

E.5.2.49 sfcadm: ERROR: number of mdsnode is over *num*

Explanation

The MDS operational information specified exceeds the number of shared nodes.

Response

Specify a value equal to or less than the number of shared nodes or **all**.

E.5.3 sfcddr command

E.5.3.1 sfcddr: buffer size cannot be zero

Explanation

A value of 0 cannot be specified as an argument of the **-b** option.

Response

Specify a value other than 0.

E.5.3.2 sfcddr: *file*: not character special

Explanation

The specified file (*file*) is not a character special file.

Response

Specify a character special file.

E.5.3.3 sfcddr: backup data is not sfcfs

Explanation

The entered data is not for the GFS Shared File System.

Response

Enter correct data.

E.5.3.4 sfcddr: partition size is different

Explanation

The size of the restore destination partition and the size of the backup source are not the same.

Response

Specify a special file having the same partition size as that of the backup source.

E.5.3.5 sfcddr: no device name

Explanation

The driver name could not be obtained.

Response

Illegal data was entered. Check the entered data again.

E.5.3.6 sfcddr: restoring to a different host

Explanation

An attempt was made to restore data to a host that is not registered as a node.

Response

Restore data to a host registered as a node, or specify the **-f** option.

E.5.3.7 sfccdr: restoring to a different device

Explanation

An attempt was made to restore data to a special file that had not been backed up.

Response

Restore data to the special file that was backed up, or specify the **-f** option.

E.5.3.8 sfccdr: sfcfs partition configuration is invalid

Explanation

The partition definition information in the entered data is invalid.

Response

The entered definition information is invalid. Check the entered data again.

E.5.3.9 sfccdr: *special* is mounted on *mount_point*

Explanation

A special file (*special*) was mounted.

Response

Unmount the special file, and execute the command.

E.5.3.10 sfccdr: unexpected end of file

Explanation

The end of a file was reached unexpectedly.

Response

Check whether the specified file is correct.

E.5.3.11 sfccdr: unexpected short write

Explanation

Write operation terminated unexpectedly.

Response

Check whether the specified special file is correct.

E.5.4 sfcdu commnad

E.5.4.1 sfcdu: *errmsg*

Explanation

An area for storing the names of directories and files was not completely gained/regained. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.4.2 sfcdu: No more processes: *errmsg*

Explanation

Child process generation is incomplete. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.4.3 sfcdu: can't process *directory*: *errmsg*

Explanation

An area for storing the directory path (*directory*) was not completely gained/regained. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.4.4 sfcdu: *directory*: *errmsg*

Explanation

Directory path (*directory*) change is incomplete.

Displayed when `/usr/xpg4/bin/sfcdu` or `/usr/bin/sfcdu` is specified with `-r` option.

Response

Follow the procedures given in the detailed information for *errmsg*.

E.5.4.5 sfcdu: {*file* | *directory*}: *errmsg*

Explanation

Incomplete `stat(2)/lstat(2)` for file and in `opendir(3C)`, `chdir(2)` and `getcwd(3C)` to directory path.

Displayed when `/usr/xpg4/bin/sfcdu` or `/usr/bin/sfcdu` is specified with `-r` option.

Response

Follow the procedures shown in detailed information for *errmsg*.

E.5.4.6 **sfcd**: can't keep track of more links: *errmsg*

Explanation

The area for storing link data was not completely regained.
Displayed when `/usr/xpg4/bin/sfcd` or `/usr/bin/sfcd` is specified with `-r` option.

Response

Follow the procedures given in the detailed information for *errmsg*.

E.5.4.7 **sfcd**: Can't reopen in *directory*: *errmsg*

Explanation

Directory path (*directory*) cannot be re-opened.
Displayed when `/usr/xpg4/bin/sfcd` or `/usr/bin/sfcd` is specified with `-r` option.

Response

Follow the procedures given in the detailed information for *errmsg*.

E.5.4.8 **sfcd**: Can't change dir to `'..'` in *directory/..*: *errmsg*

Explanation

Directories cannot be changed to a quote of `'..'` under a directory path.
Displayed when `/usr/xpg4/bin/sfcd` or `/usr/bin/sfcd` is specified with `-r` option.

Response

Follow the procedures shown in the detailed information for *errmsg*.

E.5.5 sfcedquota command

E.5.5.1 sfcedquota: ERROR: ioctl(Q_FSETQLIM) error in *quotas: errmsg*

Explanation

Failed to reflect data to internal quotas file.

Response

A physical I/O error or communication error is possible. Confirm a console message to respond.

E.5.5.2 sfcedquota: ERROR: no such user

Explanation

The user corresponding to the specified user name is not found.

Response

Check the user name and re-execute the command.

E.5.5.3 sfcedquota: ERROR: no sfcfs filesystem with quotas file

Explanation

The GFS Shared File System having a quotas file cannot be found.

Response

Create a quotas file in the root directory of a file system using a quota function, then enable the quota function and re-execute the command. If the quota function is valid, create a quotas file and execute **sfquotadm(1M)** with **-e** option to reflect a limit value.

E.5.5.4 sfcedquota: ERROR: out of memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.5.5 sfcedquota: ERROR: permission denied

Explanation

Super-user authorization is not used for execution.

Response

Use super-user authorization for execution.

E.5.5.6 sfcedquota: ERROR: Too many processes

Explanation

Excessive number of system processes.

Response

The number of system processes may reach the maximum. Check the system status.

E.5.5.7 sfcedquota: ERROR: uid *uid* too large

Explanation

The specified user ID (*uid*) is excessive.

Response

Check the user ID and re-execute the command.

E.5.5.8 sfcedquota: ERROR: uid *uid* too large for quotas

Explanation

The specified user ID (*uid*) is excessive.

Response

Check the user ID and re-execute the command.

E.5.5.9 sfcedquota: WARNING: line in mnttab exceeds *num* characters

Explanation

The number of characters of the entry in **/etc/mnttab** is over maximum *num*.

Action

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, change entry of above file system in **/etc/vfstab** not to exceed maximum *num*.

E.5.5.10 sfcedquota: WARNING: line in mnttab has too few entries

Explanation

There was an illegal entry with too few fields in **/etc/mnttab**.

Action

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in **/etc/vfstab**.

E.5.5.11 sfcedquota: WARNING: line in mnttab has too many entries

Explanation

There are too many entries in **/etc/mnttab**.

Action

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in **/etc/vfstab**.

E.5.5.12 blocks (soft = *soft*, hard = *hard*) inodes (soft = *soft*, hard = *hard*): bad format

Explanation

The format of the specified limit value is incorrect.

Action

Specify the limit value correctly using an integer equal to or greater than 0.

E.5.5.13 blocks (soft = *soft*, hard = *hard*) inodes (soft = *soft*, hard = *hard*): The value is outside the range of representable values.

Explanation

The specified limit value is too large.

Action

Reexecute the command after checking the limit value.

E.5.6 sfcfrmstart command

E.5.6.1 ERROR: sfcfrmstart: 0001: Permission denied

Explanation

Execution was attempted with other than super user permission.

Response

Retry execution with super user permission.

E.5.6.2 ERROR: sfcfrmstart: 0004: Cannot connect to cluster foundation

Explanation

Connection with the cluster control facility failed.

Response

Confirm that the cluster control facility has already been started.

E.5.6.3 ERROR: sfcfrmstart: 0005: Operation of management partition may be performed

Explanation

The management partition may be being used on either the self-node or another node.

Response

Wait for the operation of the management partition to end, and then retry.

E.5.6.4 ERROR: sfcfrmstart: 0006: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.6.5 ERROR: sfcfrmstart: 0007: Read error on sfcfsrm.conf: *errmsg*

Explanation

Setting file sfcfsrm.conf could not be read.

Response

Perform management according to *errmsg*.

E.5.6.6 ERROR: sfcfrmstart: 0008: Read error on management partition

Explanation

Reading of the management partition failed.

Response

If the disk with the management partition is broken, see [Section 16.5, "Backup and restore of the management partition information"](#) and restore the management partition. Check that a disk has not failed, and then contact local Customer Support.

E.5.6.7 ERROR: sfcfrmstart: 0009: Self-node does not belong to host group

Explanation

The self-node does not belong to a host group.

Response

Using the **sfcsetup(1M)** command, add the self-node to a host group.

E.5.6.8 ERROR: sfcfrmstart: 0010: Starting processing of sfcfrmd went wrong

Explanation

Start processing for sfcfrmd daemon failed.

Response

After confirming the state of the self-node and that sfcprmd daemon is available by using the like of the **ps(1)** command, contact local Customer Support.

E.5.6.9 ERROR: sfcfrmstart: 0011: sfcfrmd has already existed

Explanation

The sfcfrmd daemon has already been started.

Response

No particular countermeasure is necessary.

E.5.6.10 ERROR: sfcfrmstart: 0012: sfcsetup may not be performed

Explanation

As no setting is made to the management partition, the management partition cannot be accessed.

Response

Use the **sfcsetup(1M)** command to set up the management partition.

E.5.6.11 ERROR: sfcfrmstart: 0013: sfcprmd may not exist

Explanation

The communication path to sfcprmd daemon cannot be established.

Response

After confirming the state of the self-node and that sfcprmd daemon is available by using the like of the **ps(1)** command, contact local Customer Support.

E.5.6.12 ERROR: sfcfrmstart: 0014: State check processing to sfcprmd went wrong

Explanation

State confirmation processing for sfcprmd daemon using several different daemons failed.

Response

After confirming the state of self-node and that sfcprmd daemon is available by using the like of the **ps(1)** command, contact local Customer Support.

E.5.6.13 ERROR: sfcfrmstart: 0015: Write error on management partition

Explanation

Write processing for the management partition failed.

Response

After confirming that no disk error occurred, contact local Customer Supports.

E.5.6.14 ERROR: sfcfrmstart: 0016: filename: open(2) error: *errmsg*

Explanation

File open processing for editing the sfcfsrm.vers file failed.

Response

Apply countermeasures as described in *errmsg*.

E.5.6.15 ERROR: sfcfrmstart: 0017: filename: write(2) error: *errmsg*

Explanation

File write processing for editing the sfcfsrm.vers file failed.

Response

Apply countermeasures as described in *errmsg*.

E.5.6.16 ERROR: sfcfrmstart: 0018: filename: rename(2) error: *errmsg*

Explanation

File rename processing for editing the sfcfsrm.vers file failed.

Response

Apply countermeasures as described in *errmsg*.

E.5.7 sfcfrmstop command

E.5.7.1 ERROR: sfcfrmstop: 0001: Permission denied

Explanation

Execution was attempted with other than super user permission.

Response

Retry execution with super user permission.

E.5.7.2 ERROR: sfcfrmstop: 0004: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.7.3 ERROR: sfcfrmstop: 0005: sfcprmd may not exist

Explanation

The communication path to the process monitoring daemon sfcprmd daemon cannot be established.

Response

After confirming the state of the self-node and that sfcprmd daemon is available by using the like of the **ps(1)** command, contact local Customer Support.

E.5.7.4 ERROR: sfcfrmstop: 0006: stop processing of sfcfrmd went wrong

Explanation

Stop processing for sfcfrmd daemon failed.

Response

The GFS may be mounted on the self-mode. Perform umount processing, and then re-execute the **sfcfrmstop(1M)** command.

E.5.7.5 ERROR: sfcfrmstop: 0007: Self-node is still mounted

Explanation

The GFS Shared File System is mounted by the self-node.

Response

Please execute a command again after unmounting all the GFS Shared File Systems of a self-node.

E.5.8 sfcgetconf command

E.5.8.1 ERROR: sfcgetconf: 0001: Permission denied

Explanation

Execution was attempted with other than super user permission.

Response

Perform execution with super user permission.

E.5.8.2 ERROR: sfcgetconf: 0002: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.8.3 ERROR: sfcgetconf: 0003: Invalid '-i' parameter: "*iparam*"

Explanation

The parameter specified for the "i" option is invalid.

Response

Re-execute the command with a valid (numerical) parameter in the "i" option.

E.5.8.4 ERROR: sfcgetconf: 0004: Invalid parameter

Explanation

The format of the command parameter is invalid.

Response

Re-execute the command with a valid parameter according to the format.

E.5.8.5 ERROR: sfcgetconf: 0005: *backupfile* is already exist

Explanation

The specified file as a backup (*backupfile*) has already existed.

Response

Re-execute the command with a file that does not exist.

E.5.8.6 ERROR: sfcgetconf: 0006: Cannot open *backupfile*: open(2) returned *errcode=errno*

Explanation

The **open(2)** processing for the backup file (*backupfile*) failed.

Response

Re-execute the command with the correct name of the backup file after checking the location where the backup file will be created.

E.5.8.7 ERROR: sfcgetconf: 0007: Cannot fdopen *backupfile*: fdopen(3C) returned errcode= *errno*

Explanation

The **fdopen(3C)** processing for the backup file (*backupfile*) failed.

Response

Re-execute the command with the correct name of the backup file after checking the location where the backup file will be created.

E.5.8.8 ERROR: sfcgetconf: 0008: Cannot access to cluster foundation: frm_cl_open() returned errcode=*errno*

Explanation

Communication with the cluster control could not be established.

Response

Check whether the cluster control is operating normally.

E.5.8.9 ERROR: sfcgetconf: 0009: Cannot get domain wide lock: config

Explanation

Execution of this command failed as another command is currently being executed.

Response

Re-execute this command after the execution of the other command.

E.5.8.10 ERROR: sfcgetconf: 0010: Cannot open *path_to_sfcfm.conf*: fopen(3C) returned errcode=*errno*

Explanation

The **fopen(3C)** processing for *sfcfm.conf* failed while attempting to acquire a backup of the *sfcfm.conf* file according to the "-a" option.

Response

Check whether *sfcfm.conf* exists at the path location displayed in the message. If the file is missing, then **sfctestup(1M)** processing may not have been performed.

E.5.8.11 ERROR: sfcgetconf: 0011: file system configuration information parse error: frm_tabent() returned errcode=*errno*

Explanation

There is invalid data in *sfctestab* of the management partition.

Response

Use the backup of the management partition that was acquired previously to restore the management partition.

E.5.8.12 NOTICE: sfcgetconf: 0012: fsid=*fsid* is not found

Explanation

The file system configuration information that corresponds to *fsid*, which was specified as the target of the search with the '-i' option, could not be found.

Backup processing of the file system configuration information, that corresponds to the specified *fsid*, was ignored by the command.

Response

By eliminating the target of the *fsid* specification from the command argument specification, message output can be disabled.

E.5.8.13 NOTICE: sfcgetconf: 0013: dev=*raw_device_file* is not found

Explanation

The file system configuration information that corresponds to *raw_device_file*, which was specified as the target of the search with the '-d' option, could not be found.

Backup processing of the file system configuration information that corresponds to the specified *raw_device_file* was ignored by the command.

Response

By eliminating the target of the *raw_device_file* specification from the command argument specification, message output can be disabled.

E.5.9 sfchost command

E.5.9.1 sfchost: Cannot acquire lock: *type(num)*

Explanation

Cannot get the lock for file system. Other command or daemon is operating with lock.

Response

Please wait to complete operating the command or daemon of a GFS Shard file system in all nodes, and re-execute **sfchost(1M)** again.

E.5.9.2 sfchost: Cannot connect to sfcfrmd.

Explanation

Cannot connect to sfcfrmd daemon did not operate.

Response

Make sure that sfcfrmd daemon is running by **ps(1M)** command (for example, "# /usr/bin/ps ?e | grep sfcfrmd"). It is possible that sfcfrmd daemon stopped because of **sfcrmstop(1M)** was executed. In this case, execute **sfcrmstart(1M)** in order to re-start sfcfrmd daemon. Otherwise, re-start the node in question.

If sfcfrmd daemon is running when this message is output, collect the diagnostic data with fjsnap and contact local Customer Support.

E.5.9.3 sfchost: can't find *host* as hostname

Explanation

Cannot get the host information of *host*.

Response

When the host name specified by the argument is mistaken, the right host name is specified. Please re-perform **sfchost(1M)** again.

When at least one node of share node of the target file system has stopped, please re-execute **sfchost(1M)** after starting a node.

E.5.9.4 sfchost: can't find secondary connection information for *host*

E.5.9.5 sfchost: can't find primary connection information for *host*

Explanation

host is not in the connection host name list of file system.

Response

host cannot be specified. Please change node information of file system by **-m** option of **sfcdm(1M)** when you want to use specified *host*.

E.5.9.6 sfchost: can't open mnttab: *errmsg*

Explanation

The **/etc/mnttab** file could not be opened and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.9.7 sfchost: can't open mount point *mnt_point*

Explanation

mnt_point could not be opened. There is a possibility that the file system has blockaded.

Response

Please deal with console messages that MDS and AC output.

E.5.9.8 sfchost: cannot find mount point *mnt_point*

Explanation

mnt_point could not find in entry of */etc/vfstab*.

Response

Please specify the mountpoint of the GFS shared file system of the built-in predefined for */etc/vfstab*.

E.5.9.9 sfchost: cannot fstatvfs *mnt_point*: *errmsg*

Explanation

Could not get the information of *mnt_point*. There is a possibility that the file system has blockaded.

Response

Please deal with console message that MDS and AC output.

E.5.9.10 sfchost: cannot open *device*

E.5.9.11 sfchost: cannot open *mnt_point*: *errmsg*

Explanation

Could not open *device* or *mnt_point* specified by arguments.

Response

Please specify the correct block special file or the directory of mountpoint of active GFS Shared File System.

E.5.9.12 sfchost: cannot stat *device*

E.5.9.13 sfchost: cannot stat *mnt_point*

Explanation

Could not get the information of *device* or *mnt_point* specified by arguments.

Response

Please specify the correct block special file or the directory of mountpoint of active GFS Shared File System.

E.5.9.14 sfchost: file system type is not correct, *fstyp*

Explanation

The block special file specified by the argument is not the thing of a GFS Shared File System.

Response

Please specify the correct block special file or the directory of mountpoint of active GFS Shared File System.

E.5.9.15 sfchost: *file* is not directory nor block device

E.5.9.16 sfchost: *file* is not used as sfcfs filesystem

Explanation

file is not directory nor block device of active GFS Shared File System.

Response

Please specify the correct block special file or the directory of mountpoint of active GFS Shared File System.

E.5.9.17 sfchost: *host* and *host2* are same network address

Explanation

host and *host2* are same network address. Since it does not become LAN multiplexing, it considers as an error.

Response

Please specify the hostnames which represents a different network address.

E.5.9.18 sfchost: ioctl(SFCFS_MPATH_GET) failed: *errmsg*

E.5.9.19 sfchost: ioctl(SFCFS_MPATH_SWITCH) failed: *errmsg*

Explanation

Cannot get the connected hostname information from AC.

Response

Please deal with console message that MDS and AC output.

E.5.9.20 sfchost: memory allocation failed

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.9.21 sfchost: *mnt_point* is not a mount point of sfcfs file system

Explanation

The block special file specified by the argument is not the thing of a GFS Shared File System.

Response

Please specify the correct block special file or the directory of mountpoint of active GFS Shared File System.

E.5.9.22 sfchost: multihost must be less than or equal to 4

Explanation

The hostname that exceeded limiting value 4 by the argument was specified. Since the limiting value was exceeded, it considers as an error.

Response

Please adjust connected hostname to four and set it.

E.5.9.23 sfchost: sfcfs filesystem fsid=*device* is not found

Explanation

The block special file specified by the argument is not the thing of a GFS Shared File System.

Response

Please specify the correct block special file or the directory of mountpoint of active GFS Shared File System.

E.5.9.24 sfchost: specified hostname is not for same node *host*, *host2*

Explanation

host and *host2* is not hostname of same node. Since it does not become LAN multiplexing, it considers as an error.

Response

Please specify the hostnames of same node.

E.5.9.25 sfchost: top of multihost can't change by sfchost.

Explanation

The host name in the head of a connection host name list cannot be changed.

Response

In order to change the head host name of a connection host name list, **sfcadm(1M)** with **-m** option is used.

E.5.9.26 sfchost: unknown error occurred

Explanation

Unknown error occurred.

Response

Please wait to complete operating the command or daemon of a GFS Shard file system in all nodes, and re-execute **sfhost(1M)** again.

Still, please deal with console message that MDS and AC output when the same message is output.

E.5.10 sfcinfo command

E.5.10.1 sfcinfo: ERROR: Cannot allocate memory: *errmsg*

Explanation

Memory could not be allocated. Memory of system is insufficient. Processing cannot be continued.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.10.2 sfcinfo: ERROR: Cannot get major number of *drivename*: *errmsg*

Explanation

Failed to get major number of *drivename*. Processing cannot be continued.

Response

Check the system environment of device driver.

E.5.10.3 sfcinfo: ERROR: Cannot open *{/etc/mnttab|/etc/vfstab}*: *errmsg*

Explanation

Failed to open */etc/mnttab* or */etc/vfstab* file because of *errmsg*. Processing cannot be continued.

Response

Make sure that */etc/vfstab* exists. If not, restore it from a backup, or see [Section 20.2.3, "Setting /etc/vfstab"](#) and redefine it. The problem of Solaris OE occurs when */etc/mnttab* is lost in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.10.4 sfcinfo: ERROR: file system configuration information not found

Explanation

Failed to get the file system configuration information in the management partition.

Response

Restore the file system configuration information. For detail, see [chapter 16, "Management Partition."](#)

E.5.10.5 sfcinfo: ERROR: Invalid device number *dnum*

Explanation

User specified a invalid device number (*dnum*).

Response

Correct device number and this command is re-executed.

E.5.10.6 sfcinfo: ERROR: Unknown error(*ecode*)

Explanation

Failed work with unknown error.

Response

Contact local Customer Support.

E.5.10.7 sfcinfo: WARNING: Cannot get convert name of *special*

Explanation

Failed to convert *special* to block special path name. Other file systems are processed continuously.

Response

Check the definition of file system in the */etc/vfstab*.

E.5.10.8 sfcinfo: WARNING: Cannot open { *special* | *mntpoint*}: *errmsg*

Explanation

Failed to open *special* or *mntpoint* because of *errmsg*. Other file systems are processed continuously.

Response

Check the arguments of command or the system environment file.

If the logical volume of the GFS Shared File System is used, information is displayed only when the logical volume of the GFS Shared File System is activated. If the logical volume is not activated, retry after activating it.

If the GFS Shared File System management view is used for operation, retry after activating the logical volume.

E.5.10.9 sfcinfo: WARNING: Invalid magic word *0xmagic* in *special*

Explanation

Configuration data of the specified partition is invalid. Other file systems are processed continuously.

Response

Check the specified partition to determine if it is an **sfcs** file system.

**E.5.10.10 sfcinfo: WARNING: ioctl({DKIOCINFO | DKIOCGVTOC}) failed in *special*:
*errmsg***

Explanation

Failed to obtain the device data of partition *special*. Other file systems are processed continuously.

Response

Check the system environment to determine if partition *special* is correct.

E.5.10.11 sfcinfo: WARNING: Not a block device(*special*)

Explanation

The specified partition must be a block-type special file but it is a character-type special file. Other file systems are processed continuously.

Response

Specify a block-type special file as a partition.

E.5.10.12 sfcinfo: WARNING: Not a directory(*file*)

Explanation

The specified *file* must be a block-type special file or a directory. Other file systems are processed continuously.

Response

Specify a block-type special file or mount point directory.

E.5.10.13 sfcinfo: WARNING: lseek error at *special* to off: *errmsg*

Explanation

Failed to **lseek(2)** at partition *special*. Other file systems are processed continuously.

Response

Check whether the specified partition belongs to an *sfcs* file system.

E.5.10.14 sfcinfo: WARNING: read error in super block of *special*: *errmsg* So try to read alternative block

Explanation

Failed to read normal super block data. Therefore, redundant data is read out.

Response

Delete the object file system if unnecessary. For details, see [Section 20.9, "Deleting"](#) in this manual. Fully scan the object file system using **fsck_sfcs(1M)** to repair. For details, see [Appendix G.2.1, "Action for I/O errors."](#) in this manual.

E.5.10.15 sfcinfo: WARNING: read error in {alternative super block | volume information} of *special*: *errmsg*

Explanation

Failed to read alternative super block data or normal volume data. Other file systems are processed continuously.

Response

Delete the object file system if unnecessary. For details, see [Section 20.9, "Deleting"](#) in this manual. Fully scan the object file system using **fsck_sfcs(1M)** to repair. For details, see [Appendix G.2.1, "Action for I/O errors."](#) in this manual.

E.5.10.16 sfcinfo: WARNING: *special* is not *sfcs* file system

Explanation

The specified partition (*special*) does not belong to an *sfcs* file system. Other file systems are processed continuously.

Response

Check whether the specified partition belongs to an *sfcs* file system.

E.5.11 sfcproxybreak, sfcproxyinfo, sfcproxyjoin, sfcproxyop and sfcproxyrestore command

E.5.11.1 *cmdname*: copy failed

Explanation

The error occurred during copy processing and the copy went wrong.

Response

Respond according to "[22.7.11 Action at the time of unusual generating](#)" in this manual.

E.5.11.2 *cmdname*: file system configuration table entry for fsid (*fsid*) is invalid file system configuration table entry for proxy file system is invalid

Explanation

The contents of a management partition are inaccurate.

Response

Contact local Customer Support.

E.5.11.3 *cmdname*: *device*: file system already unlocked! backup may be invalid.

Explanation

After the lock of the renewal operation of a file system, although instant copy processing or separation processing was performed, the lock was canceled in the meantime. Backup may not be created correctly.

Response

Contact local Customer Support.

E.5.11.4 *cmdname*: *device*: file system must be stopped or mounted from this node

Explanation

Even if it has mounted by other nodes with the file system for operation, a command execution node is in a unmount state. Since it was in the state that cannot lock updating operation of a file system, execution of a command went wrong.

Response

At **sfcproxyjoin**, **sfcproxyop backup**, **sfcproxy part**, and **sfcproxybreak**, a unmount state and a command execution node needs to be in a mount state in all the nodes of a file system. Please check the state of a file system.

E.5.11.5 *cmdname*: *device*: not part status *cmdname*: *device*: not join status

Explanation

The command which executed the command that should be executed when a backing up agency and a backup place are in a joint state in the state of separation, or should execute it in the case of a separation state was executed in the state of combination.

Response

Please improve the executed command and the joint state of a backing up agency and a backup place.

**E.5.11.6 sfcproxybreak: *device*: master volume is copy status
sfcproxybreak: *device*: proxy volume is copy status**

Explanation

It was going to cancel correlation in the state a backing up agency or under copy of a backup place.

Response

In under copy, correlation cannot be canceled. Please wait for completion of a copy, or specify **-f** option, and perform compulsive release. However, when compulsive release is performed, the volume of a copy place is data unjust (**INVALID**).

E.5.11.7 sfcproxyjoin: *device1 device2*: not same class

Explanation

It was going to relate between the devices of the class from which GDS differs.

Response

Please improve the partition of a backup place the specified backup origin.

**E.5.11.8 sfcproxyjoin: *device*: already related as OPC
sfcproxyjoin: *device*: already related as EC**

Explanation

It was going to associate an employment pattern that is different in the file system of the same backup origin.

Response

Please improve the executed command.

**E.5.11.9 sfcproxyjoin: *device*: invalid partition specified for master file system
sfcproxyjoin: *device*: invalid partition specified for proxy file system**

Explanation

The partition inaccurate as a backing up agency or a backup place was specified. The partition of a different file system as a backing up agency or a backup place was specified.

Response

Please improve the partition composition of a backing up agency or a backup place.

E.5.11.10 sfcproxyjoin: *device*: invalid path for GDS

Explanation

The specified path is inaccurate. This message is outputted when devices other than GDS are specified.

Response

Please check the specified path.

E.5.11.11 sfcproxyjoin: host information not same: *device1* and *device2*

Explanation

A setup of the MDS node of the file system *device1* of a backing up agency and the file system *device2* of a backup place and AC node was not in agreement.

Response

Please improve a setup of a backing up agency and the MDS node of a backup place, and AC node. When the file system of a backup place is unnecessary, please delete the file system of a backup place and re-perform.

E.5.11.12 sfcproxyjoin: size not same: *device1* and *device2*

Explanation

The size of the partition *device1* of a backing up agency and the partition *device2* of the backup place corresponding to it is not equal.

Response

Please improve the partition composition of a backing up agency or a backup place.

E.5.11.13 sfcproxyjoin: WARNING: Partially joined. Execute following command after copy is completed:

<cmd>
:

Explanation

Although a part of correlation of GDS was performed, before correlation of all partitions was performed, the error occurred. Correlation will be set up partially.

Response

Please execute the shown command after completion of copy processing, and cancel the correlation by the side of GDS. Please specify and perform **-e force** at the last of the **sdxproxy break** command to cancel correlation, without waiting for completion of copy processing. However, when **-e force** is attached and performed, the volume of a copy place may be data unjust (**INVALID**). It is necessary to investigate the cause used as the error after release of correlation.

E.5.11.14 sfcproxyop: <device>: FSLOG file system, already mounted by another node

Explanation

It is not in the state that can be mounted. It is already mounted from other nodes, or the node down occurred during mount. It is necessary to investigate the cause used as the error after release of correlation.

Response

It cannot mount, when already mounted from other nodes. When a node down occurs during mount, please mount after restoration of a file system by **fsck_sfcs(1M)**.

E.5.11.15 sfcproxyop: WARNING: Partially parted. Execute sfcproxybreak and settle the problem.

sfcproxyop: WARNING: Partially rejoined. Execute sfcproxybreak and settle the problem.

Explanation

Although a part of inclusion processing of GDS or separation processing was performed, before processing of all partitions was performed, the error occurred. The joint state and the separation state will be intermingled.

Response

Please cancel correlation by **sfcproxybreak(1M)**. It is necessary to investigate the cause used as the error after release of correlation.

E.5.11.16 sfcproxyrestore: *device*: file system must be stopped

Explanation

All the nodes of the file system of a backing up agency (restoration place) were not in the unmount state.

Response

Since all the nodes of the file system of a backing up agency (restoration place) are changed into a unmount state, please re-perform.

E.5.12 sfcmbkup command

E.5.12.1 sfcmbkup: too long name *file-name*

Explanation

The specified file name (*file-name*) is too long.

Response

Check the specified file name.

E.5.12.2 sfcmbkup: illegal lines in mnttab

Explanation

There was an illegal entry in */etc/mnttab*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in */etc/vfstab*.

E.5.12.3 sfcmbkup: illegal lines in vfstab

Explanation

The */etc/vfstab* file is invalid.

Response

Correct the */etc/vfstab* file, or specify a special file instead of a mount point.

E.5.12.4 sfcmbkup: *file* is not a mount-point

Explanation

The specified file (*file*) is not a mount point.

Response

Specify a mount point or a special file.

E.5.13 sfcmrstr command

E.5.13.1 sfcmrstr: *file* is not a mount-point

Explanation

The specified file is not a mount point.

Response

Specify a mount point or a special file.

E.5.13.2 sfcmrstr: illegal lines in mnttab

Explanation

There was an illegal entry in */etc/mnttab*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in */etc/vfstab*.

E.5.13.3 sfcmrstr: illegal lines in vfstab

Explanation

The */etc/vfstab* file is invalid.

Response

Correct the */etc/vfstab* file, or specify a special file instead of a mount point.

E.5.13.4 sfcmrstr: too long name *file-name*

Explanation

The specified file name (*file-name*) is too long.

Response

Check the specified file name.

E.5.14 sfcmntgl command

E.5.14.1 ERROR: sfcmntgl: 0001: Permission denied

Explanation

Execution was attempted with other than super user permission.

Response

Retry execution with super user permission.

E.5.14.2 ERROR: sfcmntgl: 0002: Cannot open vfstab file

Explanation

/etc/vfstab could not be opened.

Response

Make sure that */etc/vfstab* exists. If not, restore it from a backup, or see [Section 20.2.3, "Setting */etc/vfstab*"](#) and redefine it.

E.5.14.3 ERROR: sfcmntgl: 0003: Both *special* and *mount_point* need to be specified in vfstab

Explanation

Special file *special*, or mount point *mount_point* is not described in */etc/vfstab*.

Response

Check the status of */etc/vfstab*.

E.5.14.4 ERROR: sfcmntgl: 0004: {*specila* | *mount_point*} not in vfstab

Explanation

The specified *filename* cannot be found in */etc/vfstab*.

Response

Check the status of */etc/vfstab*.

E.5.14.5 ERROR: sfcmntgl: 0005: {*specila* | *mount_point*}: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.14.6 ERROR: sfcmntgl: 0007: {*specila* | *mount_point*}: Cannot connect to cluster foundation

Explanation

Connection to the cluster service failed.

Response

Confirm that the cluster control facility has already been started.

E.5.14.7 ERROR: sfcmntgl: 0008: {*special* | *mount_point*}: Cannot get lock: config

Explanation

The lock to change the file system configuration information cannot be obtained. The lock to change the file system configuration information is now being obtained in an optional process of the node within the cluster configuration.

Response

Release the lock for changing the domain configuration information, and then re-execute.

E.5.14.8 ERROR: sfcmntgl: 0009: {*special* | *mount_point*}: Read error on management partition to get fsid

Explanation

Reading of the management partition to obtain the file system id failed.

Response

Confirm, using the likes of the **ps(1)** command, that sfcfrmd daemon is available.

E.5.14.9 ERROR: sfcmntgl: 0010: {*specila* | *mount_point*}: Cannot get lock: fsid

Explanation

Exclusive lock for fsid within the domain cannot be acquired. The exclusive lock for fsid has been acquired by another process running on a node in the cluster.

Response

Wait for the exclusive lock for *fsid* to be released, and then retry.

E.5.14.10 ERROR: sfcmntgl: 0011: {*special* | *mount_point*}: Read error on management partition to get node infomation

Explanation

Reading of the management partition to obtain the cluster configuration node information failed.

Response

Confirm, using the likes of the **ps(1)** command, that sfcfrmd daemon is available.

E.5.14.11 ERROR: sfcmntgl: 0012: *special* was not able to mount. Other process performed umount or some error occurred. Refer console information on each node for detailed error information.

Explanation

Failed to mount *special* because another node was unmounting it.

Response

Unmount it and perform **sfcmntgl(1M)** command again.

E.5.14.12 ERROR: sfcmntgl: 0013: {special | mount_point}: sfcfrmd may not exist

Explanation

A communications channel with sfcfrmd daemon could not be established.

Response

Check whether sfcfrmd daemon is operating using the likes of the **ps(1)** command, and then contact local Customer Support.

E.5.14.13 ERROR: sfcmntgl: 0018: The description form of vfstab is wrong

Explanation

The description format of */etc/vfstab* is incorrect.

Response

Correct the definition of */etc/vfstab*.

E.5.14.14 ERROR: sfcmntgl: 0019: {special | mount_point}: filesystem may not be created

Explanation

The specified file system {*special | mount_point*} was not registered in the management partition.

Response

Show a list of all the GFS Shared File System by **sfcinfo(1M)** command with **-a** option and make sure that the file system appears that was specified as an argument to **sfcmntgl(1M)** command.

E.5.14.15 ERROR: sfcmntgl: 0028: {special | mount_point}: hostname is already processing mount or umount

Explanation

In the host (*hostname*), mount or umount processing has already been performed.

Response

Please set time a little and execute a command again.

E.5.14.16 ERROR: sfcmntgl: 0029: {special | mount_point}: mount_point need to be representative partition

Explanation

The special block device specified as a mount point was not a representation partition.

Response

Show a list of all the GFS Shared File System by **sfcinfo(1M)** command with **-a** option and find the representative partition. Again perform **sfcmntgl(1M)** command with the name of the representative partition as an argument.

E.5.14.17 WARNING: sfcmntgl: 0020: *hostname: special* is already mounted, *mount_point* is busy, or the allowable number or mount points has been exceeded

Explanation

Either special file *special* has already been mounted in *hostname*, or mount point *mount_point* is already in use. Mount processing is not performed for *hostname*.

Response

Please check the process that is used *mountpoint* by **fuser(1M)**. Or check the file system that is used *mount_point* by **mount(1M)**.

E.5.14.18 WARNING: sfcmntgl: 0025: *hostname: {special | mount_point}*: Send mount demand processing to sfcfrmd went wrong

Explanation

Transmission processing for sfcfrmd daemon of the mount processing request made to *hostname* failed.

Response

After confirming the state of sfcfrmd daemon with the likes of the **ps(1)** command, contact local Customer Support.

E.5.14.19 WARNING: sfcmntgl: 0026: *{special | mount_point}*: Waiting processing for a response mount demand went wrong

Explanation

Reception processing of the mount processing request made to sfcfrmd daemon failed.

Response

After confirming the state of sfcfrmd daemon with the likes of the **ps(1)** command, contact local Customer Support.

E.5.14.20 WARNING: sfcmntgl: 0027: *{special |mount_point}*: Time out occurred in mount demand processing to sfcfrmd

Explanation

A timeout occurred in the reception processing of the mount processing request made to sfcfrmd daemon.

Response

After confirming the state of sfcfrmd daemon with the likes of the **ps(1)** command, contact local Customer Support.

E.5.15 sfcnewfs command

E.5.15.1 sfcnewfs: ERROR: -*: unknown flag

Explanation

An illegal option was specified.

Response

Check the specified options.

E.5.15.2 sfcnewfs:ERROR:argument is too long

Explanation

`mkfs_sfcs(1M)` could not be executed because too many parameter values were specified.

Response

Check the specified parameter values.

E.5.15.3 sfcnewfs:ERROR: can't find *hostname* as hostname

Explanation

Command argument *hostname* is wrong, or a file system recovery manager is not active on *hostname*.

Response

Check command argument *hostname*. If a file system recovery manager is not active on *hostname*, execute `sfcrmstart(1M)` command.

E.5.15.4 sfcnewfs: ERROR: Cannot open *file*: *errmsg*

Explanation

The specified file indicated by *file* could not be accessed for the reason indicated by *errmsg*.

Response

Check whether the specified file is correct.

E.5.15.5 sfcnewfs: ERROR: Cannot stat *file*: *errmsg*

Explanation

The specified file indicated by *file* could not be accessed for the reason indicated by *errmsg*.

Response

Check whether the specified file is correct.

E.5.15.6 sfcnewfs: ERROR: -C: cannot specified other options

Explanation

The `-C` option cannot be specified together with other options except `v` and `N`.

Response

Specify other options in the parameter file.

E.5.15.7 sfcnewfs: ERROR: -C: missing parameter file

Explanation

No argument was specified for the -C option.

Response

Specify a parameter file.

E.5.15.8 sfcnewfs: ERROR: -C: missing meta partition in *file*

Explanation

No representative partition was specified.

Response

Specify a representative partition in the parameter file.

E.5.15.9 sfcnewfs: ERROR: dataopt: bad dataopt preference

Explanation

The specified parameter value indicated by *dataopt* is illegal.

Response

Check the specified parameter value.

E.5.15.10 sfcnewfs: ERROR: data partition not specified

Explanation

No data partition was specified.

Response

Specify a data partition in which to place the data area.

E.5.15.11 sfcnewfs: ERROR: Line number *n* is bad entry in *file*

Explanation

A problem was found in line *n* in the file specified for the -C option.

Response

Check the explanation of the file specified for the -C option.

E.5.15.12 sfcnewfs: ERROR: *logsz*: bad update log size

Explanation

The specified parameter value indicated by *logsz* is invalid.

Response

Check the specified parameter value.

E.5.15.13 sfcnewfs: ERROR: malloc(3C) failed

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.15.14 sfcnewfs: ERROR: missing bytes per inode of *string*

Explanation

There is no argument with **nbpi** option.

Response

Specify the number of bytes per i-node according to the specification method indicated by *string*.

E.5.15.15 sfcnewfs: ERROR: missing dataopt preference of *string*

Explanation

There is no argument with **dataopt** option.

Response

Specify **y** or **n** according to the specification method indicated by *string*.

E.5.15.16 sfcnewfs: ERROR: missing data partition (raw-special-device) of *string*

Explanation

No data partition was specified as an argument.

Response

Specify a data partition according to the specification method indicated by *string*.

E.5.15.17 sfcnewfs: ERROR: missing hostname of *string*

Explanation

No host name was specified as an argument.

Response

Specify a host name group for locating the meta-data server according to the specification method indicated by *string*.

E.5.15.18 sfcnewfs: ERROR: missing hostname of *string*

Explanation

No host name was specified as an argument.

Response

Specify a host name group for sharing the file system according to the specification method indicated by *string*.

E.5.15.19 sfcnewfs: ERROR: missing max data size of *string*

Explanation

There is no argument with **maxdsz** option.

Response

Specify the maximum size of the data area according to the specification method indicated by *string*.

E.5.15.20 sfcnewfs: ERROR: missing meta data size of *string*

Explanation

There is no argument with **metasz** option.

Response

Specify the meta-data area size according to the specification method indicated by *string*.

E.5.15.21 sfcnewfs: ERROR: missing meta partition of *string*

Explanation

No representative partition was specified as an argument.

Response

Specify a representative partition according to the specification method indicated by *string*.

E.5.15.22 sfcnewfs: ERROR: missing minfree space % of *string*

Explanation

There is no argument with **mimfree** option.

Response

Specify the minimum percentage ($0 < n < 99$) of the unused file system area according to the specification method indicated by *string*.

E.5.15.23 sfcnewfs: ERROR: missing mminfree space % of *string*

Explanation

There is no argument with **mminfree** option.

Response

Specify the minimum percentage ($0 < n < 99$) of the unused V-data area according to the specification method indicated by *string*.

E.5.15.24 sfcnewfs: ERROR: missing number of blocks per one extents of *string*

Explanation

There is no argument with **nblkpext** option.

Response

Specify the average number of blocks managed in one extent according to the specification method indicated by *string*.

E.5.15.25 sfcnewfs: ERROR: missing number of maxnode of *string*

Explanation

There is no argument with **maxnode** option.

Response

Specify the maximum number of nodes sharing a file system according to the specification method indicated by *string*.

E.5.15.26 sfcnewfs: ERROR: missing number of maxvol of *string*

Explanation

There is no argument with **maxvol** option.

Response

Specify the maximum number of partitions according to the specification method indicated by *string*.

E.5.15.27 sfcnewfs: ERROR: missing parameter of -S all|*num*

Explanation

The MDS operational information is specified incorrectly.

Action

Specify the MDS operational information correctly.

E.5.15.28 sfcnewfs: ERROR: missing optimization preference of *string*

Explanation

There is no argument with **opt** option.

Response

Specify an optimization parameter according to the specification method indicated by *string*.

E.5.15.29 sfcnewfs: ERROR: missing update log size of *string*

Explanation

There is no argument with **logsz** option.

Response

Specify the update log area size according to the specification method indicated by *string*.

E.5.15.30 sfcnewfs: ERROR: missing update log partition (raw-special-device) of *string*

Explanation

There is no argument with **log** option.

Response

Specify an update log partition according to the specification method indicated by *string*.

E.5.15.31 sfcnewfs: ERROR: number of mdsnode over *num*

Explanation

The MDS operational information exceeded the limit of *num* that can be specified.

Action

Limit the MDS operational information to *num*.

E.5.15.32 sfcnewfs: ERROR: *maxdsz*: bad max data size

Explanation

The specified parameter value indicated by *maxdsz* is illegal.

Response

Check the specified parameter value.

E.5.15.33 sfcnewfs: ERROR: *maxnode*: bad number of maxnode

Explanation

The specified parameter value indicated by *maxnode* is illegal.

Response

Check the specified parameter value.

E.5.15.34 sfcnewfs: ERROR: *maxvol*: bad number of maxvol

Explanation

The specified parameter value indicated by *maxvol* is illegal.

Response

Check the specified parameter value.

E.5.15.35 sfcnewfs: ERROR: *metasz*: bad meta data size

Explanation

The specified parameter value indicated by *metasz* is illegal.

Response

Check the specified parameter value.

E.5.15.36 sfcnewfs: ERROR: *minfree*: bad minfree space %

Explanation

The specified parameter value indicated by *minfree* is illegal.

Response

Check the specified parameter value.

E.5.15.37 sfcnewfs: ERROR: *mminfree*: bad mminfree space %

Explanation

The specified parameter value indicated by *mminfree* is illegal.

Response

Check the specified parameter value.

E.5.15.38 sfcnewfs: ERROR: *nblkpext*: bad number of blocks per one extents

Explanation

The specified parameter value indicated by *nblkpext* is illegal.

Response

Check the specified parameter value.

E.5.15.39 sfcnewfs: ERROR: *nbpi*: bad bytes per inode

Explanation

The specified parameter value indicated by *nbpi* is illegal.

Response

Check the specified parameter value.

E.5.15.40 sfcnewfs: ERROR: *number*: number of partition is maxvol over

Explanation

The number of specified partitions (*number*) exceeded the upper limit.

Response

The maximum number of partitions configuring a file system can be specified using **maxvol** option. Check whether the specified number of partitions is correct.

E.5.15.41 sfcnewfs: ERROR: *opt*: bad optimization preference

Explanation

The specified parameter value indicated by *opt* is illegal.

Response

Check the specified parameter value.

E.5.15.42 sfcnewfs:WARNING: putenv failed

Explanation

putenv(3C) could not be executed because of insufficient memory of system.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.15.43 sfcnewfs: ERROR: same hostname are specified

Explanation

Only one meta-data server can be deployed for one host name.

Response

Change the specification to deploy only one meta-data server or to deploy each meta-data server on a different host.

E.5.15.44 sfcnewfs: ERROR: *special* is not a raw disk device

Explanation

The specified file indicated by *special* is not a character special file.

Response

Specify a character special file.

E.5.16 sfcquota command

E.5.16.1 sfcquota: ERROR: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.16.2 sfcquota: ERROR: Cannot lock /etc/mnttab

Explanation

Failed to acquire the lock of */etc/mnttab*.

Action

Reexecute the command after waiting until another command terminates operation.

E.5.16.3 sfcquota: ERROR: Cannot open /etc/mnttab

Explanation

The */etc/mnttab* file could not be opened and may be lost.

Action

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.16.4 sfcquota: ERROR: uid *uid* too large

Explanation

The specified user ID (*uid*) is too large.

Action

Reexecute the command after checking the user ID.

E.5.16.5 sfcquota: ERROR: *username*: unknown user

Explanation

The specified user name (*username*) is not found.

Response

Check the user name and re-execute the command.

E.5.16.6 sfcquota: ERROR: *username* (uid[*uid*]): permission denied

Explanation

A user other than a super user attempted to display other user data.

Response

Re-execute the command with the name of a log-in user.

E.5.16.7 sfcquota: WARNING: line in mnttab exceeds *num* characters

Explanation

The number of characters of the entry in **/etc/mnttab** is over maximum *num*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, change entry of above file system in **/etc/vfstab** not to exceed maximum *num*.

E.5.16.8 sfcquota: WARNING: line in mnttab has too few entries

Explanation

There was an illegal entry with too few fields in **/etc/mnttab**.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in **/etc/vfstab**.

E.5.16.9 sfcquota: WARNING: line in mnttab has too many entries

Explanation

There was an illegal entry with too many fields of **/etc/mnttab**.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in **/etc/vfstab**.

E.5.16.10 sfcquota: INFO: No quotas on this system

Explanation

No file system with a valid quota function.

Response

Enable the quota function and re-execute the command.

E.5.17 sfcquotadm command

E.5.17.1 sfcquotadm: ERROR: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.17.2 sfcquotadm: ERROR: Cannot delete. because quotas is turned on

Explanation

The internal quotas file cannot be deleted because the quota function is made valid.

Response

Invalidate the quota function, then re-execute the command.

E.5.17.3 sfcquotadm: ERROR: Cannot open /etc/mnttab

Explanation

The `/etc/mnttab` file could not be opened and may be lost.

Action

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.17.4 sfcquotadm: ERROR: file size error in *quotas*

Explanation

The external quotas file size is invalid.

Response

Execute `sfcquotadm(1M)` with `-e` option to reflect internal quotas file data to the external quotas file.

E.5.17.5 sfcquotadm: ERROR: File size of external quotas file is 0

Explanation

The external quotas file size is 0.

Response

Execute `sfcquotadm(1M)` with `-e` option to reflect internal quotas file data to the external quotas file.

E.5.17.6 sfcquotadm: ERROR: *filesystem* is read-only filesystem

Explanation

The specified file system has been mounted for read only.

Response

Specify a writable file system.

E.5.17.7 sfcquotadm: ERROR: *filesystem* not in mnttab

Explanation

The GFS Shared File System corresponding to the specified file system (*filesystem*) is not found in */etc/mnttab*.

Response

Check the file system and specify a file system being mounted.

E.5.17.8 sfcquotadm: ERROR: *fstatvfs* error in *filesystem*: *errmsg*

Explanation

Failed to acquire the specified file system (*filesystem*) status.

Response

Respond according to *errmsg*.

E.5.17.9 sfcquotadm: ERROR: illegal option

Explanation

An illegal option was specified.

Response

Check the specified option and re-execute the command with a correct option.

E.5.17.10 sfcquotadm: ERROR: *ioctl(Q_FGETQUOTA)* error in *quotas*: *errmsg*

Explanation

Failed to acquire data from the internal quotas file.

Response

A physical I/O error or communication error is possible. Confirm a console message to respond.

E.5.17.11 sfcquotadm: ERROR: *ioctl(Q_FSETQLIM)* error in *quotas*: *errmsg*

Explanation

Failed to reflect data to the internal quotas file.

Response

A physical I/O error or communication error is possible. Confirm a console message to respond.

E.5.17.12 sfcquotadm: ERROR: *ioctl(Q_IQDELETE)* error in *filesystem*: *errmsg*

Explanation

Failed to delete the internal quotas file.

Response

A physical I/O error or communication error is possible. Confirm a console message to respond.

E.5.17.13 sfcquotadm: ERROR: Length of mountpoint too long

Explanation

The pathname of mount point of the specified file system is too long.

Action

Reexecute the command after checking the pathname length of mount point and changing it to an appropriate length.

E.5.17.14 sfcquotadm: ERROR: no sfcfs filesystem with quotas file

Explanation

The specified file system is not the GFS Shared File System in which a quotas file exists.

Response

Specify a file system in which a quotas file exists.

E.5.17.15 sfcquotadm: ERROR: permission denied

Explanation

Super-user authorization is not used for execution.

Response

Use super-user authorization for execution.

E.5.17.16 sfcquotadm: ERROR: *quotas* is not a regular file

Explanation

The file named *quotas* under the mount point is not a normal file.

Response

Check the external *quotas* file status.

E.5.17.17 sfcquotadm: ERROR: read error in *quotas*: *errmsg*

Explanation

Failed to acquire data from the external *quotas* file.

Response

Respond according to *errmsg*.

E.5.17.18 sfcquotadm: ERROR: *special* not in *mnttab*

Explanation

The specified file is not found in */etc/mnttab*.

Response

Check the file system and specify a file system being mounted.

E.5.17.19 sfcquotadm: ERROR: *special* not block special file

Explanation

The specified device (*special*) is not a block special file.

Response

Specify a block special file.

E.5.17.20 sfcquotadm: WARNING: line in mnttab exceeds *num* characters

Explanation

The number of characters of the entry in */etc/mnttab* is over maximum *num*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, change entry of above file system in */etc/vfstab* not to exceed maximum *num*.

E.5.17.21 sfcquotadm: WARNING: line in mnttab has too few entries

Explanation

There was an illegal entry with too few fields in */etc/mnttab*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in */etc/vfstab*.

E.5.17.22 sfcquotadm: WARNING: line in mnttab has too many entries

Explanation

There was an illegal entry with too many fields of */etc/mnttab*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in */etc/vfstab*.

E.5.18 sfcquotaon, sfcquotaoff command

E.5.18.1 {sfcquotaon|sfcquotaoff}: ERROR: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.18.2 {sfcquotaon|sfcquotaoff}: ERROR: Cannot acquire lock: *type(num)*

Explanation

Cannot get the lock for file system. Other command or daemon is operating with lock.

Response

Please wait to complete operating the command or daemon of a GFS Shard file system in all nodes, and re-execute sfcquotaon(1M) or sfcquotaoff(1M) again.

E.5.18.3 {sfcquotaon|sfcquotaoff}: ERROR: Cannot get quota lock

Explanation

Failed to obtain quota lock because the quota function is being set.

Response

Wait until other **sfcquotaon/off(1M)** is finished, then re-execute the command.

E.5.18.4 {sfcquotaon|sfcquotaoff}: ERROR: Cannot get quota resource. *errmsg*

Explanation

Failed to obtain a quota resource.

Response

Respond according to *errmsg*.

E.5.18.5 {sfcquotaon|sfcquotaoff}: ERROR: Cannot lock /etc/mnttab

Explanation

Failed to lock **/etc/mnttab**.

Response

Wait until other commands are finished, then re-execute the command.

E.5.18.6 {sfcquotaon|sfcquotaoff}: ERROR: Cannot open /etc/vfstab

Explanation

Failed to open **/etc/vfstab**.

Response

Make sure that **/etc/vfstab** exists. If not, restore it from a backup, or see [Section 20.2.3, "Setting](#)

[/etc/vfstab](#)" and redefine it.

E.5.18.7 {sfcquotaon|sfcquotaoff}: ERROR: Cannot open /etc/mnttab

Explanation

The `/etc/mnttab` file could not be opened and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.18.8 {sfcquotaon|sfcquotaoff}: ERROR: Cannot open /etc/mnttab.temp

Explanation

Failed to open `/etc/mnttab` duplicate file.

Response

Check the system environment.

E.5.18.9 {sfcquotaon|sfcquotaoff}: ERROR: Cannot stat /etc/mnttab

Explanation

The `/etc/mnttab` file could not be get stat information and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.18.10 {sfcquotaon|sfcquotaoff}: ERROR: ioctl(Q_QUOTAON) error in *quotas: errmsg*

Explanation

Failed to make quota function valid.

Response

A physical I/O error or communication error is possible. Confirm a console message to respond.

E.5.18.11 {sfcquotaon|sfcquotaoff}: ERROR: *mount_point* cannot open

Explanation

Failed to open the mount point (*mount_point*).

Response

Check the system environment.

E.5.18.12 {sfcquotaon|sfcquotaoff}: ERROR: *mount_point* not a directory

Explanation

The specified mount point (*mount_point*) is not a directory.

Response

Check the system environment.

E.5.18.13 {sfcquotaon|sfcquotaoff}: ERROR: *special* is mounted by other node

Explanation

The quota function setting cannot be changed because the specified device is mounted by another node.

Response

Mount the own node, or execute the command with the device not mounted by another node.

E.5.18.14 {sfcquotaon|sfcquotaoff}: ERROR: *special* not a block device

Explanation

The specified device (*special*) is not a block special file.

Response

Specify a block special file.

E.5.18.15 {sfcquotaon|sfcquotaoff}: ERROR: (*special*) Distribution of mnttab entries were failed. Unexpected error occurred.

Explanation

Due to an unexpected error, */etc/mnttab* update processing for another node failed.

Action

Contact local Customer Support.

E.5.18.16 {sfcquotaon|sfcquotaoff}: ERROR: (*special: offline*) Update of file system configuration table was failed.

Explanation

The settings for validating or invalidating the quota function at the next mount failed.

Response

Apply the countermeasures described in the message that was output immediately before, in which the details of the error are described, and then re-execute the command.

E.5.18.17 {sfcquotaon|sfcquotaoff}: ERROR: (*special: online*) The first update of file system configuration table was failed.

Explanation

While the setting of the quota function was being changed, updating of the management partition, which shows the state of the setting, failed.

Response

Apply the countermeasures described in the message that was output immediately before, in which the details of the error are described, and then re-execute the command.

E.5.18.18 {sfcquotaon|sfcquotaoff}: ERROR: (*special*: online) The last update of file system configuration table was failed.

Explanation

After changing the settings of the quota function, updating of the management partition displaying the state of the setting failed.

Response

Apply the countermeasures described in the message that was been output immediately before, in which details of the error are described, and then re-execute the command.

E.5.18.19 {sfcquotaon|sfcquotaoff}: WARNING: line in mnttab exceeds *num* characters

Explanation

The number of characters of the entry in */etc/mnttab* is over maximum *num*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, change entry of above file system in */etc/vfstab* not to exceed maximum *num*.

E.5.18.20 {sfcquotaon|sfcquotaoff}: WARNING: line in mnttab has too few entries

Explanation

There was an illegal entry with too few fields in */etc/mnttab*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in */etc/vfstab*.

E.5.18.21 {sfcquotaon|sfcquotaoff}: WARNING: line in mnttab has too many entries

Explanation

There was an illegal entry with too many fields of */etc/mnttab*.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in */etc/vfstab*.

E.5.18.22 {sfcquotaon|sfcquotaoff}: WARNING: (*special*) Distribution of mnttab entries were failed. Detect error at remote node.

Explanation

/etc/mnttab update failed at a node other than the local node.

Action

Check the */etc/mnttab* status at each node.

E.5.18.23 {sfcquotaon|sfcquotaoff}: WARNING: (*special*) Distribution of mnttab entries were failed. Stopped instances or down nodes exist.

Explanation

Due to a stopped instance or node, **/etc/mnttab** update processing could not be performed at some nodes.

Action

Check the status of each node or instance using **sfcrcinfo(1M)** or the GFS management view.

E.5.18.24 {sfcquotaon|sfcquotaoff}: WARNING: (*special*) Distribution of mnttab entries were timeout. Communication error occurred.

Explanation

Timeout of **/etc/mnttab** update processing to another node occurred due to communication error. Updates may have failed.

Action

Check the **/etc/mnttab** status at each node.

E.5.19 sfcprepquota command

E.5.19.1 sfcprepquota: ERROR: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.19.2 sfcprepquota: ERROR: Cannot open /etc/mnttab: *errmsg*

Explanation

The */etc/mnttab* file could not be opened and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.19.3 sfcprepquota: ERROR: FSID is invalid

Explanation

The identification number of the specified file system is not registered in the file system configuration information.

Response

Execute **sfcinfo(1M)** to confirm the identification number of the file system.

E.5.19.4 sfcprepquota: ERROR: ioctl(Q_FGETQUOTA) error in *quotas*: *errmsg*

Explanation

Failed to obtain data from the internal *quotas* file.

Response

Respond according to *errmsg*.

E.5.19.5 sfcprepquota: ERROR: Invalid word “xxx” exists in the file system configuration information

Explanation

Settings of the quota function are incorrect. The environment file of the GFS Shared File System contains an invalid character string.

Action

Check the quota function settings of the applicable file system using **sfcinfo(1M)**, then set the quota function again using **sfcquotaon(1M)** and **sfcquotaoff(1M)**.

E.5.19.6 sfcprepquota: ERROR: file system configuration information not found

Explanation

Failed to get the file system configuration information in management partition.

Response

Restore the file system configuration information. For detail, see [chapter 16, "Management Partition."](#)

E.5.19.7 sfcprepquota: ERROR: Length of mountpoint too long

Explanation

The pathname length of mount point of the specified file system is too long.

Response

Reexecute the command after checking the pathname length of mount point and changing it to an appropriate length.

E.5.19.8 sfcprepquota: ERROR: Permission denied

Explanation

A user without super user authority attempted to display information.

Response

Execute with super user authority.

E.5.19.9 sfcprepquota: ERROR: Unknown error(*errorid*): *errmsg*

Explanation

An unexpected error occurred.

Response

Note *errmsg* and contact local Customer Support.

E.5.19.10 sfcprepquota: WARNING: last setting of quotas were not completed or quotas are setting now.

Explanation

The last **sfcquotaon/off(1M)** terminated abnormally or the quota function is currently being set.

Response

Check the quota function settings of the applicable file system using **sfcinfo(1M)**, then set the quota function again using **sfcquotaon/off(1M)**.

E.5.19.11 sfcprepquota: WARNING: line in mnttab exceeds *num* characters

Explanation

The number of characters of the entry in **/etc/mnttab** is over maximum *num*.

Response

Check **/etc/mnttab** status

E.5.19.12 sfcrepquota: WARNING: line in mnttab has too few entries

Explanation

There was an illegal entry with too few fields in **/etc/mnttab**.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in **/etc/vfstab**.

E.5.19.13 sfcrepquota: WARNING: line in mnttab has too many entries

Explanation

There was an illegal entry with too many fields of **/etc/mnttab**.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in **/etc/vfstab**.

E.5.20 sfcrcinfo command

E.5.20.1 sfcrcinfo: *special* is not a representative partition of GFS shared file system

Explanation

The special file (*special*) is not the GFS Shared File System's representative partition.

Response

Again perform **sfcmntgl(1M)** command with the name of the representative partition as an argument. If it is unclear of the representative partition, perform **sfcrcinfo(1M)** command with **-am** option to show information of all the GFS Shared File Systems.

E.5.20.2 sfcrcinfo: fsid(*fsid*) is out of range

Explanation

The specified *fsid* value lies outside the valid range.

Response

Examine the value of *fsid* and try again. If it is unclear of the representative partition, perform **sfcrcinfo(1M)** command with **-am** option to show information of all the GFS Shared File Systems.

E.5.20.3 sfcrcinfo: There is not GFS shared file system with fsid(*fsid*)

Explanation

A GFS Shared File System identified by the specified *fsid* does not exist.

Response

Examine the value of *fsid* and try again. If it is unclear of the representative partition, perform **sfcrcinfo(1M)** command with **-am** option to show information of all the GFS Shared File Systems.

E.5.20.4 sfcrcinfo: Cannot allocate communication buffer

Explanation

The initialization processing of communication failed due to insufficient memory or swap space.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.20.5 sfcrcinfo: Cannot connect to fsrm

Explanation

The file system monitoring mechanism is not operating.

Response

Check whether the *sfcfrmd* daemon exists using the **ps(1)** command. When it does not exist, activate the file system monitoring mechanism using the **sfcfrmstart(1M)** command.

E.5.20.6 sfcrcsinfo: Fs(*fsid*) config db broken

Explanation

Illegal data exists in the file system configuration information for the file system whose file system id is *fsid*.

Response

Restore the management partition using its backup previously saved.

E.5.20.7 sfcrcsinfo: Cannot get device name of *fsid(fsid)*

Explanation

Obtaining the device name for the file system whose file system id is *fsid* failed.

Response

Check the system environment whether it is proper.

E.5.20.8 sfcrcsinfo: File system config data of *fsid(fsid)* is broken

Explanation

Illegal data exists in the file system configuration information for the file system whose file system id is *fsid*.

Response

Restore the management partition using its backup previously saved.

E.5.20.9 sfcrcsinfo: Cannot open */etc/mnttab*: *errmsg*

Explanation

The */etc/mnttab* file could not be opened and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.21 sfcsetup command

E.5.21.1 ERROR: sfcsetup: 0001: Not super user

Explanation

Execution was attempted with other than super user permission.

Response

Perform execution with super user permission.

E.5.21.2 ERROR: sfcsetup: 0002: Cannot connect to cluster foundation

Explanation

Communication with the cluster control could not be established.

Response

Check whether the cluster control is operating normally.

E.5.21.3 ERROR: sfcsetup: 0003: sfcfrmd is active within cluster foundation

Explanation

Cannot change setup, because sfcfrmd daemon is active.

Response

Stop sfcfrmd daemon with **sfcfrmdstop(1M)** and then re-execute the command.

E.5.21.4 ERROR: sfcsetup: 0004: Cannot get lock: config

Explanation

Execution of this command failed as another command was being executed.

Response

Once execution of the other command has terminated, re-execute this command.

E.5.21.5 ERROR: sfcsetup: 0005: Too many arguments

Explanation

Too many arguments have been specified.

Response

Check the command format, and then re-execute.

E.5.21.6 ERROR: sfcsetup: 0006: A few option

Explanation

Too few arguments have been specified.

Response

Check the command format, and then re-execute.

E.5.21.7 ERROR: sfcsetup: 0007: Wrong command format

Explanation

The command format is invalid.

Response

Check the command format, and then re-execute.

E.5.21.8 ERROR: sfcsetup: 0008: Cannot get cipname

Explanation

The CIP address for the node executing the command could not be acquired.

Response

By remaking and restoring the management partition, back the configuration of the partition to a previous one before a value of the CIP address changed. For details to remake and restore, see [Section 16.5, "Backup and restore of the management partition information."](#)

E.5.21.9 ERROR: sfcsetup: 0009: Bad data exists in management partition

Explanation

The management partition contain invalid data.

Response

Contact local Customer Support.

E.5.21.10 ERROR: sfcsetup: 0010: Cannot write to sfcfsrm.conf

Explanation

Failed to write into `/var/opt/FJSVsfcs/sfcfsrm.conf`.

Response

Make sure that `/var` directory has a space by `df(1M)` command. Make sure that `/var/opt/FJSVsfcs` directory exists. If it dose not exist, it is possible to fail to install the FJSVsfcs package of GFS Shared File System. In this case, install the FJSVsfcs package again. Otherwise, collect the diagnostic data with `fjsnap` and contact local Customer Support.

E.5.21.11 ERROR: sfcsetup: 0011: Cannot open management partition

Explanation

The management partition could not be opened.

Response

Check whether the path to the management partition, specified in the `sfcfsrm.conf` file, is correct.

E.5.21.12 ERROR: sfcsetup: 0012: Node information area is full

Explanation

The maximum possible amount of node information has already been stored.

Response

Delete information of an unnecessary node by `sfcsetup(1M)` command with `-d` option and do it again.

E.5.21.13 ERROR: **sfcsetup: 0013: Cannot write node information**

Explanation

An error occurred when trying to write new node information.

Response

Contact local Customer Support.

E.5.21.14 ERROR: **sfcsetup: 0014: Wrong argument format**

Explanation

The format of an argument specified with the command is invalid.

Response

Check the format of the argument, and then re-execute the command.

E.5.21.15 ERROR: **sfcsetup: 0015: Invalid argument**

Explanation

An argument specified with the command contains an invalid character string.

Response

Check the argument, and then re-execute the command.

E.5.21.16 ERROR: **sfcsetup: 0016: Cannot get lock: master. sfcfrmd may be working now**

Explanation

The command cannot be executed, either because another command is being executed, or because **sfcfrmd** daemon is operating.

Response

Wait for execution of the other command to terminate or, after executing **sfcfrmstop(1M)** on all the nodes, re-execute the command.

E.5.21.17 ERROR: **sfcsetup: 0017: Node(hostid=0xhostid) is busy**

Explanation

Node (hostid=0xhostid) has still file system information in management partition. It is configured as sharing node of a GFS shared file system. **sfcsetup -d** did not do anything.

Response

Delete node information by following procedure:

1. Restart file system recovery manager by **sfcfrmstart(1M)**.
2. Check sharing node information of all GFS shared file system by **sfcinfo -an**.
3. Reconfigure sharing node information of GFS shared file system by **sfcadm -u**, or delete file system information by **sfcadm -D**.
4. Stop the file system recovery manager by **sfcfrmstop(1M)**.
5. Retry **sfcsetup -d**.

E.5.21.18 ERROR: sfcsetup: 0018: Communication initialize error

Explanation

Communication could not be initialized.

Response

Contact local Customer Support. Or, review your system environment.

E.5.21.19 ERROR: sfcsetup: 0020: data(*data*) is not found

Explanation

No information could be found for a node that was to be deleted.

Response

Show a state of the management partition by **sdxinfo(1)** command of GDS and make sure that all of the below conditions hold. (For details of the command, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide.")

- The GDS class that includes the management partition must show "shared" in "TYPE" and must show all the nodes that use GFS Shared File System in "SCOPE."

- The volume that is specified as the management partition must have 40MB or more capacity. (In other words, the value of "BLOCKS" is equal to or greater than 8192.)

- The slice that is specified as the management partition must show "ACTIVE" in "STATUS."

If not, see [Chapter 16, "Management Partition"](#) and try the setting of the management partition again. If this message is output when all the conditions above hold, collect the diagnostic data with fjsnap and contact local Customer Support.

E.5.21.20 ERROR: sfcsetup: 0021: ERROR occured. errcode= *ecode*

Explanation

An internal error occurred.

Response

Contact local Customer Support.

E.5.21.21 ERROR: sfcsetup: 0022: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.21.22 ERROR: sfcsetup: 0023: Read error of sfcfsrm.conf

Explanation

Reading of sfcfsrm.conf failed.

Response

Collect the diagnostic data with `fjsnap` and contact local Customer Support.

E.5.21.23 ERROR: sfcsetup: 0025: sfcfrmd is active on this node

Explanation

The file system recovery manager daemon is running on a node for which an attempt was made to execute a command.

Response

Execute `sfcfrmdstop(1M)`, and then re-execute the command.

E.5.21.24 ERROR: sfcsetup: 0026: Cluster function may not be working

Explanation

The cluster system may not be operating normally

Response

Contact local Customer Support.

E.5.21.25 ERROR: sfcsetup: 0027: Input error

Explanation

An input error was made.

Response

Contact local Customer Support.

E.5.21.26 ERROR: sfcsetup: 0028: *raw_device_file* is already used as management partition

Explanation

The specified character device (*raw_device_file*) has already been initialized as a management partition.

Response

After checking whether this device really should be initialized, re-execute the command with the `-f` option specified.

E.5.21.27 ERROR: sfcsetup: 0029: Management partition read error

Explanation

Reading of the management partition failed.

Response

Show a state of the management partition by `sdxinfo(1)` command of GDS and make sure that all of the below conditions hold. (For details of the command, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide.")

- The GDS class that includes the management partition must show "shared" in "TYPE" and must show all the nodes that use GFS Shared File System in "SCOPE."

- The volume that is specified as the management partition must have 40MB or more capacity. (In other words, the value of "BLOCKS" is equal to or greater than 8192.)

- The slice that is specified as the management partition must show "ACTIVE" in "STATUS."

If not, see [Chapter 16, "Management Partition"](#) and try the setting of the management partition again. If this message is output when all the conditions above hold, collect the diagnostic data with fjsnap and contact local Customer Support.

E.5.21.28 ERROR: sfcsetup: 0030: Management partition write error

Explanation

Writing to the management partition failed.

Response

Show a state of the management partition by **sdxinfo(1)** command of GDS and make sure that all of the below conditions hold. (For details of the command, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide.")

- The GDS class that includes the management partition must show "shared" in "TYPE" and must show all the nodes that use GFS Shared File System in "SCOPE."

- The volume that is specified as the management partition must have 40MB or more capacity. (In other words, the value of "BLOCKS" is equal to or greater than 8192.)

- The slice that is specified as the management partition must show "ACTIVE" in "STATUS."

If not, see [Chapter 16, "Management Partition"](#) and try the setting of the management partition again. If this message is output when all the conditions above hold, collect the diagnostic data with fjsnap and contact local Customer Support.

E.5.21.29 ERROR: sfcsetup: 0031: Cannot initialize management partition. errcode=ecode

Explanation

Initialization of the management partition failed.

Response

Show a state of the management partition by **sdxinfo(1)** command of GDS and make sure that all of the below conditions hold. (For details of the command, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide.")

- The GDS class that includes the management partition must show "shared" in "TYPE" and must show all the nodes that use GFS Shared File System in "SCOPE."

- The volume that is specified as the management partition must have 40MB or more capacity. (In other words, the value of "BLOCKS" is equal to or greater than 8192.)

- The slice that is specified as the management partition must show "ACTIVE" in "STATUS."

If not, see [Chapter 16, "Management Partition"](#) and try the setting of the management partition again. If this message is output when all the conditions above hold, collect the diagnostic data with fjsnap and contact local Customer Support.

E.5.21.30 ERROR: sfcsetup: 0032: *raw_device_file* is invalid disk device

Explanation

Either the specified device (*raw_device_file*) does not exist, or is not a GDS volume.

Response

Show a state of the management partition by **sdxinfo(1)** command of GDS and make sure that all of the below conditions hold. (For details of the command, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide.")

- The GDS class that includes the management partition must show "shared" in "TYPE" and must show all the nodes that use GFS Shared File System in "SCOPE."

- The volume that is specified as the management partition must have 40MB or more capacity. (In other words, the value of "BLOCKS" is equal to or greater than 8192.)

- The slice that is specified as the management partition must show "ACTIVE" in "STATUS."

If not, see [Chapter 16, "Management Partition"](#) and try the setting of the management partition again. If this message is output when all the conditions above hold, collect the diagnostic data with `fjsnap` and contact local Customer Support.

E.5.21.31 ERROR: sfcsetup: 0033: Cannot open *vfstab*

Explanation

The */etc/vfstab* file could not be opened.

Response

Make sure that */etc/vfstab* exists. If not, restore it from a backup, or see [Section 20.2.3, "Setting */etc/vfstab*"](#) and redefine it.

E.5.21.32 ERROR: sfcsetup: 0034: *raw_device_file* is already exist in *vfstab*

Explanation

The device specified (*raw_device_file*) in the management partition may already be being used by the system.

Response

Either specify a different device in the management partition or, if an entry in */etc/vfstab* is not needed, delete it and then re-execute the command.

E.5.21.33 WARNING: sfcsetup: 0035: Illegal line exists in *vfstab*

Explanation

An invalid row is included in */etc/vfstab*.

Response

Confirm the contents of */etc/vfstab* and correct it if necessary.

E.5.21.34 ERROR: sfcsetup: 0036: Cannot open mnttab

Explanation

The `/etc/mnttab` file could not be opened and may be lost.

Response

The problem of Solaris OE occurs when generated in the multiuser mode. Please investigate and deal with the message of Solaris OE.

E.5.21.35 ERROR: sfcsetup: 0037: *raw_device_file* is already exist in mnttab

Explanation

The device specified (*raw_device_file*) in the management partition may already be being used by the system.

Response

Either specify a different device in the management partition or, if an entry in the `/etc/mnttab` file is not needed, delete it and then re-execute the command.

E.5.21.36 ERROR: sfcsetup: 0038: Illegal line exists in mnttab

Explanation

There was an illegal entry in `/etc/mnttab`.

Response

Please check whether the entry of which file system is a cause, and unmount it at once. Moreover, correct entry of above file system in `/etc/vfstab`.

E.5.21.37 ERROR: sfcsetup: 0039: Cannot open *raw_device_file*

Explanation

The specified device file (*raw_device_file*) could not be opened.

Response

Show a state of the management partition by `sdxinfo(1)` command of GDS and make sure that all of the below conditions hold. (For details of the command, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide.")

- The GDS class that includes the management partition must show "shared" in "TYPE" and must show all the nodes that use GFS Shared File System in "SCOPE."

- The volume that is specified as the management partition must have 40MB or more capacity. (In other words, the value of "BLOCKS" is equal to or greater than 8192.)

- The slice that is specified as the management partition must show "ACTIVE" in "STATUS."

If not, see [Chapter 16, "Management Partition"](#) and try the setting of the management partition again. If this message is output when all the conditions above hold, collect the diagnostic data with `fjsnap` and contact local Customer Support.

E.5.21.38 ERROR: sfcsetup: 0040: Cannot get vtoc information

Explanation

The VTOC information for the disk containing the management information could not be acquired.

Response

Show a state of the management partition by **sdxinfo(1)** command of GDS and make sure that all of the below conditions hold. (For details of the command, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide.")

- The GDS class that includes the management partition must show "shared" in "TYPE" and must show all the nodes that use GFS Shared File System in "SCOPE."

- The volume that is specified as the management partition must have 40MB or more capacity. (In other words, the value of "BLOCKS" is equal to or greater than 8192.)

- The slice that is specified as the management partition must show "ACTIVE" in "STATUS."

If not, see [Chapter 16, "Management Partition"](#) and try the setting of the management partition again. If this message is output when all the conditions above hold, collect the diagnostic data with fjsnap and contact local Customer Support.

E.5.21.39 ERROR: sfcsetup: 0041: Cannot get major number

Explanation

The major number of disk containing the management partition could not be obtained.

Response

Check whether the system environment is satisfactory, and whether the specified special file is correct.

E.5.21.40 ERROR: sfcsetup: 0042: Cannot get driver name

Explanation

The driver name for the disk containing the management partition could not be obtained.

Response

Check whether the system environment is satisfactory, and whether the specified special file is correct.

E.5.21.41 ERROR: sfcsetup: 0043: Cannot get node information

Explanation

Node information could not be acquired from the management partition.

Response

Collect the diagnostic data with fjsnap and contact local Customer Support.

E.5.21.42 ERROR: sfcsetup: 0044: Node information does not exist

Explanation

The information for the node for which the command was executed has already been deleted.

Response

None.

E.5.21.43 ERROR: sfcsetup: 0045: *raw_device_file* is not raw_device_file

Explanation

The specified device (*raw_device_file*) is other than a character special device.

Response

Check the device name, and then re-execute the command.

E.5.21.44 ERROR: sfcsetup: 0046: Illegal entry exists, or no entry of path to management partition in sfcfstrm.conf

Explanation

The sfcfstrm.conf file contains no entry that indicates the path to the management partition. Either node information has not yet been added to the management partition, or it was on a node that has already been deleted.

Response

Check whether node information has not yet been added to the management partition, or whether the node containing the entry has already been deleted.

E.5.21.45 ERROR: sfcsetup: 0047: *raw_device_file* does not exist

Explanation

The specified special file (*raw_device_file*) does not exist in the system environment.

Response

Check whether the specified special file is correct.

E.5.21.46 ERROR: sfcsetup: 0048: Too small disk size, need more than 40MB

Explanation

The size of the device specified in the management partition is too small.

Response

Specify a shared disk with a volume size of 40 MB or more.

E.5.21.47 WARNING: sfcsetup: 0049: Illegal entry exists, or no entry of path to management partition in sfcfstrm.conf

Explanation

An invalid row is included in setup file sfcfstrm.conf.

Response

Confirm the contents of the setup file sfcfstrm.conf.

E.5.21.48 ERROR: sfcsetup: 0050: Cannot open sfcfstrm.conf: msg

Explanation

Due to the reason displayed in the message (*msg*), opening of the FsRM setup file *sfcfstrm.conf* failed.

Response

Apply the countermeasures described in the message.

E.5.21.49 ERROR: sfcsetup: 0051: raw_device_file is not management partition

Explanation

The volume of the specified shared disk (*raw_device_file*) is not initialized for the management partition.

Response

Perform **sfcsetup(1M)** command with **-c** option to initialize the volume of the shared disk and then re-execute.

E.5.21.50 ERROR: sfcsetup: 0052: sfcfstrm.conf does not exist. sfcsetup may not be executed

Explanation

The setup file *sfcfstrm.conf* cannot be found. Node information may not be registered in the node.

Response

Register node information in the management partition by using the **sfcsetup(1M)** command.

E.5.21.51 ERROR: sfcsetup: 0053: Volume(raw_device_file) may not be active. Please activate volume

Explanation

The volume of the shared disk that has been specified in the management partition may not be valid.

Response

Make sure that the slice that is specified as the management partition shows "ACTIVE" in "STATUS" by **sdxinfo(1)** command. If not, start the slice by **sdxvolume(1M)** command of GDS. For details, see "PRIMECLUSTER Global Disk Services Configuration and Administration Guide."

E.5.21.52 INFO: sfcsetup: 0002: Waiting to be able to get lock:master

Explanation

The acquisition of domain-wide lock is in progress.

Response

Wait for a short while.

E.5.21.53 INFO: sfcsetup: 0003: Waiting to be able to get lock:config

Explanation

The acquisition of domain-wide lock is in progress.

Response

Wait for a short while.

E.5.22 sfcstat command

E.5.22.1 sfcstat:can't open /dev/kstat

Explanation

The `/dev/kstat` was not found.

Response

Check the system environment.

E.5.22.2 sfcstat:missing entry of sfcfs fstotal

Explanation

Statistical information could not be obtained.

Response

Check whether the file system is normal.

E.5.22.3 sfcstat:no active sfcfs

Explanation

The file system is not mounted.

Response

Mount the file system.

E.5.22.4 sfcstat:*special* not mounted

Explanation

special is not mounted.

Response

After mount the file system, re-execute the command.

E.5.23 sfctune command

E.5.23.1 Bad super block *special*:

Explanation

The super block in the specified special file indicated by *special* could not be read.

Response

Contact local Customer Support.

E.5.23.2 bread: llseek:

E.5.23.3 FS SEEK:

E.5.23.4 llseek:

Explanation

llseek(2) could not be executed for the specified special file indicated by *special*.

Response

Contact local Customer Support.

E.5.23.5 sfctune: cannot acquire lock for filesystem fsid =n

Explanation

Cannot get the lock for file system. Other command or daemon is operating with lock.

Response

Please wait to complete operating the command or daemon of a GFS Shard file system in all nodes, and re-execute sfctune(1M) again.

E.5.23.6 Cannot open *special*:

E.5.23.7 cannot open *special*

Explanation

The specified special file indicated by *special* could not be opened.

Response

Check the specified special file name indicated by *special*.

E.5.23.8 /etc/vfstab: open:

Explanation

The **/etc/vfstab** could not be opened.

Response

Make sure that **/etc/vfstab** exists. If not, restore it from a backup, or see [Section 20.2.3, "Setting /etc/vfstab"](#) and redefine it.

E.5.23.9 sfctune: ERROR: Cannot acquire lock: *type(num)*

Explanation

Cannot get the lock for file system. Other command or daemon is operating with lock.

Response

Please wait to complete operating the command or daemon of a GFS Shard file system in all nodes, and re-execute `sfctune(1M)` again.

E.5.23.10 sfctune: ERROR: File system is corrupted.

Explanation

The file system cannot be created because it is not shut down normally during partition addition.

Response

Repair the file system according to the following message using `fsck_sfdfs(1M)`. Complete partition addition using `sfadd(1M)`, then re-execute `sfctune(1M)`.

E.5.23.11 FS WRITE:

Explanation

The system could not write the specified special file indicated by *special*.

Response

Contact local Customer Support.

E.5.23.12 loctl:

Explanation

`ioctl(2)` could not be executed for the specified special file indicated by *special*.

Response

Contact local Customer Support.

E.5.23.13 malloc:

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.23.14 maximum contiguous block count changes from *current* to *new*

Explanation

The maximum number of contiguous blocks in the file system was changed from *current* to *new*.

Response

No response is required.

E.5.23.15 minimum percentage of free space changes from *current%* to *new%*

Explanation

The minimum percentage of the unused file system area was changed from *current* to *new*.

Response

No response is required.

E.5.23.16 optimization preference changes from *current* to *new*

Explanation

The optimization preference for file system strategy was changed from *current* to *new*.

Response

No response is required.

E.5.23.17 sfctune: Could not find raw device for *mount_point*

Explanation

The raw device corresponding to mount point (*mount_point*) specified in the */etc/mnttab* or the */etc/vfstab* file was not found.

Response

Check the specified mount point indicated by *mount_point*.

E.5.23.18 sfctune: malloc failed

E.5.23.19 sfctune: *option*: bad optimization preference (options are `space' or `time')

Explanation

An invalid option was specified for indicating the file system strategy optimization preference.

Response

Specify **space** or **time** as the optimization preference for the file system strategy.

E.5.23.20 sfctune: file system configuration information not found.

Explanation

The file system configuration information in the management partition was not found.

Response

Restore the file system configuration information. For detail, see [chapter 16, "Management Partition."](#)

E.5.23.21 sfctune: sfctune: maximum contiguous block count must be ≥ 1

Explanation

The specified maximum number of contiguous blocks is invalid. The maximum number of contiguous blocks must be 1 or more.

Response

Specify a value of 1 or more as the maximum number of contiguous blocks.

E.5.23.22 sfctune: *special*:

Explanation

stat(2) failed for the specified special file indicated by *special*.

Response

Check the specified special file name indicated by *special*.

E.5.23.23 sfctune: sfctune: bad minimum percentage of free space

Explanation

The specified minimum percentage for unused area is invalid.

Response

Specify a positive number as the minimum percentage of unused area.

E.5.23.24 sfctune: sfctune: bad Vnode minimum percentage of free space

Explanation

The specified minimum percentage for unused virtual node area is invalid.

Response

Specify a positive number as the minimum percentage of unused virtual node area.

E.5.23.25 read:

Explanation

The system could not read data from the specified special file indicated by *special*.

Response

Contact local Customer Support.

E.5.23.26 sfctune: *special* is mounted, can't tune

Explanation

The specified special file indicated by *special* is mounted. **sfctune(1M)** cannot be executed.

Response

Check the specified mount point. If there is no problem, unmount the specified mount point, and reexecute the command.

E.5.23.27 sfctune: *special*: not a block or character device

Explanation

The specified special file indicated by *special* is neither a block nor a character device.

Response

Check the specified special file name indicated by *special*.

E.5.23.28 sfctune: *special* is not used as sfcfs filesystem.

Explanation

The special file name indicated by *special* is not GFS shared file system.

Response

Check the specified special file name indicated by *special*.

E.5.23.29 sfctune: *special* not a meta partition

Explanation

The specified special file indicated by *special* is not a representative partition.

Response

Check the specified special file name indicated by *special*.

E.5.23.30 *special*: bad magic number

Explanation

The specified special file indicated by *special* is not a partition in the GFS Shared File System.

Response

Check the specified special file name indicated by *special*.

E.5.23.31 Vnode minimum percentage of free space changes from *current*% to *new*%

Explanation

The minimum percentage for unused virtual node area in the file system was changed from *current* to *new*.

Response

No response is required.

E.5.24 sfcumntgl command

E.5.24.1 ERROR: sfcumntgl: 0001: Permission denied

Explanation

Execution was attempted with other than super user permission.

Response

Retry execution with super user permission.

E.5.24.2 ERROR: sfcumntgl: 0002: Cannot open vfstab file

Explanation

`/etc/vfstab` could not be opened.

Response

Make sure that `/etc/vfstab` exists. If not, restore it from a backup, or see [Section 20.2.3, "Setting `/etc/vfstab`"](#) and redefine it.

E.5.24.3 ERROR: sfcumntgl: 0003: Both *special* and *mount_point* need to be specified in vfstab

Explanation

It needs to be described by `/etc/vfstab` when both the special file *special* and mount point (*mount_point*) are specified as an operand.

Response

Check the entry of `/etc/vfstab`, and see [Section 20.2.3, "Setting `/etc/vfstab`"](#) and redefine it.

E.5.24.4 ERROR: sfcumntgl: 0004: {*special* | *mount_point*} not in vfstab

Explanation

It was not described by `/etc/vfstab` although {*special* | *mount_point*} was specified as a special file or a mount point.

Response

Check the status of `/etc/vfstab`.

E.5.24.5 ERROR: sfcumntgl: 0005: *device*: Cannot allocate memory

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.5.24.6 ERROR: sfcumntgl: 0006: {*special* | *mount_point*}: Cannot connect to cluster foundation

Explanation

Connection with the cluster control facility failed.

Response

Confirm that the cluster control facility has already been started.

E.5.24.7 ERROR: sfcumntgl: 0007: {*special* | *mount_point*}: Cannot get lock: config

Explanation

The lock to change the file system configuration information cannot be obtained. The lock to change the file system configuration information is now being obtained in an optional process of the node within the cluster configuration.

Response

Re-execute after releasing the lock for changing the domain configuration information.

E.5.24.8 ERROR: sfcumntgl: 0008: {*special* | *mount_point*}: Read error on management partition to get fsid

Explanation

Reading of the management partition to obtain the file system id failed.

Response

Confirm, using the likes of the **ps(1)** command, that sfcfrmd daemon is available.

E.5.24.9 ERROR: sfcumntgl: 0009: *device*: Cannot get lock: fsid

Explanation

Exclusive lock for fsid within the domain cannot be acquired. The exclusive lock for fsid has been acquired by another process running on a node in the cluster.

Response

Wait for the exclusive lock for fsid to be released, and then retry.

E.5.24.10 ERROR: sfcumntgl: 0010: {*special* | *mount_point*}: Read error on management partition to get node information

Explanation

Reading of the management partition to obtain the cluster configuration node information failed.

Response

Confirm, using the likes of the **ps(1)** command, that sfcfrmd daemon is available.

E.5.24.11 ERROR: sfcumntgl: 0011: {special | mount_point}: sfcfrmd may not exist

Explanation

A communication path to sfcfrmd daemon could not be established.

Response

Confirm, using the likes of the **ps(1)** command, that sfcfrmd daemon is available.

E.5.24.12 ERROR: sfcumntgl: 0012: hostname: {special | mount_point} busy

Explanation

Unmount processing for {special | mount_point} failed with *hostname*.

Response

In the node *hostname*, try to find a process that is using the file system by **fuser(1M)** command (for example, "**fuser -c hostname**"). If such a process exists, kill the process and unmount the file system again.

E.5.24.13 ERROR: sfcumntgl: 0013: The description form of vfstab is wrong

Explanation

The description format of */etc/vfstab* is incorrect.

Response

Confirm the contents of */etc/vfstab*.

E.5.24.14 ERROR: sfcumntgl: 0021: filename: Character device cannot be specified as mount point. Please specify block device

Explanation

Character-type special device name *filename* was specified as the mount point.

Response

Specifying a block-type special device name and then re-execute.

E.5.24.15 ERROR: sfcumntgl: 0022: special is not block device

Explanation

A block device was not specified as the argument of the **sfcumntgl(1M)** command.

Response

Specify a block device.

E.5.24.16 ERROR: sfcumntgl: 0029: {special | mount_point}: self-node is already processing mount or umount

Explanation

In the self-node, mount or umount processing has already been performed.

Response

Please set time a little and execute a command again.

E.5.24.17 ERROR: sfcumntgl: 0030: {*special* | *mount_point*}: mount_point need to be representative partition

Explanation

The special block device specified as a mount point was not a representation partition.

Response

Show a list of all the GFS Shared File System by **sfcinfo(1M)** command with -a option and find the representative partition. Again perform **sfcumntgl(1M)** command with the name of the representative partition as an argument.

E.5.24.18 WARNING: sfcumntgl: 0014: *device*: Cannot get lock: fsid, but processing is continued compulsorily

Explanation

Exclusive lock for fsid within the domain could not be acquired, but processing is performed forcibly.

Response

None.

E.5.24.19 WARNING: sfcumntgl: 0016: *hostname*: *special*: mount is in progress

Explanation

Mount processing had already been performed to *special* at *hostname*

Response

Either wait a short while and retry the unmount processing, or perform the unmount processing in forced mode.

E.5.24.20 WARNING: sfcumntgl: 0017: *hostname*: *special*: fail over is in progress

Explanation

Failover processing had already been performed to *special* at *hostname*

Response

Either wait a short while and retry the unmount processing, or perform the unmount processing in forced mode.

E.5.24.21 WARNING: sfcumntgl: 0018: *hostname*: *special*: umount is in progress

Explanation

Umount processing had already been performed to *special* at *hostname*

Response

None.

E.5.24.22 WARNING: sfcumntgl: 0026: *hostname*: {*special* | *mount_point*}: Send umount demand processing to frmd went wrong

Explanation

Transmission of the unmount processing request to sfcfrmd daemon failed with *hostname*.

Response

After confirming the state of sfcfrmd daemon with the likes of the **ps(1)** command, contact local Customer Support.

E.5.24.23 WARNING: sfcumntgl: 0027: {*special* | *mount_point*}: Waiting processing for a response umount demand went wrong

Explanation

Reception processing of the unmount processing request to sfcfrmd daemon failed.

Response

After confirming the state of sfcfrmd daemon with the likes of the **ps(1)** command, contact local Customer Support.

E.5.24.24 WARNING: sfcumntgl: 0028: {*special* | *mount_point*}: Time out occurred in umount demand processing to sfcfrmd

Explanation

A timeout occurred in the reception processing of the unmount processing request made to sfcfrmd daemon.

Response

After confirming the state of sfcfrmd daemon with the likes of the **ps(1)** command, contact local Customer Support.

E.6 General Command messages Specific to GFS Shared File System

This section explains the GFS Shared File System general command messages.

E.6.1 sfccp and sfcmv commands

E.6.1.1 *source*: failed to get acl entries

Explanation

Failed to acquire ACL information of the specified *source*.

Response

Check whether the specified *source* is correct.

E.6.1.2 *source*: unknown file type *mode*

Explanation

The type of copy source file is unknown.

Response

Contact local Customer Support.

E.6.1.3 {sfccp|sfcmv}: *directory/file*: Name too long

Explanation

The pathname of the copy source *directory/file* contains 1024 or more characters.

Response

Check whether the pathname of the arguments is correct. If incorrect, modify the pathname so that it consists of less than 1024 characters.

E.6.1.4 {sfccp|sfcmv}: *file*: *errmsg*

Explanation

File operation of the specified *file* failed.

Response

Retry after removing the cause of file operation failure by referring to *errmsg*.

E.6.1.5 {sfccp|sfcmv}: *file*: *fstatvfs*: *errmsg*

Explanation

The **statvfs(2)** system call issued to *file* relating to extent attribute information failed.

Response

Contact local Customer Support.

E.6.1.6 {sfccp|sfcmv}: *source* and *target* are identical

Explanation

The same file is specified as *source* and *target*.

Response

Check whether *source* and *target* are the intended pathnames.

E.6.1.7 {sfccp|sfcmv}: *target*: {gettext|setext}: *errmsg*

Explanation

Failed to {acquire|set} the extend attribute concerning *target*.

Response

Retry after removing the cause of failure to {acquire|set} the extend attribute with referring to *errmsg*.

E.6.1.8 {sfccp|sfcmv}: *target*: not a directory.

Explanation

target is not a directory.

Response

Check whether the specified *target* is correct. To copy or move a directory hierarchy, the directory must be specified as *target*.

E.6.1.9 {sfccp|sfcmv}: *target* not found

Explanation

Target cannot be found.

Response

Check whether the path to the copy destination directory *target* is correct.

E.6.1.10 {sfccp|sfcmv}: cannot access *source*

Explanation

A {stat(2)|lstat(2)} system call issued to the specified *source* failed.

Response

Check whether the specified *source* is correct. The search authority of the parent directory may not have been granted.

E.6.1.11 {sfccp|sfcmv}: cannot clear S_ISUID and S_ISGID bits in *target*: *errmsg*

Explanation

The set user ID or set group ID bits of *target* could not be cleared.

Response

Determine why the bits could not be cleared referring to *errmsg*. Clear the bits manually if necessary.

E.6.1.12 {sfccp|sfcmv}: cannot create *target*: *errmsg*

Explanation

Failed to create a specified *target*.

Response

Retry to create *file* after removing the cause of failure referring to *errmsg*.

E.6.1.13 {sfccp|sfcmv}: cannot open *source*: *errmsg*

Explanation

Failed to open the specified *source*.

Response

Check whether the specified *source* is correct referring to *errmsg*.

E.6.1.14 {sfccp|sfcmv}: cannot rmdir *target*: *errmsg*

Explanation

Failed to delete the directory *target*.

Response

Retry to delete the directory after removing the cause of failure referring to *errmsg*. Write authority for the parent directory of *target* may not have been granted.

E.6.1.15 {sfccp|sfcmv}: failed to set acl entries on *target*

Explanation

Failed to copy ACL information to *target* under the following conditions:

- The **-p** option is specified and
- An attempt was made to copy or move the information to a file system that does not support ACL.

Response

Take one of the following actions:

- Re-execute the command without specifying the **-p** option.
- Copy the information to a file system that supports ACL.

E.6.1.16 {sfccp|sfcmv}: Insufficient memory...

Explanation

Memory could not be allocated. Memory of system is insufficient.

Response

Wait until other processes terminate or request the system administrator to expand the swap area.

E.6.1.17 {sfccp|sfcmv}: Internal error!

Explanation

An internal inconsistency occurred in the command.

Response

Contact local Customer Support.

E.6.1.18 {sfccp|sfcmv}: Target *target* file name length exceeds MAXPATHLEN 1024

Explanation

The pathname of the copy destination directory *target* contains more than 1024 characters.

Response

Check whether the pathname of *target* is correct. If incorrect, modify so that the pathname does not exceed 1024 characters.

E.6.1.19 sfccp: *source* is a directory

Explanation

The specified *source* is a directory.

Response

To copy a directory hierarchy, use the **-r** or **-R** option.

E.6.1.20 sfccp: cannot create special file *target*: *errmsg*

Explanation

Failed to create a special file *target*.

Response

Retry to create a special file after removing the cause of failure referring to the reference manuals of *errmsg* and **mknod(2)**.

E.6.1.21 sfcmv: *source* is a directory

Explanation

The specified *source* is a directory.

Response

Check whether *source* is correct. A file other than the directory with the same name may exist at the destination.

E.6.1.22 sfcmv: cannot rename *source_file*: *errmsg*

Explanation

Failed to issue a **rename(2)** system call to *source* failed.

Response

Check whether the specified *source* is correct referring to the reference manuals of *errmsg* and **rename(2)**.

E.6.1.23 sfcmv: cannot unlink *file*: *errmsg*

Explanation

Failed to delete the file *file*.

Response

Retry to delete *file* after removing the cause of failure referring to *errmsg*. Write authority for the parent

directory of *file* may not have been granted.

E.6.2 sfccpio command

E.6.2.1 sfccpio: "*file*" ?

Explanation

Failed to open *file*.

Response

Retry to open *file* after removing the cause of failure.

E.6.2.2 sfccpio: "*file*" - checksum error

Explanation

A checksum error was detected while expanding *file*.

Response

Data in *file* may be incorrect. Check whether the archive is correct.

E.6.2.3 sfccpio: "*file*": failed to set acl, errno *num*, *errmsg*

Explanation

Failed to set ACL information in the copy destination *file*.

Response

File is copied while ACL information is missing. Check for any error in the created *file* referring to *errmsg*.

E.6.2.4 sfccpio: *file* : gid too large for archive format

Explanation

The group ID of *file* is too large to store information in this archive format.

Response

The correct group ID information cannot be saved. Change the group manually during expansion or use USTAR format to copy the information.

E.6.2.5 sfccpio: "*file*" has disappeared

Explanation

File has disappeared.

Response

Check for any problem that may be caused by the missing *file* in the archive.

E.6.2.6 sfccpio: "*file*" is not a directory

Explanation

File is not a directory.

Response

Specify a directory as the copy destination.

E.6.2.7 sfccpio: "*logname*" is not a valid user id

Explanation

Logname is not a valid user ID in */etc/passwd*.

Response

Specify a user ID that is registered in */etc/passwd*.

E.6.2.8 sfccpio: *file*: linkname *linkname* is greater than 100

Explanation

Since the link name *linkname* of the file *file* exceeds 100 characters, the file cannot be saved in the archive.

Response

Adjust so that the link name does not exceed 100 characters.

E.6.2.9 sfccpio: *file*: name too long.

Explanation

The file name of *file* is too long.

Response

Reduce the size of file name to less than 256 characters.

E.6.2.10 sfccpio: *file*: too large to archive in current mode

Explanation

file is too large to archive in this mode.

Response

To archive a file exceeding 2GB, use ASCII header format.

E.6.2.11 sfccpio: *file*: uid too large for archive format

Explanation

The user ID of *file* is too large to store information in this archive format.

Response

The correct user ID information cannot be saved. Change the owner manually during expansion or use USTAR format.

E.6.2.12 sfccpio: acl count error

Explanation

Failed to read ACL information.

Response

Check whether sufficient resources are available.

E.6.2.13 sfccpio: aclfromtext failed

Explanation

Failed to read ACL information.

Response

Check whether sufficient resources are available.

E.6.2.14 sfccpio: Attempt to pass a file to itself.

Explanation

An attempt was made to copy a file to the same file.

Response

Check whether the pathname for copying is correct.

E.6.2.15 sfccpio: Bad header - checksum error.

Explanation

The header is incorrect. A checksum error was detected.

Response

Check whether the archive contents are correct.

E.6.2.16 sfccpio: Bad magic number/header.

Explanation

A header with an incorrect magic number was detected.

Response

Use another archiver (such as **cpio(1)** and **tar(1)**) for expansion. Otherwise, specify the **-k** option for expansion.

E.6.2.17 sfccpio: can create security information

Explanation

Failed to create security information.

Response

Retry to create the information after removing the cause of failure referring to the previous error message displayed.

E.6.2.18 sfccpio: {Cannot|Unable to} append to this archive, errno *num*, *errmsg*

Explanation

Failed to add to this archive.

Response

Retry to add the archive after removing the cause of failure referring to *errmsg*. The arguments may be specified incorrectly.

E.6.2.19 sfccpio: Cannot change media types in mid-stream.

Explanation

The media type cannot be changed during operation.

Resolution

Retry the media change using the same media type.

E.6.2.20 sfccpio: Cannot {chmod()}|chown()} "*file*", *errno num*, *errmsg*

Explanation

A {**chmod(2)**|**chown(2)**} system call to *file* failed.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.21 sfccpio: Cannot create "*file*", *errno num*, *errmsg*

Explanation

Failed to create *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*. Write authority for the parent directory of *file* may not have been granted.

E.6.2.22 sfccpio: {Cannot|Unable to} create directory for "*file*", *errno num*, *errmsg*

Explanation

A directory for the copy destination file *file* cannot be created.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*. Write authority for the parent directory of *file* may not have been granted.

E.6.2.23 sfccpio: Cannot create extent information file

Explanation

Failed to create extent information of *file*.

Response

Re-execute the command after removing the cause of failure referring to the previous error message displayed.

E.6.2.24 sfccpio: Cannot create temporary file, *errno num*, *errmsg*

Explanation

Failed to create a temporary file.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.25 sfccpio: Cannot follow "*file*", errno *num*, *errmsg*

Explanation

No link destination of the symbolic file *file* can be found.

Response

Check whether the link destination of *file* is correct referring to *errmsg*.

E.6.2.26 sfccpio: Cannot fstatvfs "*file*", errno *num*, *errmsg*

Explanation

Failed to issue a **fstatvfs(2)** system call to the saved file.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.27 sfccpio: Cannot getext "*file*", errno *num*, *errmsg*

Explanation

Failed to acquire extent information of *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.28 sfccpio: Cannot get temporary file name.

Explanation

Failed to create a temporary file.

Response

Check whether sufficient free space is available for the file system.

E.6.2.29 sfccpio: Cannot link "*name1*" and "*name2*", errno *num*, *errmsg*

Explanation

Failed to link *name1* and *name2*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.30 sfccpio: Cannot lseek(), errno *num*, *errmsg*

Explanation

Failed to seek the media.

Response

Retry to seek the media after removing the cause of failure referring to *errmsg*.

E.6.2.31 sfccpio: Cannot mknod() "*file*", errno *num*, *errmsg*

Explanation

Failed to create a special file *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.32 sfccpio: Cannot open "*string*", errno *num*, *errmsg*

Explanation

Failed to open the file *string*.

Response

Check whether the specified *string* is correct referring to *errmsg*.

E.6.2.33 sfccpio: Cannot open "*file*" for append

Explanation

Failed to open *file* for addition.

Response

Check whether *file* is specified correctly.

E.6.2.34 sfccpio: Cannot open "*file*" for input

Explanation

Failed to open the pattern file *file*.

Response

Check whether *file* is specified correctly.

E.6.2.35 sfccpio: Cannot open "*file*" for output

Explanation

Failed to open *file* for output.

Response

Check whether *file* is specified correctly.

E.6.2.36 sfccpio: Cannot open "*file*", skipped, errno *num*, *errmsg*

Explanation

file cannot be opened and will be skipped.

Response

Retry to open the file after removing the cause of failure referring to *errmsg*.

E.6.2.37 sfccpio: Cannot open "*file*" to read patterns

Explanation

Failed to open the pattern file *file*.

Response

Check whether *file* is specified correctly.

E.6.2.38 sfccpio: Cannot overwrite "*file*"

Explanation

file cannot be overwritten. Only a super user can overwrite a special file.

Response

If *file* must be overwritten, re-execute the command as a super user.

E.6.2.39 sfccpio: Cannot read "*file*", errno *num*, *errmsg*

Explanation

Failed to read *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.40 sfccpio: Cannot read symbolic link "*file*", errno *num*, *errmsg*

Explanation

Failed to read symbolic link information of *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.41 sfccpio: Cannot read tty.

Explanation

Failed to read input from the terminal.

Response

Check whether the terminal used for input is correct. Also check whether standard input is disabled.

E.6.2.42 sfccpio: Cannot recover.

Explanation

Failed to preserve sections that are not damaged.

Response

The correct header cannot be found even if the damaged section is skipped. No file can be read from this media.

E.6.2.43 sfccpio: Cannot recover original version of "*file*", *errno num*, *errmsg*

Explanation

Failed to restore the original version of *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*. *file* may have been lost.

E.6.2.44 sfccpio: Cannot rename, "*ttyname*" missing

Explanation

Failed to open the terminal.

Response

Check whether the terminal used for execution is correct. Also check whether standard input or standard output is disabled.

E.6.2.45 sfccpio: Cannot remove incomplete "*file*", *errno num*, *errmsg*

Explanation

Failed to delete the file being transferred when copying was interrupted.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*. Otherwise, delete *file* manually.

E.6.2.46 sfccpio: Cannot remove temp file "*file*", *errno num*, *errmsg*

Explanation

Failed to delete the temporary file *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*. Otherwise, delete *file* manually.

E.6.2.47 sfccpio: Cannot remove the directory "*file*", *errno num*, *errmsg*

Explanation

Failed to delete the directory *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*. Write authority for the parent directory of *file* may not have been granted.

E.6.2.48 sfccpio: Cannot reset file after checksum, *errno num*, *errmsg*

Explanation

Failed to reset the file after checksum calculation.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.49 sfccpio: Cannot setext "*file*", errno *num*, *errmsg*

Explanation

Failed to set extent information to *file*.

Response

Re-execute the command after removing the cause of failure referring to *errmsg*. Assume the following causes:

- Free space in the file system at the expansion destination is insufficient.
 - An attempt was made to create a file exceeding the file size limit (ulimit value).
- *file* is neither in the GFS Shared File System nor the GFS Local File System.

E.6.2.50 sfccpio: Cannot swap bytes of "*file*", odd number of bytes

Explanation

Failed to swap the byte order of *file*. The file size is odd-numbered

Response

The file size is not a multiple of 2. Re-execute the command after setting an even number of bytes.

E.6.2.51 sfccpio: Cannot swap halfwords of "*file*", odd number of halfwords

Explanation

Half-word swap of *file* failed. The file size is an odd-numbered multiple of the half word.

Action

The file size is not a multiple of 4. Re-execute the command after setting a multiple of 4.

E.6.2.52 sfccpio: Cannot unlink() {current|temp file} "*file*", errno *num*, *errmsg*

Explanation

Failed to delete *file*.

Action

Retry to delete the file after removing the cause of failure referring to *errmsg*. Write authority for the parent directory of *file* may not have been granted.

E.6.2.53 sfccpio: Cannot write "*file*", errno *num*, *errmsg*

Explanation

Failed to write to *file*.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.54 sfccpio: Can't read input: end of file encountered before expected end of archive.

Explanation

Failed to read data from the input. The end of file was detected before the end of archive.

Action

Check whether the media is correct. If correct, contents of the archive may be invalid. Recreate the archive if possible.

E.6.2.55 sfccpio: close error, errno *num*, *errmsg*

Explanation

Failed to close the output file or archive file.

Action

Write operation to the output file or archive file may have failed. Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.56 sfccpio: Corrupt header, file(s) may be lost.

Explanation

The header is damaged. The file may have been lost.

Action

Check whether the archive contents are correct. Even if the archive is damaged, it may be recoverable by using the **-k** option.

E.6.2.57 sfccpio: Error cannot unlink "*file*", errno *num*, *errmsg*

Explanation

File could not be deleted.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*. Write authority for the parent directory of *file* may not have been granted.

E.6.2.58 sfccpio: Error computing checksum.

Explanation

An error was detected during checksum calculation.

Action

Refer to the action given in the next error message.

E.6.2.59 sfccpio: Error during access() of "*file*", errno *num*, *errmsg*

Explanation

An **access(2)** system call to *file* failed.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.60 sfccpio: Error during chown() of "*file*", errno *num*, *errmsg*

Explanation

Failed to change the owner of *file*.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.61 sfccpio: Error during initialization, errno *num*, *errmsg*

Explanation

An error occurred during device initialization.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.62 sfccpio: Error during stat() of "*file*", errno *num*, *errmsg*

Explanation

A **stat(2)** system call to *file* failed.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.63 sfccpio: Error during stat() of archive, errno *num*, *errmsg*

Explanation

A **stat(2)** system call to an archive failed.

Action

Check whether the archive file is correct by referring to *errmsg*.

E.6.2.64 sfccpio: Error during stat(*file*) of file, errno *num*, *errmsg*

Explanation

A **stat(2)** system call to *file* failed.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.65 sfccpio: Error with acl() of "*file*", errno *num*, *errmsg*

Explanation

Failed to acquire the number of ACL of *file*.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.66 sfccpio: Error with getacl() of "*file*", errno *num*, *errmsg*

Explanation

Failed to acquire ACL information of *file*.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.67 sfccpio: Error with lstat() of "*file*", *errno num*, *errmsg*

Explanation

A `lstat(2)` system call to *file* failed.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.68 sfccpio: Error with malloc() of "*file*", *errno num*, *errmsg*

Explanation

Memory for *file* could not be allocated. Memory of system is insufficient.

Action

Please check the state of a system, and wait for the termination of another process, or request the system administrator to expand the swap area.

E.6.2.69 sfccpio: Existing "*file*" same age or newer

Explanation

The existing *file* was updated more recently than the one being expanded.

Action

Determine why *file* is more recent. If it must be overwritten, re-execute the command using the `-u` option.

E.6.2.70 sfccpio: Impossible header type.

Explanation

An invalid header type was detected.

Action

The specified archive was not created with `sfccpio(1)` or the header is damaged. Check the contents of the archive.

E.6.2.71 sfccpio: I/O error - cannot continue, *errno num*, *errmsg*

Explanation

I/O error occurred. Processing cannot be continued.

Action

Re-execute the command after removing the cause of I/O error referring to *errmsg*.

E.6.2.72 sfccpio: I/O error on "{output|input}", *errno num*, *errmsg*

Explanation

I/O error occurred at the {output destination|input source}.

Action

Re-execute the command after removing the cause of I/O error referring to *errmsg*. Depending on the situation, the media must be replaced.

E.6.2.73 sfccpio: Name exceeds maximum length - skipped.

Explanation

Since the name exceeds the maximum length, it was skipped.

Action

Reduce the size of the name to 256 characters or less.

E.6.2.74 sfccpio: No space left for output file.

Explanation

No free space to store the output file can be found.

Action

Retry after increasing free space at the destination.

E.6.2.75 sfccpio: No sfcs file system. Extent information in *file* was lost.

Explanation

Since the recovery destination is not on the GFS Shared File System, extent attribute information could not be recovered. Data, however, can be recovered.

Action

Check for any error in the recovery destination information.

E.6.2.76 sfccpio: Not a sfccpio file, bad header.

Explanation

The archive format is incorrect.

Action

Use the same header format or archiver (such as **cpio(1)** and **tar(1)**) as that used for creating the archive for expansion.

E.6.2.77 sfccpio: Out of memory.

Explanation

Memory could not be allocated. Memory of system is insufficient.

Action

Please check the state of a system, and wait for the termination of another process, or increase the swap space or real memory. After it and re-execute the command.

E.6.2.78 sfccpio: out of memory for acl

Explanation

Memory for acl could not be allocated. Memory of system is insufficient.

Action

Please check the state of a system, and wait for the termination of another process, or increase the

swap space or real memory. After it and re-execute the command.

E.6.2.79 sfccpio: remove_dir() failed to chdir("path") , errno num, errmsg

Explanation

Failed to move to the directory *path*.

Action

Refer to the action given in the next error message.

E.6.2.80 sfccpio: remove_dir() failed to chdir("..") , errno num, errmsg

Explanation

Failed to move to the parent directory.

Action

Refer to the action given in the next error message.

E.6.2.81 sfccpio: remove_dir() failed to opendir("path") , errno num, errmsg

Explanation

Failed to open the directory *path*.

Action

Refer to the action given in the next error message.

E.6.2.82 sfccpio: remove_dir() failed to remove_dir("name") , errno num, errmsg

Explanation

Failed to delete files under the directory *name*.

Action

Refer to the action given in the next error message.

E.6.2.83 sfccpio: remove_dir() failed to rmdir("path") , errno num, errmsg

Explanation

Failed to delete the directory *path*.

Action

Refer to the action given in the next error message.

E.6.2.84 sfccpio: remove_dir() failed to stat("name") , errno num, errmsg

Explanation

A **stat(2)** system call to the file *name* failed.

Action

Refer to the action given in the next error message.

E.6.2.85 sfccpio: remove_dir() failed to unlink("name") , errno num, errmsg

Explanation

Failed to delete the file *name*.

Action

Refer to the action given in the next error message.

E.6.2.86 sfccpio: Searching for magic number/header.

Explanation

A header with the correct magic header is being searched.

Action

There are no need for action. Check the next error message.

E.6.2.87 sfccpio: Skipping "file": exceeds ulimit by num bytes

Explanation

file will be skipped. The upper limit of file size is exceeded by *num* bytes.

Action

Re-execute the command after increasing the upper limit of file size.

E.6.2.88 sfccpio: Target is not a file on sfafs file system. Extent information in file was lost.

Explanation

Since the restore destination is not on the GFS Shared File System, extent attribute information could not be recovered. Data, however, can be recovered.

Action

Check for any error in the destination information.

E.6.2.89 sfccpio: ulimit reached for output file.

Explanation

The upper limit of the output file size is reached.

Action

Re-execute the command after increasing the upper limit of file size.

E.6.2.90 sfccpio: Unable to determine current directory.

Explanation

Failed to acquire the current directory.

Action

Check whether the current directory is enabled.

E.6.2.91 sfccpio: Unable to reset access time for "*file*", *errno num*, *errmsg*

Explanation

Failed to set the reference time to *file*.

Action

If necessary, re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.92 sfccpio: Unable to reset modification time for "*file*", *errno num*, *errmsg*

Explanation

Failed to set the update time to *file*.

Action

If necessary, re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.2.93 sfccpio: Unable to write this medium, try another.

Explanation

Failed to write to the media.

Action

Re-execute the command after replacing the media. The media may be write-protected.

E.6.2.94 sfccpio: Unexpected end-of-file encountered.

Explanation

An unexpected end-of-file was detected.

Action

Check whether the media is correct.

E.6.3 sfcgetext command

E.6.3.1 sfcgetext: *file* is not a file on sfcfs file system

Explanation

The specified *file* is not a file on the GFS Shared File System.

Action

Specify a file on the GFS Shared File System.

E.6.3.2 sfcgetext: *file* is not a regular file

Explanation

The specified *file* is not an ordinary file.

Action

Specify an ordinary file.

E.6.3.3 sfcgetext: cannot fstatvfs *file*: *errmsg*

Explanation

Failed to issue an **fstatvfs(2)** system call to the specified *file* failed.

Action

Re-execute the command after removing the cause of **fstatvfs(2)** system call failure referring to *errmsg*.

E.6.3.4 sfcgetext: cannot getext *file*: *errmsg*

Explanation

Failed to acquire extent information of the specified *file*.

Action

Re-execute the command after removing the cause of failure referring to *errmsg*.

E.6.3.5 sfcgetext: cannot open *file*: *errmsg*

Explanation

Failed to open the specified *file*.

Action

Check whether the specified *file* is correct by referring to *errmsg*.

E.6.3.6 sfcgetext: cannot stat *file*: *errmsg*

Explanation

A **stat(2)** system call to the specified *file* failed.

Action

Check whether the specified *file* is correct by referring to *errmsg*.

E.6.4 sfcsetext command

E.6.4.1 sfcsetext: *file* is not a file on sfcfs file system

Explanation

The specified *file* is not on the GFS Shared File System.

Response

Create a file on the GFS Shared File System.

E.6.4.2 sfcsetext: cannot create *file*: *errmsg*

Explanation

Failed to create the specified *file*.

Response

Re-execute the command after removing the cause of failure to create *file* referring to *errmsg*.

E.6.4.3 sfcsetext: cannot fstatvfs *file*: *errmsg*

Explanation

Failed to issue an **fstatvfs(2)** system call to the specified *file*.

Response

Re-execute the command after removing the cause of failure of the **fstatvfs(2)** system call to *file* referring to *errmsg*.

E.6.4.4 sfcsetext: cannot sfcsetext: *errmsg*

Explanation

Failed to set extent information of the specified *file*.

Response

Re-execute the command after removing the cause of failure by referring to *errmsg*. Failure may have resulted from either of the following causes:

If contig is not specified:

- The area for init-blocks could not be acquired.

In this case, re-execute the command after deleting unnecessary files, specifying another GFS Shared File System, or adding a data area.

If contig is specified:

- A contiguous area for init-blocks could not be acquired (if inc-blocks are 0).
- A contiguous area for init-blocks could not be acquired in inc-blocks.
- When init-blocks are not a multiple of inc-blocks, a contiguous area for the fraction could not be acquired.

In these cases, take the following actions:

- If 0 is set to inc-blocks, examine the value of init-blocks.
- If any value other than 0 is set to inc-blocks, decrease the value specified to inc-blocks. If the area still cannot be acquired, re-execute the command after deleting unnecessary files, specifying GFS Shared File System, or adding a data area.

E.6.4.5 sfcsetext: File exists: *file*

Explanation

The specified *file* already exists.

Response

Check whether the specified *file* is correct. To recreate *file*, delete it before retrying.

E.7 GFS Shared File System Management View Messages

This appendix provides information on message dialog associated with GFS Shared File System Management View, and wherever possible, a recovery procedure is provided to locate and correct the problem.

E.7.1 Error messages

E.7.1.1 6000: Executing the command was unsuccessful.

Explanation

Executing the command was unsuccessful.

Response

To proceed with operation, press <Yes> button. To cancel press <No> button.

E.7.1.2 6001: Command line output: (detailed information)

Explanation

The error given in (detailed information) occurred when executing the command.

Response

Identify the cause referring to the command line output. When you cannot identify the cause, contact local Customer Support.

E.7.1.3 6002: Abnormality occurred in the log monitoring mechanism on node *node*. Monitoring will stop. Restart Global File Services.

Explanation

Abnormality occurred in the log monitoring mechanism on node *node*.

Response

Restart GFS management.

E.7.1.4 6003: Specified meta partition is not found at server side.

Explanation

The specified meta partition could not be found.

Response

Specify the name of a meta partition that can be shared by the specified hosts, and re-execute the command.

E.7.1.5 6004: Specified data partition is not found at server side.

Explanation

The specified data partition could not be found.

Response

Specify the name of a data partition that can be shared by the specified hosts, and re-execute the command.

E.7.1.6 6005: Specified log partition is not found at server side.

Explanation

The specified log partition could not be found.

Response

Specify the name of a log partition that can be shared by the specified hosts, and re-execute the command.

E.7.1.7 6006: Specified File system is not found at server side.

Explanation

The specified file system could not be found.

Response

Specify the name of a file system that can be shared by the specified hosts, and re-execute the command.

E.7.1.8 6007: Specified file system still exists in /etc/vfstab.

Explanation

The specified file system was registered in the `/etc/vfstab`.

Response

Check whether the specified file system is correct.

E.7.2 Warning messages

E.7.2.1 6200: The file system is active. Cannot change the status.

Explanation

The active file system cannot delete or change.

Response

Unmount the file system, and do again.

E.7.2.2 6201: Specified file system has error status, so can not modify status.

Explanation

Specified file system has error status.

Response

Investigate the cause of abnormalities by reference of a Log.

E.7.2.3 6202: There is no enough update log area in the specified configuration. Please do reconfiguration.

Explanation

The update log area is too small.

Response

Change partition configuration to get enough update log area, and press <Next> or <Create> button.

E.7.2.4 6203: The input mount point is not available. Change the mount point.

Explanation

The input mount point is not available.

Response

Change the mount point, and press <Next> button.

E.7.2.5 6204: There is no disk shared by the specified node. Please change the selection of the node.

Explanation

There is no disk shared by the specified node.

Response

Please change the selection of the node.

E.7.2.6 6205: There is no disk shared by the specified node. Please change the selection of the node.

Explanation

There is no disk shared by the specified node.

Response

Please change the selection of the node.

E.7.2.7 6206: There are no unused disks. so You can not create new file system.

Explanation

There are no unused disks.

Response

Nothing.

E.7.2.8 Specified file system will be deleted. Data in the file system is lost. Are you sure?

Explanation

When the file system is deleted, data in the file system is lost.

Response

To delete the file system, press the <Yes> button. To cancel the deletion of the file system, press the <No> button.

E.7.2.9 Do you want to quit file system creation?

Explanation

Quit file system creation.

Response

To quit, press the <Yes> button; otherwise, press the <No> button.

E.7.3 Information messages

E.7.3.1 A file system has been created.

Explanation

A file system was created.

Response

Press the <OK> button to close the message.
Start the cluster service to make available the file system.

E.7.3.2 Changing configuration was completed.

Explanation

Changing configuration of the file system was completed.

Response

Press the <OK> button to close the message.

E.7.3.3 Changing attribution was completed.

Explanation

Changing attribution of the file system was completed.

Response

Press the <OK> button to close the message.

E.7.3.4 A file system has been deleted.

Explanation

A file system was deleted.

Response

Press the <OK> button to close the message.

E.7.3.5 Exit GFS management?

Explanation

Exits GFS management.

Response

Press <Yes> to exit, <No> to continue operation.

E.8 Web-Based Admin View Messages

For information about the Web-Based Admin View error messages displayed when GFS Shared File System is active, see "*PRIMECLUSTER Web-Based Admin View Operation Guide*."

E.9 Log Reference Messages

For information about the Log Reference Messages, see "*PRIMECLUSTER Cluster Foundation Configuration and Administration*" or "*PRIMECLUSTER Installation Guide for Solaris*."

E.10 Operating Environment Setting Messages

For information about the error messages output during operating environment setting, see "*PRIMECLUSTER Installation Guide for Solaris.*"

Appendix F Reference Manual

This appendix is intended for use as a reference manual for the GFS Shared File System.

F.1 Commands for Common File System Management

F.1.1 df_sfdfs(1M) Reports the free disk area of GFS Shared File System

SYNOPSIS

```
df -F sfdfs [ generic_options ] [ -o specific_options ] [ directory | special ]
```

DESCRIPTION

The **df** displays the number of free disk blocks and the number of free i-nodes of a GFS Shared File System.

Note that the area reported as being in use and available for use is smaller than the file system area. This is because the system reserves a part of the file system and uses it for processing a file system allocation routine. The reserve area is generally **10** percent. This value can be adjusted with [sfctune\(1M\)](#). If all file system areas other than the reserve area is in use, only a super user can allocate data blocks for new and existing files. If such over-allocation of the file system occurs, the **df** display indicates that more than **100** percent of the file system is being used.

Either a special file name or a mount point name can be specified. If neither is specified, all mounted file systems are displayed.

If a file system has configured with multiple partitions with a multi-partition function, specify the path name of the representative partition as a special file name.

If a GFS Shared File System is configured with multi-partitions, information for each partition can be displayed.

For meta-data area, the following information is displayed:

- i-node use status (i-node count, free i-node count)
- Free extent status (leaf count, free leaf count)
- V-data use status (block count, free block count)

For file data area, the following information is displayed:

- Free file data area size

For update log area, the following information is displayed:

- Update log area size

OPTIONS

The following options can be specified:

generic_options

generic_options is supported by the general-purpose **df(1M)**.

See the options of **df(1M)**.

-a

Outputs information even for a file system whose */etc/mnttab* entry contains an ignore option set.

-b

Outputs the capacity available for use in kilobytes.

-e

Outputs only the number of files available for use.

-g

Outputs statvfs structures.

-k

Outputs the allocation state in kilobytes.

-l

Outputs only local file systems.

- n**
Outputs only the file system type.
- t**
Outputs a complete list, including a total.
- V**
Displays a command line, but does not execute the command.

-o *specific_options*

Allows options specific to GFS Shared File System to be specified. Insert a comma delimiter between options.

- i**
Reports the counts for i-nodes in use and free i-nodes.
- v**
Displays information about each partition if the file system is configured with multi-partitions.

OPERANDS

The following operands can be specified:

directory

Specifies path name of a directory for the GFS Shared File System.

special

Specifies path name of a block-special file for the representative partition of the GFS Shared File System.

EXAMPLES

The following example writes information about the file system for single-partition:

```
# df -F sfdfs -o v /mnt
Filesystem:/mnt
      inodes  free  vdata free  leaf  free  Type
/dev/sfdsk/gfs/dsk/volume1
      14248 12192  4781 4615  7128  7127  META
      kbytes  used  free %used
/dev/sfdsk/gfs/dsk/volume1
      113992 62664 51328 55%          DATA
      kbytes
/dev/sfdsk/gfs/dsk/volume1
      5120          LOG
```

F.1 Commands for Common File System Management

The following example writes information about the file system for multi-partition:

```
# df -F sfcfs -o v /mnt
Filesystem:/mnt
      inodes  free vdata free leaf free  Type
/dev/sfdsk/gfs/dsk/volume1
      29686 27630 9922 9756 14848 14846 META
      kbytes used free %used
/dev/sfdsk/gfs/dsk/volume1
      105752 31832 73920 30%          DATA
      kbytes
/dev/sfdsk/gfs/dsk/volume2
      5120          LOG
      kbytes used free %used
/dev/sfdsk/gfs/dsk/volume3
      131736 30832 100904 23%          DATA
      inodes free vdata free leaf free
/dev/sfdsk/gfs/dsk/volume4
      32768 32714 32768 2819 0 0 META
      kbytes
/dev/sfdsk/gfs/dsk/volume4
      38502          FREE
```

SEE ALSO

[mkfs_sfcfs\(1M\)](#), [sfcnewfs\(1M\)](#), [sfc tune\(1M\)](#).

[df\(1M\)](#), [statvfs\(2\)](#), [mnttab\(4\)](#), [attributes\(5\)](#) in the "Solaris X Reference Manual Collection."

NOTE

- **df_sfcfs** cannot display accurate values while a partition is added. Use [sfcinfo\(1M\)](#) to confirm whether the partition is being processed, then re-execute **df_sfcfs** following the addition of the partition.

F.1.2 `fsck_sfdfs(1M)` Checks the GFS Shared File System for consistency and recovers the file system

SYNOPSIS

`fsck -F sfdfs [generic_options] [-o specific_options] [special ...]`

DESCRIPTION

The `fsck` command checks the GFS Shared File System for consistency and restores the file system.

High-speed recovery of the GFS Shared File System can be performed by using the update log function. Specify whether to perform high-speed recovery by executing an update log replay when `fsck` is executed. If an update log replay cannot be executed because of a damaged update log or for some other reason, `fsck` executes a full file system check. To specify a file system to be checked, use a character-special device or specify a mount point for the file system that has existing entries in `/etc/vfstab`.

For *special*, specify the character-special device or the mount point where the file system has been built. A block-special device should not be specified because `fsck` is not executed if a block device is mounted.

Unless *special* is specified, `fsck` checks all GFS Shared File Systems using the `fsckdev` entry in `/etc/vfstab`. When the `p` (`prin`) option is specified, `fsck` checks GFS Shared File Systems with `fsckpass` being 1 or above.

If there is a serious inconsistency, the operator is prompted to reply **yes** or **no** before `fsck` executes recovery. If the operator does not have write permission for the file system, `fsck` executes the operation with the `-n` option (no corrections are made).

`fsck` executes automatic recovery for minor inconsistencies such as:

- i-node not referenced
- Too many links in an i-node
- An incorrect block included in a free list
- A block appearing in both a free list and a file
- An invalid value detected in a super block

When the inconsistency has been corrected, a message is displayed. When recovery of the file system is successful, `fsck` displays the number of files on the file system, the counts for blocks in use and free blocks, and the fragmentation rate.

A full check is executed for the following inconsistencies:

- A block is acquired by multiple i-nodes or free lists.
- A block is requested by an i-node or free list outside the range of the file system.
- Invalid link count
- Invalid directory size
- i-node with invalid syntax
- A block not referenced from anywhere
- Directory check, a file pointing to an unallocated i-node, i-node number falling outside the range, the first two entries (`.` and `..`) in a directory
- Super block check: i-node blocks exceed the number of blocks in the file system.
- A free block list with invalid syntax
- An invalid total for the free block count and free i-node count

Independent files and directories (allocated, but not referenced) are placed in a lost+found directory and reunited with the operator's consent. They are assigned i-node numbers as names. If a lost+found directory does not exist, create it. If area for the lost+found directory is not sufficient, expand the directory.

If a file system is configured with multiple partitions, specify the path name of the representative partition as a special file name.

OPTIONS

The *generic-options* consist of the following options:

-v

Displays a command line, but does not execute the command.

-m

Executes a check, but does not make corrections.

-n|N

Automatically replies "no" to all inquiries. Does not open the file system for writing.

-y|Y

Automatically replies **yes** to all inquiries.

-o specific_options

Allows options specific to the GFS Shared File System to be specified. Insert a comma delimiter between options.

f

Executes a forcible check, regardless of the super block status flag.

p

Checks for non-conversational synopsis and makes corrections.

w

Checks only for write-enabled file systems.

log|elog|nolog

Specifies whether to perform an update log replay.

The default is **log**.

log

Executes an update log replay. Executes a full file system check if the log data is physically destroyed.

elog

Executes an update log replay. The command terminates immediately if the log data is physically destroyed without a full file system check being executed.

nolog

Executes a full file system check without executing an update log replay.

work=raw_device

The work disk at the time of **fsck** execution is specified. Two or more work disks can be specified for specification of **work=raw_device** and by enumerating by ".". A work disk can be specified to 16 partitions.

OPERAND

The following operand can be specified:

special

Specifies a character-special file for the representative partition of the GFS Shared File System or path name of the mount-point.

EXAMPLES

```
# fsck -F sfcfs /dev/sfdsk/gfs/rdsk/volume1
# fsck -F sfcfs -o nolog /dev/sfdsk/gfs/rdsk/volume1
```

EXIT STATUS

- 0 File system recovery has been completed normally, or no recovery was necessary.
- 1 An incorrect parameter was specified.
- 32 The file system is unmounted in all nodes, and a check is required (only when **fsck -m** is specified).
- 33 The file system is mounted.
- 34 **stat(2)** for the device failed.
- 36 The processing has terminated because a serious problem requiring operator intervention has occurred during the execution of the command with the **p** (prin) option specified.
- 37 An interrupt has occurred during command execution.
- 39 The processing has terminated because of an abnormal event occurring during command execution.
- 50 Although command execution with the **elog** option specified was attempted, an update log replay could not be executed the update log was physically damaged.

SEE ALSO

[mkfs_sfcfs\(1M\)](#), [sfcnews\(1M\)](#).

crash(1M), **fsck(1M)**, **fstyp(1M)**, **mkfs(1M)**, **reboot(1M)**, **stat(2)**, **vfstab(4)**, **attributes(5)**, and **largefile(5)** in the "*Solaris X Reference Manual Collection*."

NOTES

- When the calculated size using the following formula is 512 megabytes or more, work disks for the full file system check are needed.
40MB + [number of data partitions] x 32MB + [file system size (gigabyte unit)] x 65MB
- A file system can be checked only when all hosts that share the file system have started operation. The reason is that the information for these hosts is required.

F.1.3 `fstyp_sfdfs(1M)` Determines the type of a file system

SYNOPSIS

`fstyp` [`-v`] *special*

DESCRIPTION

The `fstyp` command determines the type of the file system that has been specified.

`fstyp` executes modules for checking different file system types.

When a module matches the GFS Shared File System, the `fstyp` command displays file system name "sfdfs" as standard output and terminates with a termination status of 0.

If all modules are unsuccessful, the command displays the error message `unknown_fstyp` (no matches) and terminates with a termination status of 1.

If multiple modules are successful, the command displays an error message of `unknown_fstyp` (multiple matches) and terminates with the termination status of 2.

If a file system is configured with multiple partitions, specify the path name of a representative partition as a special file name.

OPTION

The following option can be specified:

`-v`

Outputs detailed information.

Displays information about super block and partition information.

OPERAND

The following operand can be specified:

special

Specifies path name of the special file for the target partition.

EXAMPLE

```
# fstyp /dev/sfdsk/gfs/dsk/volume1
sfdfs
```

SEE ALSO

[mkfs_sfdfs\(1M\)](#), [sfctune\(1M\)](#).

`fstyp(1M)`, `attributes(5)`, `largefile(5)` in the "Solaris X Reference Manual Collection."

NOTE

- `fstyp_sfdfs` may print incorrect detailed information while a partition is being added. Use [sfinfo\(1M\)](#) to confirm whether the partition is being processed. If so, re-execute [sfadd\(1M\)](#), then execute `fstyp(1M)`.

F.1.4 mkfs_sfcfs(1M) Constructs the GFS Shared File System

SYNOPSIS

```
mkfs -F sfcfs [ generic_options ] [ -o specific_options ] raw_device_file
```

DESCRIPTION

The **mkfs_sfcfs** command builds the GFS Shared File System on the disks that were specified.

Once the GFS Shared File System has been created, its entry is added to file system configure information in a management partition. Thereafter, file system configure information in a management partition is used to manage the GFS Shared File System.

OPTIONS

generic_options

generic_options is supported by a general-purpose **mkfs(1M)**.

See the options of **mkfs(1M)** for details.

- o *specific_options*

Allows options specific to GFS Shared File System to be specified. Insert a comma delimiter between options.

N

Displays parameters used for creating a file system. Does not actually create a file system.

clrmeta

A meta-data area is initialized altogether.

data=raw_device_file

Specifies the character-special devices for a file data partition. If a file system is built with multiple partitions that are used as file data partitions, specify multiple entries for this parameter.

dataopt=y|n

Specifies whether a representative partition includes a file data area if file data partitions are specified with the **data** sub-option. If **y** is specified, the representative partition does not include a file data area. If **n** is specified, the representative partition includes a file data area.

The default is **n**.

If no file data partitions are specified, **n** is set, regardless of whether this parameter is specified.

free=n

Specifies the minimum percentage of free area in the file system.

The default is **10%**.

A value can be specified in the range from **0** to **99**.

force

Specify the **force** option to re-create a file system previously registered in file system configure information in a management partition when its configuration is unchanged. If, however, the target file system is being mounted, specifying this option results in an error.

log=raw_device_file

Specifies a character-special device for an update log partition.

The portion of an update log partition not used, as an update log area becomes an unused area. Specify a partition that is the same size as the update log area.

logsz=n

Specifies the size of the update log area. The unit is megabytes.

The default is about **1%** of the file system size. When **maxdsz** has been specified, the default is about **1%** of the **maxdsz** value. If the default value is smaller than **5** megabytes, **5** megabytes is set. If the default value is greater than **50** megabytes, **50** megabytes is set.

A value can be specified in the range from **5** to **100**.

maxdsz=*n*

Specifies the maximum size of the file data areas of each file system that includes additional file data partitions. The unit is megabytes. Based on the value specified for this parameter, the size of meta-data and the update log area size are set to proper values after taking into account the additional file data partitions.

If this option is not specified, the sizes of meta-data area and the update log area are determined based on the file system size of the specified partition.

A value greater than **32** terabyte (**33,554,432** megabytes) cannot be specified.

maxnode=*n*

Specifies the maximum number of nodes, including any anticipated in future expansion with [sfcnode\(1M\)](#), that share a file system constructed on disks.

The default is **16**.

A value can be specified in the range from **1** to **32**.

maxvol=*n*

Specifies the maximum number of partitions, including any anticipated in future expansion with [sfcadd\(1M\)](#), that constitute the file system constructed on disks.

The default is **16**.

A value can be specified in the range from **1** to **32**.

mdsnodel=*all|num*

Specifies the number of MDS placing nodes. MDS placing nodes are selected from head of the host names specified with **node** option. When **all** is specified, all nodes are set up as MDS placing node. If and only if **all** is specified, the node added to node information by [sfcnode\(1M\)](#) after constructing the file system is also set up as a MDS placing node.

The default is **2**.

0 or less value and value bigger than the number of nodes specified with **node** option cannot be specified.

metasz=*n*

Specifies the size of the meta-data area. The unit is megabytes.

The default is about **10%** of the file system size. When **maxdsz** has been specified, the default is set at about **10%** of the **maxdsz** value.

The size of the meta-data area shall not exceed the size of the representative partition.

mfrees=*n*

Specifies the minimum percentage of free V-data area.

The default is **10%**.

A value can be specified in the range from **0** to **99**.

nblkpext=*n*

Specifies an average number of blocks to be managed by one extent.

This parameter is used as a fragmentation factor for the file data area.

The default is **1**.

A value can be specified in the range from **1** to **100**.

nbpi=*n*

Specifies the byte count of data per i-node. Specify a larger value for this parameter if the meta-data area is to be saved.

The default is **8192**.

A value can be specified in the range from **512** to **1,048,576**.

node=*host[:host],...*

Registers the name of a host that will share the file system. If multiple nodes share the file system, specify multiple entries for this parameter.

Hosts whose names are to be specified with *host* must be able to communicate with each other.

MDS placing nodes are selected from head of the host names specified with this option. The number of MDS(meta-data server) placing nodes are specified by the **mdsnodel** option.

Unless this option is specified, only the host name of the local node is registered. A maximum of **four** entries can be specified.

Use a colon delimiter to specify multiple host names. The multiplexed network path is used for

communication path with AC and MDS, The host name of each node can be specified to **four** maximums.

opt=st

Selects either space or time on which to base the optimization of free block allocation.

s indicates space and **t** indicates time.

The default is **t**.

OPERAND

The following operand can be specified:

raw_device_file

Specifies a character-special device for the representative partition of the GFS Shared File System.

A device specified for a file data partition or an update log partition cannot be specified for the representative partition.

EXAMPLES

```
# mkfs -F sfcfs -o node=host1,host2 /dev/sfdsk/gfs/rdsk/volume1

# mkfs -F sfcfs -o node=host1,host2 -o log=/dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/gfs/rdsk/volume1

# mkfs -F sfcfs -o node=host1,host2 -o log=/dev/sfdsk/gfs/rdsk/volume2,
data=/dev/sfdsk/gfs/rdsk/volume3 -o data=/dev/sfdsk/gfs/rdsk/volume4 /dev/sfdsk/gfs/rdsk/volume1

# mkfs -F sfcfs -o mdsnode=all -o node=host1,host2,host3,host4 -o log=/dev/sfdsk/gfs/rdsk/volume2 /dev/
sfdsk/gfs/rdsk/volume1

# mkfs -F sfcfs -o mdsnode=3 -o node=host1,host2,host3,host4 -o log=/dev/sfdsk/gfs/rdsk/volume2 /dev/sf
dsk/gfs/rdsk/volume1
```

FILE**/etc/vfstab**

Environment file for the file system.

SEE ALSO

[sfcadd\(1M\)](#), [sfcnewfs\(1M\)](#), [sfcnode\(1M\)](#), [sfcsetup\(1M\)](#), [sfcstune\(1M\)](#).

[fsck\(1M\)](#), [mkfs\(1M\)](#), [vfstab\(4\)](#), and [attributes\(5\)](#) in the "Solaris X Reference Manual Collection."

NOTES

- *host* is necessary to be the host name of the node which constitutes the GFS common file system registered into the management partition by [sfcsetup\(1M\)](#).
- A file system can be created only when all hosts that share the file system have started operation and GFS Shared File System has been installed. The reason is that the information for these hosts is required.
- If another option is specified after a comma (,) that follows a node option, the option will have no effect because it will be interpreted as a host name.
- In the GFS Shared File System, the data block size is fixed at **8192** bytes and cannot be altered.
- To re-create the GFS Shared File System by modifying its configuration, first delete the file system with [sfcadm\(1M\)](#).
- When a file system is created, no device being added can be specified. Use [sfcinfo\(1M\)](#) to confirm whether the device to be specified is being added.
- The file system configuration must be MDS can place in all nodes, if building the file system (if possible).
- The I/O performance of the application has influence while MDS perform meta-data area initialization

F.1 Commands for Common File System Management

after the file system start. Specify a **clrmeta** option to avoid this influence.

F.1.5 mount_sfcs(1M) Mounts the GFS Shared File System on the self-mode.

SYNOPSIS

mount -F sfcs *special* | *mount_point*
mount -F sfcs *special* *mount_point*

DESCRIPTION

The **mount** command performs mount processing for GFS Shared File System on the self-node only. This command can be executed from any node that can use GFS Shared File System, and during the mount processing, can start **sfcsd** and **MDS** (if necessary).

Only a super user can perform mount processing.

When mount processing goes wrong, error information is outputted to a console. Please perform management according to the outputted error information.

OPTIONS

The following options are supported:

-o *mount_options*

Specifies the mount options of the GFS Shared File System entry in **vfstab**. Multiple options specific to the file system must be delimited with a comma.

The following sub-options can be used:

closesync

Reflects all update data not yet reflected in the file when the file is closed for the last time.

intr | **nointr**

Enables (**intr**) or disables (**nointr**) for the file system a keyboard interrupt that forcibly terminates a suspended process on the locked file system that is awaiting a file operation. The default is **intr**.

noatime

Does not update the file access time.

noforcedirectio | **forcedirectio**

Mounts the file system with forced direct I/O on (**forcedirectio**) or off (**noforcedirectio**). If **forcedirectio** is specified, access to the file system is forcibly attempted by direct I/O access. If the access request does not meet the requirements for direct I/O access, normal access is performed. Direct I/O access directly transfers data between the user space and a disk without the intervention of a data transfer buffer that has been prepared in the kernel space by the file system. The default is **noforcedirectio**, in which case normal access is performed.

nosuid

The file system is mounted with **setuid** execution enabled. If, however, **nosuid** is specified, the file system is mounted with **setuid** execution disabled.

rw | **ro**

Mounts the file system as read/write enabled (**rw**) or read only (**ro**). The default is **rw**.

OPERANDS

The following operands can be specified:

special

Specifies the block special device name of a representative partition of the file system to be mounted.

mount_point

Specifies the name of the directory to be mounted.

EXAMPLES

```
# mount -F sfcfs /dev/sfdsk/gfs/dsk/volume1 /mount_point
# mount -F sfcfs /dev/sfdsk/gfs/dsk/volume1
# mount -F sfcfs /mount_point
```

FILES

/etc/mnttab

Table for the currently mounted file system.

/etc/vfstab

Environment file for the file system.

SEE ALSO

[umount_sfcfs\(1M\)](#), [sfcmntgl\(1M\)](#), [sfcumntgl\(1M\)](#).

mount(1M) , **mnttab(4)**, **vfstab(4)** in the "*Solaris X Reference Manual Collection*."

NOTES

- sfcfrmd daemon must be running to enable the execution of this command.
- Nodes outside the MDS positioning range alone cannot be mounted.
- The *special* and *mount_point* operands must be specified in **/etc/vfstab** in the same combination.
- Mount processing is always performed using the mount options for the mount point described in **/etc/vfstab**. When no mount options are described in **/etc/vfstab**, mount processing is performed without any options.
- **mount_sfcfs** is not accepted generic_option of **mount(1M)**.

F.1.6 `umount_sfcfs(1M)` Unmounts the GFS Shared File System from the self-node.

SYNOPSIS

`umount special | mount_point`

DESCRIPTION

The `umount_sfcfs` command is used to unmount the GFS Shared File System from a self-node.

During the unmount processing, an stop `sfcsd` and MDS when running on the node.

Only a superuser can perform unmount processing.

When unmount processing goes wrong, error information is outputted to a console. Please perform management according to the outputted error information.

OPERANDS

The following operands can be specified:

special

Specifies the name of the block special device of a representative partition of the file system to be mounted.

mount_point

Specifies the name of the directory to be mounted.

EXAMPLES

```
# umount /dev/sfdsk/gfs/dsk/volume1
# umount /mount_point
```

FILES

`/etc/mnttab`

Table for the currently mounted file system.

`/etc/vfstab`

Environment file for the file system.

SEE ALSO

[mount_sfcfs\(1M\)](#), [sfcmntgl\(1M\)](#), [sfcumntgl\(1M\)](#).

`umount(1M)`, `mnttab(4)`, `vfstab(4)` in the "*Solaris X Reference Manual Collection*."

NOTE

- `umount_sfcfs` is not support `-f` option of `umount(1M)`.

F.2 Management Commands Specific to the GFS Shared File System

F.2.1 sfcadd(1M) Adds file data partitions

SYNOPSIS

```
sfcadd -D raw_device_file [, raw_device_file, ...] t_raw_device_file  
sfcadd -M raw_device_file [, raw_device_file, ...] t_raw_device_file
```

DESCRIPTION

sfcadd adds the file data partition, and the meta partition of the GFS Shared File System having the representative partition specified by operand *t_raw_device_file*, and separates an update log area. No data has to be saved or restored.

The file data partition of the file system can be added even if the file system is mounted status or not used due to the file system is in unmounted status.

If the partition is not added due to a system failure, the target partition is correctly managed as being currently processed as the file system. Use [sfcinfo\(1M\)](#) to confirm whether the partition is being processed.

OPTIONS

The following options can be specified:

-D *raw_device_file* [, *raw_device_file*]

Uses a character special device to specify the partition to be added as the file data area. When adding multiple partitions, a comma must separate the character special devices specified.

-M *raw_device_file* [, *raw_device_file*]

Uses a character special device to specify the partition to be added as the meta data area. When adding multiple partitions, a comma must separate the character special devices specified.

OPERAND

The following operand can be specified:

t_raw_device_file

Specifies path name of a character-special device for the representative partition of the GFS Shared File System to target for the transaction.

EXAMPLES

Newly adding /dev/sfdsk/gfs/rdsk/volume2 and /dev/sfdsk/gfs/rdsk/volume3 as file data partitions into the existing GFS Shared File System whose representative partition is /dev/sfdsk/gfs/rdsk/volume1.

```
# sfcadd -D /dev/sfdsk/gfs/rdsk/volume2,/dev/sfdsk/gfs/rdsk/volume3 /dev/sfdsk/gfs/rdsk/volume1
```

Newly adding /dev/sfdsk/gfs/rdsk/volume2 as meta partitions into the existing GFS Shared File System whose representative partition is /dev/sfdsk/gfs/rdsk/volume1.

```
# sfcadd -M /dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/gfs/rdsk/volume1
```

SEE ALSO

[fsck_sfcs\(1M\)](#), [mkfs_sfcs\(1M\)](#), [sfcadm\(1M\)](#), [sfcinfo\(1M\)](#), [sfcnewfs\(1M\)](#), [sfcnode\(1M\)](#).

NOTES

- To use **sfcadd** during the file system is mounted, the target file system must be one mounted as writable and readable by the command execution node.
- A file system can be added partitions only when all hosts that share the file system have started operation. The reason is that the information for these hosts is required.
- **sfcadd** can add partitions up to the maximum number of partitions (**maxvol**) specified by [sfcnewfs\(1M\)](#). If the maximum data size (**maxdsz**) specified by [sfcnewfs\(1M\)](#) is exceeded because of partition addition, a free area that cannot be used due to a management area shortage may be generated.
- To execute **sfcadd** for the unmounted file system in all nodes, the status of the file system must be normal. If the file system status is abnormal, sfcadd outputs the message "file system (*partition-name*) is corrupted.
TO FIX: fsck -F sfcfs *partition-name*"
Correct the file system status with [fsck_sfcfs\(1M\)](#), then reexecute sfcadd.
- Only the addition of new file data partitions or meta partitions can be made for the file system that is operating mounted file system. This function can be executed only in the node in which the target file system is in mounted mounted status.
- Assume that **sfcadd** terminates abnormally. In this case, if the target file system contains a partition being processed, reexecute the sfcadd command to complete the processing. If to do not that, otherwise, the partition being processed cannot be applied elsewhere.

F.2.2 sfcadm(1M) Changes partition information settings

SYNOPSIS

```

sfcadm -r [ -a ] raw_device_file
sfcadm -L size raw_device_file
sfcadm -N raw_device_file,...
sfcadm -c {all | num} raw_device_file
sfcadm -u host[:host],... [-g p_mds[,s_mds,mds,...]] [-p p_port[,s_port]] [-o keyword=value,...]
raw_device_file,...
sfcadm -m host[:host],... [-g p_mds[,s_mds,mds,...]] [-p p_port[,s_port]] [-o keyword=value,...]
raw_device_file,...
sfcadm -g p_mds[,s_mds,mds,...] raw_device_file
sfcadm -o keyword=value,... raw_device_file
sfcadm -p p_port[,s_port] raw_device_file
sfcadm -h ohost nhost
sfcadm -D raw_device_file,...
sfcadm raw_device_file,...
  
```

DESCRIPTION

The **sfcadm** command changes the partition information settings for the GFS Shared File System.

In the GFS Shared File System, the information about the group of partitions that makes up the file system is retained in each partition.

Partition information settings must be changed in the following cases:

- Device information (path name, device number) for the file system is changed.
- Information about the nodes sharing the file system is changed.
- The host name is changed
- The file system is migrated to another system.
- The update log area size of a update log partition is changed.

Use the command in the following cases:

- Recovery for a super block and partition configuration data.
- For the partition configuration data and copying of a super block
- Recovery if a copy of the partition configuration data no longer matches the original information
- Changing the information in the file system configure information in a management partition and deleting the file system that is specified
- Change the number of placing MDS (meta-data server) nodes.
- Change the priority of nodes that places MDS (meta-data server) as primary MDS or secondary MDS.

The **sfcadm** updates the entries in the file system configure information in a management partition, which changes the partition information for the GFS Shared File System and/or the file system environment information. If changes are made to the partition information settings, the information retained in each of the partitions is updated.

This command can only be used if the file system is unmounted in all nodes beforehand.

OPTIONS

The following options can be specified:

-D *raw_device_file*,...

Deletes the GFS Shared File System in which the representative partition provided by file system configure information in a management partition has been registered. Use a comma delimiter to specify multiple files.

-L *size*

Sets the update log area size of the log partition. For *size*, specify an appropriate update log area size. The unit is megabytes. Specify a value in the range from **5** to **100**. The update log area must have been separated beforehand if this option is used.

-N or no options specified

Updates the node and device information for all hosts in the partition configuration data. Use this option if the minor numbers of the specified devices for hosts are different after the special file name of the file system in use has been changed or copying between partitions is performed. If all hosts have the same minor number, specify only the device information, without specifying the **-N** option. Then updates the file system configure information in a management partition.

-a

When the **-r** option is specified, performs the recovery using a copy of the super block in the partition of the recovery source and a copy of the partition configuration data.

-c {all|num}

Change the MDS placing information. **Zero** or less value and value bigger than the number of nodes that share the file system cannot be specified. When **all** is specified, all nodes are set up as MDS placing node. If and only if **all** is specified, the node added to node information by [sfcnode\(1M\)](#) after changing the MDS placing information is also set up as a MDS placing node. The file system configure information in a management partition is also updated.

-g p_mds,s_mds,mds,...

Specifies the host name that MDS (meta-data server) can place delimited with a comma. To change either primary MDS node or secondary MDS node, use the *p_mds* or *s_mds* syntax and specify the new host name. In order to change host except primary and secondary, all hosts must be specified. To change a configuration with only a primary MDS, specify only *p_mds* (a comma delimiter is not required). The file system configure information in a management partition is also updated.

-h o_host n_host

Enables a host name to be changed. Use the "*o_host n_host*" syntax and specify the host names. In the file system configure information in a management partition, then, all occurrences of *o_host* are changed to *n_host*. The host names registered in the partition configuration data also change.

-m host[:host],...

Changes the node and device information in the partition configuration data to the information specified. Use a comma delimiter to specify multiple host names.

Use a colon delimiter to specify multiple host names. The multiplexed network path is used for communication path with AC and MDS, The host name of each node can be specified to **four** maximums.

And specifies the **-g** options for MDS host names. The file system configure information in a management partition is also updated. Use this option when a file system in use is being migrated to another system.

-p p_port,s_port

Specifies the primary port name and the secondary port name delimited with a comma. To change either port name, use the *p_port*, or *,s_port* syntax and specify the new port name.

To change only the primary port name, specify only *p_port* (a comma delimiter is not required).

And specifies the **-g**, and **-c** options for MDS node. The file system configure information in a management partition is also updated.

-o keyword=value,...

Specifies a variable setting that can be tuned in the GFS Shared File System using the syntax "*variable-name=value*" syntax. Use a comma delimiter to specify multiple variables. When deleting a registered variable, use the specification "*variable-name=*". You can specify the timeout of communication between nodes such as followings:

-o CON_TIMEOUT=60

For example, the timeout value of the mutilation between nodes is 60 seconds. The file system configure information in a management partition is also updated.

-r raw_device_file.

Specifies the character-special device that will be the recovery source when a super block and partition configuration data are restored. When the **-a** option is specified, redundant blocks in the specified source partition are used for restoring the data into other partitions.

-u host[:host],...

Changes the node and device information in the partition configuration data to the information

F.2 Management Commands Specific to the GFS Shared File System

specified. Use a comma delimiter to specify multiple host names.

Use a colon delimiter to specify multiple host names. The multiplexed network path is used for communication path with AC and MDS, the host name of each node can be specified to **four** maximums.

In addition, the **-g**, and **-p** options can be used to change MDS host names, and port names.

The file system configure information in a management partition is also updated. Use this option when the node and device information must be rewritten after the special file name of the file system in use has been changed or copying between partitions is performed.

OPERANDS

The following operands can be specified:

raw_device_file,...

Specify character-special devices for all partitions in the GFS Shared File System using a comma delimiter (.).

raw_device_file

Specify a character-special device for the representative partition of the GFS Shared File System.

EXAMPLES

- Perform recovery of another partition using a partition information of the recovery source.

```
# sfcadm -r /dev/sfdsk/gfs/rdsk/volume1
```

- Perform recovery of another partition using a copy of a partition information of the recovery source.

```
# sfcadm -r -a /dev/sfdsk/gfs/rdsk/volume1
```

- Change the size of the update log area to 50 megabytes.

```
# sfcadm -L 50 /dev/sfdsk/gfs/rdsk/volume1
```

- Update the node and device information for all hosts in the partition information.
Hosts have different minor numbers.

```
# sfcadm -N /dev/sfdsk/gfs/rdsk/volume1,/dev/sfdsk/gfs/rdsk/volume2
```

Hosts have the same minor number.

```
# sfcadm /dev/sfdsk/gfs/rdsk/volume1,/dev/sfdsk/gfs/rdsk/volume2
```

- Rewrite the node and device information in the partition information.

```
# sfcadm -u host1,host2 -g host1,host2 -p port1,port2 /dev/sfdsk/gfs/rdsk/volume1,/dev/sfdsk/gfs/volume2,/dev/sfdsk/gfs/rdsk/volume3
```

- Migrate a specific file system to another system.

```
# sfcadm -m host1,host2,host3 -g host1,host2 -p port1,port2 /dev/sfdsk/gfs/rdsk/volume1,/dev/sfdsk/gfs/rdsk/volume2,/dev/sfdsk/gfs/rdsk/volume3
```

- Change the number of nodes placing MDS.
Change the number of the nodes placing MDS to 3.

```
# sfcadm -c 3 /dev/sfdsk/gfs/rdsk/volume1
```

Place MDS to the all nodes.

```
# sfcadm -c all /dev/sfdsk/gfs/rdsk/volume1
```

- Change the priority ranking of nodes placing MDS.
Change primary MDS to host1.

```
# sfcadm -g host1, /dev/sfdsk/gfs/rdsk/volume1
```

Change secondary MDS to host2.

```
# sfcadm -g ,host2 /dev/sfdsk/gfs/rdsk/volume1
```

Change the priority ranking of nodes placing MDS.

```
# sfcadm -g host1,host2,host3 /dev/sfdsk/gfs/rdsk/volume1
```

- Assign the same port name to the primary and secondary ports.

```
# sfcadm -p sfcfs-1 /dev/sfdsk/gfs/rdsk/volume1
```

- Register the variable XXX=60.

```
# sfcadm -o XXX=60 /dev/sfdsk/gfs/rdsk/volume1
```

- Change the host name host1 to host2.

```
# sfcadm -h host1 host2
```

- Delete a specific file system.

```
# sfcadm -D /dev/sfdsk/gfs/rdsk/volume1
```

SEE ALSO

[mkfs_sfcfs\(1M\)](#), [sfcadd\(1M\)](#), [sfcinfo\(1M\)](#), [sfcnewfs\(1M\)](#), [sfcnode\(1M\)](#), [sfcsetup\(1M\)](#).

NOTES

- A file system can be changed only when all hosts that share the file system have started operation and GFS Shared File System has been installed. The reason is that the information for these hosts is required.
- *host* is necessary to be the host name of the node which constitutes the GFS common file system registered into the management partition by [sfcsetup\(1M\)](#).
- Using **-u** option with specified new node name or device name, **-g** options must be specified.
- When the representation partition name of the GFS Shared File System is changed by **sfcadm**, or when performing the shift to other file systems, **/etc/vfstab** needs to be set up.
- Using **sfcadm** with **-D** option to delete file system which is linked as backup source or backup destination using [sfcproxyjoin\(1M\)](#) causes an error. Unlink file system before deletion.

F.2.3 `sfcdrr(1M)` Restores backup data and checks device restoring

SYNOPSIS

```
sfcdrr [ -f ] [ if=file ] of=special [ ibs=n ] [ obs=n ]  
sfcdrr -i [ if=file ] [ ibs=n ]
```

DESCRIPTION

The **sfcdrr** command restores to a disk GFS Shared File System backed up with **dd(1M)**.

Before writing the data to the disk, **sfcdrr** checks the following:

- The specified input file contains GFS Shared File System data or GFS Local File System data.
- The specified disk is the same disk that was backed up.
- The disk is not being mounted.
- All disk partitions are the same size.

If all of the above conditions are not true, the command terminates without restoring the data. Unless an **if** operand is specified, the standard input is used. An **of** operand must be specified.

As with **dd(1M)**, input and output block size can be specified in bytes. The alphabetic characters **k**, **b**, and **w**, if specified after a number, indicate **1024** bytes, **512** bytes, and **2** bytes, respectively. In addition, multiplication can be expressed by inserting an **x** between two numeric values.

After restoring, the **sfcdrr** command reports the number of input/output blocks, the number of complete blocks transferred, and the number of partial blocks transferred.

OPTIONS

The following options can be specified:

- f
Does not check to see whether the output device is the same as the device used for backing-up. Use this option if to restore backed-up data to a disk different from which the data was backed up.
- i
Displays the configuration of the partitions of the backed-up file system, the time required to create the file system, and the last mounted directory path. This option does not restore the data.

OPERANDS

The following operands can be specified:

- if=*file***
Input file name. The default is the standard input.
- of=*special***
Output device name.
- ibs=*n***
Specify the input block size in *n* bytes. The default is **512**.
- obs=*n***
Specify the output block size in *n* bytes. The default is **1024k**.

EXIT STATUS

The termination status is returned as follows:

- 0 The backed-up data has been restored correctly.
- >0 An error has occurred.

DIAGNOSTICS

f+p records in(out)

The number of complete (*f*) and partial (*p*) blocks that have been read (written)

SEE ALSO

[fck_sfcfs\(1M\)](#), [sfadd\(1M\)](#), [sfcinfo\(1M\)](#), [sfcmbkup\(1M\)](#), [sfcmrstr\(1M\)](#), [dd\(1M\)](#), [uncompress\(1\)](#) in the "Solaris X Reference Manual Collection."

NOTES

- If the backed-up data has been restored to a disk other than the one that the data was backed because the **-f** option is specified, the partition information must be changed by executing [sfadm\(1M\)](#).
- Before **sfddr** is used, the data may have to be uncompressed with **uncompress(1)**. Uncompressing data will be required if the **sfddr** command is used for a backup created with [sfcmbkup\(1M\)](#) with the **-c** option specified.

```
# dd if=/dev/rmt/0 ibs=256k | uncompress -c | sfddr -i
```

- Do not backing-up the file system is being added a partition. Use [sfcinfo\(1M\)](#) to confirm whether the partition is being processed, if partition is being processed, execute **sfddr** after reexecute [sfadd\(1M\)](#) to complete process.

F.2.4 sfcdu(1M) Summarize disk usage

SYNOPSIS

```
/usr/bin/sfcdu [-adkLr] [-o | -s] [file ... ]  
/usr/xpg4/bin/sfcdu [ -a | -s ] [ -krx ] [ file ... ]
```

DESCRIPTION

The **sfcdu** command is **du** command corresponding to large-scale file system which size is over 1 terabytes.

sfcdu operates with file system with scale not less than 2 terabytes. **sfcdu** also operates on a system other than the GFS Shared File System.

sfcdu writes to standard output the size of the file space allocated to, and the size of the file space allocated to each subdirectory of, the file hierarchy rooted in each of the specified files. The size of the file space allocated to a file of type directory is defined as the sum total of space allocated to all files in the file hierarchy rooted in the directory plus the space allocated to the directory itself.

Files with multiple links will be counted and written for only one entry. The directory entry that is selected in the report is unspecified. By default, file sizes are written in 512-byte units, rounded up to the next **512-byte** unit.

[/usr/xpg4/bin/sfcdu]

When **sfcdu** cannot obtain file attributes or read directories (see **stat(2)**), it will report an error condition and the final exit status will be affected.

OPTIONS

The following options can be specified for **/usr/bin/sfcdu** and **/usr/xpg4/bin/sfcdu**:

- a**
In addition to the default output, report the size of each file not of type directory in the file hierarchy rooted in the specified file. Regardless of the presence of the **-a** option, non-directories give, as *file* operands will always be listed.
- k**
Write the files sizes in units of **1024** bytes, rather than the default **512-byte** units.
- s**
Instead of the default output, report only the total sum for each of the specified files.

[/usr/bin/sfcdu]

The following options are supported for **/usr/bin/sfcdu** only:

- L**
Process symbolic links by using the file or directory which the symbolic link references, rather than the link itself.
- d**
Do not cross file system boundaries. For example, **sfcdu -d /** reports usage only on the root partition.
- o**
Do not add child directories' usage to a parent's total. Without this option, the usage listed for a particular directory is the space taken by the files in that directory, as well as the files in all directories beneath it. This option does nothing if **-s** is used.
- r**
Generate messages about directories that cannot be read, files that cannot be opened, and so forth, rather than being silent (the default).

[/usr/xpg4/bin/sfcdu]

The following options are supported for **/usr/xpg4/bin/sfcdu** only:

- r**

By default, generate messages about directories that cannot be read, files that cannot be opened, and so forth.

-x

When evaluating file sizes, evaluate only those files that have the same device as the file specified by the *file* operand.

OPERAND

The following operand can be specified:

file

The path name of a file whose size is to be written. If no *file* is specified, the current directory is used.

EXAMPLES

```
% sfcdu
1324 .
% sfcdu -k
662 .
% sfcdu -a
60 ./file00
400 ./file01
28 ./file02
832 ./file03
2 ./dir00
1324 .
```

EXIT STATUS

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

SEE ALSO

[sfcinfo\(1M\)](#)

[ls\(1\)](#), [stat\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [XPG4\(5\)](#) in the " Solaris X Reference Manual Collection "

NOTES

- A file with two or more links is counted only once. If, however, there are links between files in different directories where the directories are on separate branches of the file system hierarchy, **sfcdu** will count the excess files more than once.
- Files containing holes will result in an incorrect block count.

F.2.5 sfcdump(1M) Dumps a GFS shared file system

SYNOPSIS

sfcdump [*options*] [*arguments*] *files_to_dump*

DESCRIPTION

sfcdump backs up all files specified by *files_to_dump* (normally either a whole file system or files within the GFS Shared File System changed after a certain date) to magnetic tape, diskette, or disk file. When running **sfcdump**, the file system must be inactive; otherwise, the output of **sfcdump** may be inconsistent and restoring files correctly may be impossible. A file system is inactive when the file system is unmounted in all nodes. A file system is not considered inactive if one tree of the file system is quiescent while another tree has files or directories being modified.

options is a single string of one-letter **sfcdump** options.

arguments may be multiple strings whose association with the options is determined by order. That is, the first argument goes with the first option that takes an argument; the second argument goes with the second option that takes an argument, and so on.

files_to_dump is required and must be the last argument on the command line. See OPERAND for more information.

With most devices **sfcdump** can automatically detect the end-of-media. Consequently, the **d**, **s**, and **t** options are not necessary for multi-volume dumps, unless **sfcdump** does not understand the way the device detects the end-of-media, or the files are to be restored on a system with an older version of the restore command.

The **sfcdump** and **ufsdump** commands are functionally compatible, though the backup formats are different. As a result, the **ufsrestore** command cannot use the backup data created using the **sfcdump** command for restoration. In the same way, the **sfcrestore** command cannot use the backup data created using the **ufsdump** command for restoration.

OPTIONS

The following options can be specified:

0-9

The "dump level." All files specified by *files_to_dump* that have been modified since the last **sfcdump** at a lower dump level are copied to the *dump_file* destination (normally a magnetic tape device). For instance, if a "level 2 " dump was done on Monday, followed by a "level 4 " dump on Tuesday, a subsequent "level 3 " dump on Wednesday would contain all files modified or added since the "level 2 " (Monday) backup. A "level 0 " dump copies the entire file system to the *dump_file*.

a *archive_file*

Archive file. Archive a dump table-of-contents in the specified *archive_file* to be used by [sfcrestore\(1M\)](#) to determine whether a file is in the dump file that is being restored.

b *factor*

Blocking factor. Specify the blocking factor for tape writes. The default is **20** blocks per write for tapes of density less than 6250BPI (bytes-per-inch). The default blocking factor for tapes of density 6250BPI and greater is **64**. The default blocking factor for cartridge tapes (c option) is **126**. The highest blocking factor available with most tape drives is **126**. Note: the blocking factor is specified in terms of **512**-byte blocks, for compatibility with **tar(1)**.

c

Cartridge. Set the defaults for cartridge instead of the standard half-inch reel. This sets the density to 1000BPI and the blocking factor to 126. Since **sfcdump** can automatically detect the end-of-media, only the blocking parameter normally has an effect. When cartridge tapes are used, and this option is not specified, **sfcdump** will slightly miscalculate the size of the tape. If the **b**, **d**, **s** or **t** options are specified with this option, their values will override the defaults set by this option.

d *bpi*

Tape density. Not normally required, as **sfcdump** can detect end-of-media. This parameter can be

used to keep a running tab on the amount of tape used per reel. The default density is 6250BPI except when the **c** option is used for cartridge tape, in which case it is assumed to be 1000BPI per track. Typical values for tape devices are:

1/2" tape

6250 BPI

1/4" cartridge

1000 BPI (The tape densities and other options are documented in the st(7D) man page.)

D

Diskette. Dump to diskette.

f dump_file

Dump file. Use *dump_file* as the file to dump to, instead of /dev/rmt/0. If *dump_file* is specified as -, dump to standard output.

If the name of the file is of the form *machine:device*, the dump is done from the specified machine over the network using **rmt(1M)**. Since **sfcdump** is normally run by root, the name of the local machine must appear in the *.rhosts* file of the remote machine. If the file is specified as *user@machine:device*, **sfcdump** will attempt to execute as the specified user on the remote machine. The specified user must have a *.rhosts* file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.

I (lower-case L)

Autoload. When the end-of-tape is reached before the dump is complete, take the drive offline and wait up to two minutes for the tape drive to be ready again. This gives autoloading (stackloader) tape drives a chance to load a new tape. If the drive is ready within two minutes, continue. If it is not, prompt for another tape and wait.

L string

Sets the tape label to *string*, instead of the default **none**. *string* may be no more than fifteen characters long. If it is longer, it is truncated and a warning printed; the dump will still be done.

n

Notify all operators in the sys group that **sfcdump** requires attention by sending messages to their terminals, in a manner similar to that used by the **wall(1M)** command. Otherwise, such messages are sent only to the terminals (such as the console) on which the user running **sfcdump** is logged in.

o (lower-case o)

Offline. Take the drive offline when the dump is complete or the end-of-media is reached and rewind the tape, or eject the diskette. In the case of some autoloading 8mm drives, the tape is removed from the drive automatically. This prevents another process that rushes in to use the drive, from inadvertently overwriting the media.

s size

Specify the *size* of the volume being dumped to. Not normally required, as **sfcdump** can detect end-of-media. When the specified size is reached, **sfcdump** waits for you to change the volume. **sfcdump** interprets the specified size as the length in feet for tapes and cartridges, and as the number of 1024-byte blocks for diskettes. The values should be a little smaller than the actual physical size of the media (for example, 425 for a 450-foot cartridge). Typical values for tape devices depend on the **c** option, for cartridge devices, and the **D** option for diskettes:

1/2" tape

2300 feet

60-Mbyte 1/4" cartridge

425 feet

150-Mbyte 1/4" cartridge

700 feet

diskette

1422 blocks (Corresponds to a 1.44-Mbyte diskette, with one cylinder reserved for bad block information.)

S

Size estimate. Determine the amount of space that is needed to perform the dump without actually doing it, and display the estimated number of bytes it will take. This is useful with incremental dumps to

F.2 Management Commands Specific to the GFS Shared File System

determine how many volumes of media will be needed.

t tracks

Specify the number of tracks for a cartridge tape. Not normally required, as **sfcdump** can detect end-of-media. The default is **9** tracks. The **t** option is not compatible with the **D** option. Values for Sun-supported tape devices are:

60-Mbyte cartridge

9 tracks

150-Mbyte cartridge

18 tracks

T timeout [hms]

Sets the amount of time to wait for an autoload command to complete. This function modifier is ignored unless the **I** function modifier has also been specified. The default time period to wait is **two** minutes. Specify time units with a trailing **h** (for hours), **m** (for minutes), or **s** (for seconds). The default unit is minutes.

u

Update the dump record. Add an entry to the file `/etc/sfdfs_dumpdates`, for each file system successfully dumped that includes the file system name, date, and dump level.

v

Verify. After each tape or diskette is written, verify the contents of the media against the source file system. If any discrepancies occur, prompt for new media, then repeat the dump/verification process. The file system must be unmounted in all nodes. This option cannot be used to verify a dump to standard output.

w

Warning. List the file systems that have not been backed up within a day. This information is gleaned from the files `/etc/sfdfs_dumpdates` and `/etc/vfstab`. When the **w** option is used, all other options are ignored. After reporting, **sfcdump** exits immediately.

W

Warning with highlight. Similar to the **w** option, except that the **W** option includes all file systems that appear in `/etc/sfdfs_dumpdates`, along with information about their most recent dump dates and levels. File systems that have not been backed up within a day are highlighted.

OPERAND

The following operand can be specified:

files_to_dump

Specifies the files to dump. Usually it identifies a whole GFS Shared File System by its raw device name of the representative partition (for example, `/dev/sfdsk/gfs/rdsk/volume1`). Incremental dumps (levels **1** to **9**) of files changed after a certain date only apply to a whole file system. Alternatively, *files_to_dump* can identify individual files or directories. All files or directories are dumped, which is equivalent to a level **0** dump; however, `/etc/sfdfs_dumpdates` is not updated, even with the **u** option specified. In all cases, the files must be contained in the same file system, and the file system must be local to the system where **sfcdump** is being run. *files_to_dump* is required and must be the last argument on the command line. If no options are given, the default is `9uf /dev/rmt/0 files_to_dump`.

EXAMPLES

Example 1: A sample display of the **sfcdump** command. To make a full dump of the GFS Shared File System configured on `/dev/sfdsk/gfs/rdsk/volume1` onto a 150-Mbyte tape unit 0, use:

```
# sfcdump 0cfu /dev/rmt/0 /dev/sfdsk/gfs/rdsk/volume1
```

To make and verify an incremental dump at level 5 of the GFS Shared File System configured on `/dev/sfdsk/gfs/rdsk/volume6`, on a 1/2" reel tape unit 1, use:

```
# sfcdump 5fuv /dev/rmt/1 /dev/sfdsk/gfs/rdsk/volume1
```

EXIT STATUS

While running, **sfcdump** emits many verbose messages. **sfcdump** returns the following exit values:

0	Normal exit.
1	Startup errors encountered.
3	Abort - no checkpoint attempted.

FILES

/dev/rmt/0	default unit to dump to
/etc/sfcfs_dumpdates	dump date record
/etc/group	to find group sys
/etc/hosts	to gain access to remote system with drive
/etc/vfstab	list of file systems

SEE ALSO

[sfcrestore\(1M\)](#).

cpio(1), **tar(1)**, **dd(1M)**, **devnm(1M)**, **prvtoc(1M)**, **rmt(1M)**, **shutdown(1M)**, **volcopy(1M)**, **wall(1M)**, **st(7D)** in the "Solaris X Reference Manual Collection."

NOTES**Read errors**

Up to **32** read errors on the file system are ignored.

Process Per Reel

Because each reel requires a new process, parent processes for reels that are already written hang around until the entire tape is written.

Operator intervention

sfcdump requires operator intervention on these conditions: end of volume, end of dump, volume write error, volume open error or disk read error (if there are more than a threshold of **32**). In addition to alerting all operators implied by the **n** option, **sfcdump** interacts with the operator on **sfcdump**'s control terminal at times when **sfcdump** can no longer proceed, or if something is grossly wrong. All questions **sfcdump** poses must be answered by typing **yes** or **no**, as appropriate.

Since backing up a disk can involve a lot of time and effort, **sfcdump** checkpoints at the start of each volume. If writing that volume fails for some reason, **sfcdump** will, with operator permission, restart itself from the checkpoint after a defective volume has been replaced.

Suggested Dump Schedule

It is vital to perform full, "level **0**", dumps at regular intervals. When performing a full dump, bring the machine down to single-user mode using **shutdown(1M)**. While preparing for a full dump, it is a good idea to clean the tape drive and heads. Incremental dumps should be performed with the system running in single-user mode.

Incremental dumps allow for convenient backup and recovery of active files on a more frequent basis, with a minimum of media and time. However, there are some tradeoffs. First, the interval between backups should be kept to a minimum (once a day at least). To guard against data loss as a result of a media failure (a rare, but possible occurrence), capture active files on (at least) two sets of dump volumes. Another consideration is the desire to keep unnecessary duplication of files to a minimum to save both operator time and media storage. A third consideration is the ease with which a particular backed-up version of a file can be located and restored. The following four-week schedule offers a reasonable tradeoff between these goals.

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday
The first week :	0(full)	5	5	5	5	3
The second week :		5	5	5	5	3
The third week :		5	5	5	5	3
The fourth week :		5	5	5	5	3

Although the Tuesday through Friday incremental contains "extra copies" of files from Monday, this scheme assures that any file modified during the week can be recovered from the previous day's incremental dump.

Process priority of the **sfdump** command

sfdump uses multiple processes to allow it to read from the disk and write to the media concurrently. Due to the way it synchronizes between these processes, any attempt to run dump with a nice (process priority) of '-5' or better will likely make **sfdump** run slower instead of faster.

Partition overlap

The partition that covers the entire disk contains other partitions on the disk. But even if multiple GFS Shared File Systems are configured on one disk, **sfdump** must not be executed on the entire disk partition. **sfdump** can handle only one GFS Shared File System at a time. To dump the multiple GFS Shared File Systems configured on one disk, **sfdump** must be executed on each file systems.

Warning when the **w** and **W** options are specified

The `/etc/vfstab` file does not allow the desired frequency of backup for file systems to be specified. Consequently, the **w** and **W** options assume file systems should be backed up daily, which limits the usefulness of these options.

Notices in case of using incremental dump

Incremental dump works by using the dump date record file (`/etc/sfcfs_dumpdates`). The file is updated only on the node where **sfdump** is executed. Consequently, before incremental dump being made on another node, the dump date record file on the node where previous dump was made must be copied.

Incremental dump in case that some node did not stop normally

In case that some node using the file system did not stop normally because of panic or the like, files updated on the node may not be dumped correctly by using incremental dump.

When incremental dump is made, **sfdump** judges whether the file should be dumped or not by comparing time stamp of the file and previous dump date recorded dump date record file. But in case that some node did not stop normally, only the contents of the files updated on the node may be updated in spite of the time stamps being not updated. These files may not be dumped by incremental dump and be dumped by next full dump.

Therefore, after some node using the file system did not stop normally, it is recommended that full dump be made next time.

F.2.6 sfcedquota(1M) Edits quota limits for each user

SYNOPSIS

```
sfcedquota [ -p proto_user ] username ...
sfcedquota -t
```

DESCRIPTION

sfcedquota is an editor for editing quota limits for each user of the GFS Shared File System. One or more users can be specified from the command line.

Assume that the GFS Shared File Systems that are mounted in the node that have executed **sfcedquota** include quotas files. For these file systems, the current quota limits of the user specified by operand *username* can be created in a temporary file in ASCII representation and displayed with the editor. Thereafter, the displayed quota limits can be corrected or new quota limits can be added. When the editor ends, **sfcedquota** reads the temporary file to change the quota limits.

The editor to be called is **vi**(1) so long as environment variable **EDITOR** does not specify another editor.

Only a super user can edit the quota limits. To set quota limits, a quotas file owned by the root is required in the root directory of the file system. The quotas file is first created with **touch**(1) and **chmod**(1).

proto_user and *username* can specify the login name or UID of a user. An unassigned UID can be specified, but an unassigned login name cannot be specified. If an unassigned user ID is specified, the quota information is valid when UID is assigned.

If the **-t** option is omitted, the editor displays the temporary file in the following format to edit the limit values for the number of blocks and the number of files:

```
fs mount_point blocks (soft = number, hard = number) inodes (soft = number, hard = number)
```

block indicates one block of **1,024** bytes (1K). *number* is changed to a value to be set as the number of blocks or files.

If the **-t** option is specified, the editor displays the temporary file in the following format to edit the soft time limit for the number of blocks and files:

```
fs mount_point blocks time limit = number tmunit, files time limit = number tmunit
```

If *number* indicates 0 and *tmunit* indicates **default**, this indicates the default time limit (one week). If *number* indicates a value other than 0, *tmunit* indicates a larger unit from among **month**, **week**, **day**, **hour**, **min**, and **sec**.

A soft time limit is set by editing *number* and *tmunit*. Because the characters following a *tmunit* unit are ignored, **months** or **minutes** can be specified.

OPTIONS

The following options can be specified:

- p** Copies the quota information of specified *proto_user* onto each specified *username*. This option is used to initialize quota information for a group of users.
- t** Edits the soft time limit of each file system.

OPERAND

The following operand can be specified:

username

username can specify the login name or UID of a user.

EXAMPLES

```
# sfcquota user1
# sfcquota user1 user2
# sfcquota -p protuser user1 user2
# sfcquota -t
```

EXIT STATUS

The following end statuses are returned:

0	Normal termination
32	Error

FILES

quotas

quota information file in the root directory of the file system

/etc/mnttab

Table of the file system being mounted

SEE ALSO

[sfcprepquota\(1M\)](#), [sfcquota\(1M\)](#), [sfcquotadm\(1M\)](#), [sfcquotaon\(1M\)](#).

chmod(1), **touch(1)**, **vi(1)** in the "Solaris X Reference Manual Collection."

F.2.7 `sfcrmstart(1M)` Starts FsRM on the self node

SYNOPSIS

`sfcrmstart`

DESCRIPTION

The `sfcrmstart` command is used to perform `sfcrmd` daemon start processing on the self-node. Only a super user performs `sfcrmd` daemon start processing.

EXAMPLES

```
# sfcrmstart
```

FILE

`/var/opt/FJSVsfcs/sfcfsrcm.conf`

Environment file of `sfcrmd` daemon

SEE ALSO

[sfcrmstop\(1M\)](#), [sfcsetup\(1M\)](#).

NOTES

- To enable the use of this command, the self-node must be multi user mode. This command cannot be used by single user mode.
- Within a given node, this command cannot be executed more than once.
- Within a given node, this command cannot be executed at the same time as [sfcrmstop\(1M\)](#).

F.2.8 `sfcrmstop(1M)` Stops FsRM on the self-node

SYNOPSIS

`sfcrmstop`

DESCRIPTION

The `sfcrmstop` command is used to stop FsRM operation on the self-node.

The GFS Shared File System can perform only in the state where it is not mounted by the applicable node.

Only a super user can perform FsRM stop processing.

EXAMPLE

```
# sfcrmstop
```

SEE ALSO

[sfcrmstart\(1M\)](#).

NOTES

- To enable the use of this command, the self-node must be multi user mode. This command cannot be used by single user mode.
- Within a given node, this command cannot be executed more than once.
- Within a given node, this command cannot be executed at the same time as [sfcrmstart\(1M\)](#).

F.2.9 sfcgetconf(1M) Backs up the management partition

SYNOPSIS

```
sfcgetconf backupfile
sfcgetconf -a backupfile
sfcgetconf -i fsid [-i fsid] backupfile
sfcgetconf -d raw_device_file [-d raw_device_file] backupfile
```

DESCRIPTION

The sfcgetconf command outputs next items to *backupfile*:

- The file system configuration information in management partition.
- The procedure for using the [sfcadm\(1M\)](#) command to rebuild the configuration.
- The contents of the **sfcsrm.conf** file when the **-a** option is specified.

The output backup file contains the procedure for executing the [sfcadm\(1M\)](#) command in the sh(1) shell script Format.

Management partition recovery is done using the script output by **sfcgetconf**,(1M) after the management partition has been initialized with the [sfcsetup\(1M\)](#) command.

OPTIONS

The following options can be specified:

-a

Outputs the contents of **sfcsrm.conf** to the first part of the backup file in the form of comments. (The output for no **-a** option will follow.)

-i fsid

When you specify this option (not for the all the file systems), only the information for the file system corresponding to the value specified in *fsid*, and the procedure for executing the [sfcadm\(1M\)](#) command that re-creates the information in the management partition for that file system, are output.

-d raw_device_file

Specifies the character device containing the partition that configures the file system. When you specify this option (not for all file systems), only the information for the file system corresponding to the specified device, and the procedure for executing the [sfcadm\(1M\)](#) command that re-creates the information in the management partition for that file system, are output.

OPERANDS

The following operands can be specified:

backupfile

Specifies the name of the output destination file. When the output destination file already exists, backup is not performed.

EXAMPLES

When a backup of the entire file system is to be output to file sfccfg.bkup.1.

```
# sfcgetconf sfccfg.bkup.1
```

When the contents of the **sfcsrm.conf** file are to be recorded at the head of the backup file, and the backup for the entire file system is to be output to sfccfg.bkup.2.

```
# sfcgetconf -a sfccfg.bkup.2
```

When backups of the file systems corresponding to *fsid* values 4, 5, and 6 are to be output to sfccfg.bkup.3.

F.2 Management Commands Specific to the GFS Shared File System

```
# sfcgetconf -i 4 -i 5 -i 6 sfccfg.bkup.3
```

When backups of the file system corresponding to *fsid* value 4, and the settings for file system corresponding to `/dev/sfdsk/gfs/rdisk/volume1` are to be output to `sfccfg.bkup.4`.

```
# sfcgetconf -i 4 -d /dev/sfdsk/gfs/rdisk/volume1 sfccfg.bkup.4
```

FILE

`/var/opt/FJSVsfcs/sfcfsrm.conf`

Environment file of `sfcfrmd` daemon

SEE ALSO

[sfcadm\(1M\)](#), [sfcsetup\(1M\)](#).

NOTE

- To enable the use of this command, the self-node must be multi user mode.

F.2.10 sfchost(1M) Displays and changes a connection host name with MDS

SYNOPSIS

```
sfchost { block_special | mntpoint }  
sfchost -p hostname { block_special | mntpoint }  
sfchost -E -p hostname { block_special | mntpoint }  
sfchost -c hostname1 hostname1[:hostname2...] {block_special | mntpoint }
```

DESCRIPTION

sfchost displays and changes by the host name for the information about the communication course of AC and primary MDS which are used now, about the file system specified by operand *block_special* or *mntpoint*.

The display of the present connection host name is effective only during primary MDS operation of the target file system.

A change of a connection host name is made by specifying argument *hostname* to be **-p option**. Change of a connection host name is effective only during MDS operation of the target file system. If unmount and down recovery of MDS occur, change of a connection host name will become invalid.

When change of a connection host name goes wrong by the abnormalities in communication, the connection host name before change is used as it is.

-c option is specified in order to change the use priority of a connection host name. It is possible only when the specified file system has stopped. AC chooses the host name which can be used in an order from the inside of the host name specified at the time of starting, and connects it to primary MDS. The host name in the head of a connection host name list cannot be changed. In order to change a top host, [sfcadm\(1M\)](#) is used.

OPTIONS

The following options can be specified:

-c

A connection host name list is changed. It is possible only when the specified file system has stopped. About the connection host name list to the host name specified by the argument *hostname1*, it changes into the connection host name list specified by *hostname1:hostname2:hostname3...* AC chooses the host name which can be used in order of *hostname1:hostname2:hostname3* at the time of starting. A maximum of four entries can be specified.

-E

Connection is tried by the host name in the connection host name list of other than *hostname*. Connection by the specified host name is tried in the *hostname* specification which does not specify **-E** option.

-p

The communication information on primary MDS are set up.

OPERANDS

The following operands can be specified:

hostname

A connection host name is changed into operand *hostname*. One of the connection host name list set up by [mkfs_sfcfs\(1M\)](#), [sfcadm\(1M\)](#), [sfcnewfs\(1M\)](#), or [sfcnode\(1M\)](#) is specified to be *hostname*. It changes into one other than *hostname* in a connection host name list at the time of **-E** option specification.

Specify whether **-p** option specification is carried out and primary MDS connection host name is

changed.

block_special

The block special device of the file system under target employment is specified. The representation partition of a file system is specified.

mountpoint

The mount point of the file system under target employment is specified.

EXAMPLES

When a mount point refers to a connection host name with primary MDS to the file system which is /shared_work.

```
# sfchost /shared_work
primary: hostA-1
```

When a mount point changes the connection host name to primary MDS into hostA-1 to the file system which is /shared_work.

```
# sfchost -p hostA-1 /shared_work
```

When a mount point changes the connection host name to primary MDS to the file system which is /shared_work in addition to hostA-1.

```
# sfchost -E -p hostA-1 /shared_work
```

SEE ALSO

[mkfs_sfcfs\(1M\)](#), [sfcadm\(1M\)](#), [sfcinfo\(1M\)](#), [sfcnewfs\(1M\)](#), [sfcnode\(1M\)](#), [sfcsetup\(1M\)](#).

F.2.11 sfcinfo(1M) Displays partition information

SYNOPSIS

```

sfcinfo [-V] [-n|-N] {block_special | mount_point} [ {block_special | mount_point} ...]
sfcinfo [-V] [-n|-N] -d dev[,dev,...]
sfcinfo -a [-V][-n|-N]
sfcinfo -q {block_special | mount_point} [ {block_special | mount_point} ...]
sfcinfo -q [-a]
sfcinfo -e {block_special | mount_point}
sfcinfo -e [-a]
sfcinfo -c {block_special | mount_point} [ {block_special | mount_point} ...]
sfcinfo -c [-a]

```

DESCRIPTION

sfcinfo displays the partition information, the MDS (metadata server) placing information, and the quota function setting status of the GFS Shared File System specified by *block_special* or *mount_point*.

block_special specifies one partition of the partitions making up the file system as a block special file name. The first partition displayed in the output results is the representative partition of the file system.

mount_point specifies the path name of the mount point of the file system. To specify the mount point path name, the file system must already be specified in **/etc/vfstab**.

If neither *block_special* nor *mount_point* is specified and the **-a** option is omitted, **sfcinfo** displays the information on all GFS Shared File Systems being mounted in the command execution node.

The GFS Shared File System retains partition information in each partition. The partition information relates to the partition group making up the file system. **sfcinfo** outputs the following detailed information:

[Partition information]

- FS identification number
- Special file name (device number)
- Partition size (in kilobytes)
- 0 while configuration change is in progress
- Usage of partitions
- (META: meta-data area | DATA: file data area | LOG: update log area)**
- Mount point in a command execution host
- "-----" when no file system is mounted
- Partition processing-ongoing status
- TRANSIT (add)** indicates ongoing partition addition.

[Node information]

- FS identification number
- Host ID
- Start status (**RUN** or **STOP**)
- Host name
- Plurality at the time of a LAN setup, it is a host name after the 2nd. It is displayed at the time of **-N** specification, using a blank as a punctuation mark.

[quota setting information]

- FS identification number
- Special file name of the representative partition
- quota function setting status (**on**, **off**, **on-off**, or **off-on**)
- Mount point in a command execution host

[MDS placing information]

- Special file name of the representative partition.
- Host names of the placing MDS (meta-data server)
- The setup which placing MDS by all nodes

OPTIONS

The following options can be specified:

- V**
Displays the version of the GFS Shared File System.
- a**
Displays the information for all file systems described in the file system configure information in a management partition.
- c**
Displays the host names of placing MDS (meta-data server).
- d *dev[,dev,...]***
Specifies a device number in hexadecimal for the partition to be displayed.
Use a comma delimiter to specify multiple device numbers.
- e**
Displays file system tuning information set by [sfcadm\(1M\)](#).
- n**
Displays node information.
If no parameters are specified, the information for all mounted file systems is displayed.
- N**
Plurality node information including a LAN setup is displayed. When an argument has not been set up, the information all mounted file systems are displayed.
- q**
Displays the setting status of the current quota function. **on** indicates that the quota is valid. **off** indicates that the quota is invalid. **off-on** indicates that the function setting is being changed by [sfcquotaon\(1M\)](#). **on-off** indicates that the function setting is being changed by [sfcquotaoff\(1M\)](#).

OPERAND

The following operand can be specified:

block_special | mount_point

Specifies a block-special device for the representative partition or the mount point of the file system.

EXAMPLES

```
# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special                size Type mount
 1 /dev/sfdsk/gfs/dsk/volume1(3f40021)  711874 META /mnt
 1 /dev/sfdsk/gfs/dsk/volume1(3f40021)  6578704 DATA /mnt
 1 /dev/sfdsk/gfs/dsk/volume1(3f40021)  51200 LOG /mnt

# sfcinfo -V /dev/sfdsk/gfs/dsk/volume1
sfdfs v1.1 file system
/dev/sfdsk/gfs/dsk/volume1:
FSID special                size Type mount
 1 /dev/sfdsk/gfs/dsk/volume1(3f40021)  711874 META /mnt
 1 /dev/sfdsk/gfs/dsk/volume1(3f40021)  6578704 DATA /mnt
 1 /dev/sfdsk/gfs/dsk/volume1(3f40021)  51200 LOG /mnt

# sfcinfo /mnt
/dev/sfdsk/gfs/dsk/volume1:
FSID special                size Type mount
 1 /dev/sfdsk/gfs/dsk/volume1(3f40021)  711874 META /mnt
 1 /dev/sfdsk/gfs/dsk/volume1(3f40021)  6578704 DATA /mnt
```



```

1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 51200 LOG /mnt

# sfcinfo -d 3f40021
/dev/sfdsk/gfs/dsk/volume1:
FSID special size Type mount
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 711874 META /mnt
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 6578704 DATA /mnt
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 51200 LOG /mnt

# sfcinfo -n /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID hostID status hostname
1 809f3901 RUN prime01
2 80a75644 STOP prime02

# sfcinfo -a
/dev/sfdsk/gfs/dsk/volume1:
FSID special size Type mount
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 711874 META ----
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 6578704 DATA ----
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 51200 LOG ----

/dev/sfdsk/gfs/dsk/volume2:
FSID special size Type mount
2 /dev/sfdsk/gfs/dsk/volume2(3f40022) 25418 META /mnt2
2 /dev/sfdsk/gfs/dsk/volume2(3f40022) 232256 DATA /mnt2
2 /dev/sfdsk/gfs/dsk/volume2(3f40023) 5120 LOG /mnt2
2 /dev/sfdsk/gfs/dsk/volume3(3f40023) 262794 META /mnt2
2 /dev/sfdsk/gfs/dsk/volume4(3f40024) 7341778 DATA /mnt2

```

Example of displaying the quota status

```

# sfcinfo -q /dev/sfdsk/gfs/dsk/volume1
FSID special quota mount
1 /dev/sfdsk/gfs/dsk/volume1 off ----

# sfcinfo -q /mnt
FSID special quota mount
2 /dev/sfdsk/gfs/dsk/volume2 on /mnt

# sfcinfo -q -a
FSID special quota mount
1 /dev/sfdsk/gfs/dsk/volume1 off ----
2 /dev/sfdsk/gfs/dsk/volume2 on /mnt

```

Example of display when a file data partition is being added

```

# sfcinfo /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID special size Type mount
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 711874 META /mnt
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 6578704 DATA /mnt
1 /dev/sfdsk/gfs/dsk/volume1(3f40021) 51200 LOG /mnt
1 /dev/sfdsk/gfs/dsk/volume7( ---- ) 0 DATA /mnt TRANSIT(add)

```

Example of displaying the MDS placing information

```
# sfcinfo -c /dev/sfdsk/gfs/dsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
node=node1,node2,node3,node4
mdsnode=all

# sfcinfo -c /dev/sfdsk/gfs/dsk/volume2
/dev/sfdsk/gfs/dsk/volume2:
node=node1,node2,node3
```

EXIT STATUS

- 0 Normal exit.
- 1 An argument is not right.
- 33 The specified special file is not the GFS Shared File System.

FILES

/etc/mnttab

Mounted file system table

/etc/vfstab

List of default parameters for each file system

SEE ALSO

[sfcadd\(1M\)](#), [sfcadm\(1M\)](#), [sfcquotaon\(1M\)](#).

[mnttab\(4\)](#), [vfstab\(4\)](#) in the "Solaris X Reference Manual Collection."

NOTES

- If the partition status is the processing-ongoing status (**TRANSIT**) and change processing terminates abnormally, an appropriate response to that status is required. For details, see [sfcadd\(1M\)](#).
- A response is required if the quota status is **off-on** or **on-off** and [sfcquotaon\(1M\)](#) or [sfcquotaoff\(1M\)](#) is not operating. For details, see [sfcquotaon\(1M\)](#).

F.2.12 **sfcmbkup(1M)**, **sfcmrstr(1M)** Generates a shell script for backing up

SYNOPSIS

```
sfcmbkup [ -a ] [ -c ] [ -b n ] [ -o filename ] special | mount_point backup_dev
sfcmbkup -e [ -c ] [ -b n ] special | mount_point
sfcmrstr [ -a ] [ -c ] [ -b n ] [ -o filename ] special | mount_point backup_dev
```

DESCRIPTION

The **sfcmbkup** command generates shell scripts for backing up the GFS Shared File System specified by the representative device name specified for *special* or a mount point specified for *mount_point*. The command generates two types of scripts: a script for backing-up and a script for restoring. The script for backing-up script creates a backup of each every volume using **dd(1M)**. The script for restoring restores each volume using **sfcddr(1M)**. The **sfcmrstr** command generates only a script for restoring.

These commands and the scripts generated by them must be executed with super user authority (root).

OPTIONS

The following options can be specified:

- a**
Specifies one medium to which the backup output is linked. This option requires a no-rewind device to be specified as the *backup_dev* parameter.
- c**
Generates a script by which data is compressed with **compress(1)** during backing-up and data is uncompressed during restoring.
- b n**
Specifies the block size of the backup medium as *n* bytes. A character string (value) specified for this parameter is supplied as is to **dd(1M)** and to **sfcddr(1M)** as **obs** for backing-up and **ibs** for restoring. For **ibs** in a **dd(1M)** command line, **k** or **b** can be appended to the value or multiplication can be specified with **x**. The default is **256k**.
- o filename**
Specifies the file name of the shell script to be generated. A script for backing-up is generated with the path name of the specified file name, which has a ".bkup" extension. A script for restoring is generated with the path name of the specified file name, which has a ".rstr" extension. The default is "sfcs."
- e**
Displays the space required for backing-up. Does not generate a shell script.
If the **-c** option is specified concurrently with this option, the command displays the space required for the file system with the data compressed. For a large file system, some time may elapse before the required space is displayed.

OPERANDS

The following operands can be specified:

- special**
Specifies a special file name for the representative partition of the GFS Shared File System.
- mount_point**
Specifies a mount point for the GFS Shared File System.
- backup_dev**
Specifies the device to which the backup data is transferred.

EXAMPLES

To back up the GFS Shared File System whose representative partition is the special file

F.2 Management Commands Specific to the GFS Shared File System

/dev/sfdsk/gfs/dsk/volume1 to a tape device, execute the following command.

```
# sfcmbkup /dev/sfdsk/gfs/dsk/volume1 /dev/rmt/0
```

When the command is executed, a script for backing up with the file name sfcfs.bkup (see below) is generated in the current working directory.

```
#!/bin/sh
IBS=1024k
OBS=256k
TAPE=/dev/rmt/0
SPECS="/dev/sfdsk/gfs/rdsk/volume1 /dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/gfs/rdsk/volume3"

n=`echo $SPECS | /usr/bin/wc -w`
echo "It will need $n volume(s) to backup."

i=1
for s in $SPECS
do
    echo "Backup $s"
    echo "Prepare volume #${i}/${n} for $TAPE and press RETURN:\c"
    read yes
    until /usr/bin/dd if=$s of=$TAPE ibs=$IBS obs=$OBS
    do
        echo "Press RETURN and retry:\c"
        read yes
    done
    i = `usr/bin/expr $i + 1`
done

echo "Backup done."
```

A script for restoring with the file name sfcfs.rstr is generated as shown below:

```
#!/bin/sh
IBS=256k
OBS=1024k
TAPE=/dev/rmt/0
SPECS="/dev/sfdsk/gfs/rdsk/volume1 /dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/gfs/rdsk/volume3"

n=`echo $SPECS | /usr/bin/wc -w`
echo "It will need $n volume(s) to restore."

i=1
for s in $SPECS
do
    if echo $s | /usr/bin/grep "^!"; then
        s=`echo $s | /usr/bin/cut -c2-`
        opt="-f"
    else
        opt=""
    fi
    echo "Restore $s"
    echo "Prepare volume #${i}/${n} for $TAPE and press RETURN:\c"
    read yes
    until /usr/lib/fs/sfcfs/sfcddr $opt if=$TAPE of=$s ibs=$IBS obs=$OBS
    do
        echo "Change to CORRECT volume #${i}/${n} and press RETURN:\c"
        read yes
    done
done
```

```
done
i = `usr/bin/expr $i + 1`
done

echo "Restore done."
```

If multiple tape drives are used for backing-up and restoring, execute the **sfcmbkup** command once for each tape drive. Next, edit the setting of shell script variable **SPECS** to assign the partitions to their respective shell scripts.

```
# sfcmbkup -o ex1 /dev/sfdsk/gfs/dsk/volume1 /dev/rmt/0
# sfcmbkup -o ex2 /dev/sfdsk/gfs/dsk/volume1 /dev/rmt/1
```

To restore partitions to a disk that is different from the disk from which the backup was made, edit the setting of variable **SPECS** in the script for restoring. Type an exclamation point (!) before the edited special file name to indicate that a volume information check is not to be executed.

```
SPECS="/dev/sfdsk/gfs/rdsk/volume1 ! /dev/sfdsk/gfs/rdsk/volume2 /dev/sfdsk/gfs/rdsk/volume3"
```

SEE ALSO

[sfcadm\(1M\)](#), [sfcddr\(1M\)](#).

compress(1) and **dd (1M)** in the *"Solaris X Reference Manual Collection."*

NOTES

- If the backed-up data is restored to a disk other than the one from which the backup was made, the structure of the disk on which the data is restored must be the same as that of the original disk.
- Do not specify a partition that includes the first cylinder of the disk as the partition to which another partition data is restored. If this is done by mistake, the disk's configuration information will be destroyed and the disk will not be able to be used in the current configuration.
- Once the backed-up data has been restored into to disk other than the original one, the partition information changed with [sfcadm\(1M\)](#). Specify all partitions in the GFS Shared File System and execute [sfcadm\(1M\)](#).

F.2.13 sfcmntgl(1M) Mounts the GFS Shared File System on all nodes

SYNOPSIS

```
sfcmntgl special | mount_point  
sfcmntgl special mount_point
```

DESCRIPTION

The **sfcmntgl** mounts all nodes sharing the same file system according to the specified operands.

The **sfcmntgl** can be executed from any node which are specified in the GFS configuration node data made by [sfcsetup\(1M\)](#). It is necessary to set up `/etc/vfstab`, in advance.

Only a super user can perform mount processing.

When mount processing goes wrong, error information is outputted to the console of each node. Please perform management according to the outputted error information.

OPERAND

The following operand can be specified:

special

Specifies the block special device name of a representative partition of the file system to be mounted.

mount_point

Specifies the name of the directory to be mounted.

EXAMPLE

```
# sfcmntgl /dev/sfdsk/gfs/dsk/volume1 /mount_point  
# sfcmntgl /dev/sfdsk/gfs/dsk/volume1  
# sfcmntgl /mount_point
```

FILES

/etc/mnttab

Table for the currently mounted file system.

/etc/vfstab

Environment file for the file system.

SEE ALSO

[mount_sfdfs\(1M\)](#), [sfcsetup\(1M\)](#), [sfcumntgl\(1M\)](#), [umount_sfdfs\(1M\)](#).

mount(1M), **mount(4)**, **vfstab(4)** in the "*Solaris X Reference Manual Collection*."

NOTES

- `sfcfrmd` daemon must be running to enable the execution of this command.
- When both *special* and *mount_point* are specified, they must also be described in ***/etc/vfstab*** in all node which shared a file system.
- Mount processing always is performed using the mount options for the mount point described in ***/etc/vfstab***. When no mount options are described in ***/etc/vfstab***, mount processing is performed without any options.

F.2.14 sfcnewfs(1M) Builds the GFS Shared File System

SYNOPSIS

```
sfcnewfs [-v] [-N] [mkfs-options] [raw_device_file]
sfcnewfs -C file
```

DESCRIPTION

The **sfcnewfs** command is a [mkfs_sfcfs\(1M\)](#) front end for building a GFS Shared File System. The **sfcnewfs** command calculates the appropriate parameters and calls [mkfs_sfcfs\(1M\)](#).

Interactive execution of **sfcnewfs** from a standard input terminal enables information about the file system to be confirmed before it is created.

OPTIONS

The following options can be specified:

-v

Redundant mode. Displays operations, including the passing of parameters to [mkfs_sfcfs\(1M\)](#).

-N

Displays the parameters to be used for creating the file system.
The file system is not actually created.

-C file

Specifies the file containing the parameters that be used for creating the file system. When this option is used, a *raw_device_file* parameter is not required. Other options cannot be specified. For the syntax of the parameter file, see the example.

mkfs-options

These options are used to change the default parameters passed to [mkfs_sfcfs\(1M\)](#).

-A

Specify this option so that the file system is configured with a representative partition that has no file data area or other partitions if a file data partition or partitions are specified with the **-D** option.

When the **-D** option is not specified, the **-A** option is invalid if specified.

-D raw_device_file,...

Specifies the character-special devices for a file data partition. Use a comma delimiter to specify multiple files.

-H host[:host],...

Registers the name of a host that shares the file system. If multiple nodes share the file system, use a comma delimiter to specify multiple hosts.

Hosts whose names are to be specified with *host* must be able to communicate with each other. MDS (meta-data server) placing nodes are selected from head of the host names specified with **-H** option. The numbers of MDS placing nodes are specified by the **-S** option.

If this option is not specified, **one** host name, which is that of the local node, is registered.

A maximum of **four** entries can be specified.

Use a colon delimiter to specify multiple host names. The multiplexed network path is used for communication path with AC and MDS, The host name of each node can be specified to **four** maximums.

-L raw_device_file

Specifies the character-special devices for the update log partition.

The portion of the update log partition not used, as an update log area becomes an unused area. Specify a partition that is the same size as the update log area.

-S all|num

Specifies the number of MDS placing nodes. MDS placing nodes are selected from head of the host names specified with **-H** option. When **all** is specified, all nodes are set up as MDS placing node. If and only if **all** is specified, the node added to node information by [sfcnode\(1M\)](#)

after constructing the file system is also set up as a MDS placing node.

The default is **2**.

0 or less value and value bigger than the number of nodes specified with **-H** option cannot be specified.

-Z

Specify the **-Z** option to re-create a file system previously registered in the file system configure information in a management partition with the same configuration. If, however, the target file system is being mounted, specifying this option results in an error.

-a metasz

Specifies the size of the meta-data area. The unit is megabytes.

The default is about **10%** of the file system size. When *maxdsz* has been specified, the default is set at about **10%** of the *maxdsz* value.

A size greater than the size of the representative partition cannot be specified.

-c

A meta-data area is initialized altogether.

-d maxdsz

Specifies the maximum size of the total file data area of the file system, including additional file data partitions. The unit is megabytes. Based on the value specified for this parameter, the size of the meta-data and the update log area size are set to appropriate values that take into account additional file data partitions.

The default is the size of the file system containing the partition or partitions specified less the meta-data area size and the update log area size.

A value greater than **32** terabyte (**33,554,432** megabytes) cannot be specified.

-e nblkpext

Specifies the average number of blocks that are managed by one extent. This parameter is used as a fragmentation factor for the file data area.

The default is **1**.

A value can be specified in the range from **1** to **100**.

-f free

Specifies the minimum percentage of free area for the file system.

The default is **10%**.

A value can be specified in the range from **0** to **99**.

-g logsz

Specifies the size of the update log area. The unit is megabytes.

The default is about **1%** of the file system size. When *maxdsz* has been specified, the default is about **1%** of the *maxdsz* value. If the default value is smaller than **5** megabytes, **5** megabytes is set. If the default value is greater than **50** megabytes, **50** megabytes is set.

A value can be specified in the range from **5** to **100**.

-i nbpi

Specifies the byte count per i-node. Specify a larger value for this parameter if the meta-data area is to be saved.

The default is **8192**.

A value can be specified in the range from **512** to **1,048,576**.

-l maxvol

Specifies the maximum number of partitions that can make a file system built on disks. The value should include any partitions added in a future expansion with [sfcadd\(1M\)](#).

The default is **16**.

A value can be specified in the range from **1** to **32**.

-m mfree

Specifies the minimum percentage of the free V-data area.

The default is **10%**.

A value can be specified in the range from **0** to **99**.

-n maxnode

Specifies the maximum number of nodes that share a file system built on disks. The value should include any partitions added in a future expansion with [sfcnnode\(1M\)](#).

The default is **16**.

A value can be specified in the range from **1** to **32**.

-o space|time

Selects either space or time on which to base the optimization of free block allocation. **space** indicates space, and **time** indicates time. The default is **time**.

OPERAND

The following operand can be specified:

raw_device_file

Specifies a character-special device for the representative partition. A device specified for a file data partition or the update log partition cannot be specified for the representative partition.

EXAMPLES

```
# sfcnewfs -H host1,host2 /dev/sfdsk/gfs/rdisk/volume1

# sfcnewfs -H host1,host2 -L /dev/sfdsk/gfs/rdisk/volume2 /dev/sfdsk/gfs/rdisk/volume1

# sfcnewfs -H host1,host2 -D /dev/sfdsk/gfs/rdisk/volume3, /dev/sfdsk/gfs/rdisk/volume4 /dev/sfdsk/gfs/rdisk/volume2

# sfcnewfs -S all -H host1,host2,host3,host4 /dev/sfdsk/gfs/rdisk/volume2

# sfcnewfs -S 3 -H host1,host2,host3,host4 /dev/sfdsk/gfs/rdisk/volume2

# cat paramfile
meta: /dev/sfdsk/gfs/rdisk/volume1      # representative partition
log : /dev/sfdsk/gfs/rdisk/volume2      # -L option argument
data: /dev/sfdsk/gfs/rdisk/volume3      # -D option argument
data: /dev/sfdsk/gfs/rdisk/volume4      # -D option argument
logsz: 10                               # -g option argument
free: 10                                # -f option argument
node: host1 host2                       # -H option argument
mdsnode: all                            # -S option argument
# sfcnewfs -v -C paramfile
```

EXIT STATUS

- 0 The GFS Shared File System has been created correctly.
 - 1 An abnormal event during the execution of **sfcnewfs** has been detected.
 - 10 Inappropriate options specified for **sfcnewfs** have been detected.
- Other than above an abnormal event during the execution of **mkfs_sfcs(1M)** has been detected.

SEE ALSO

[fsck_sfcs\(1M\)](#), [mkfs_sfcs\(1M\)](#), [sfcadd\(1M\)](#), [sfcnode\(1M\)](#), [sfcsetup\(1M\)](#), [sftune\(1M\)](#).

NOTES

- *host* is necessary to be the host name of the node which constitutes the GFS common file system registered into the management partition by [sfcsetup\(1M\)](#).
- A file system can be created only when all hosts that share the file system have started operation and GFS Shared File System has been installed. The reason is that the information for these hosts is required.
- In the GFS Shared File System, the data block size is fixed at **8192** bytes and cannot be altered.
- To re-create the GFS Shared File System by modifying its configuration, first delete the file system with [sfcadm\(1M\)](#).
- The file system configuration must be MDS can place in all nodes, if building the file system (if

F.2 Management Commands Specific to the GFS Shared File System

- possible).
- The I/O performance of the application has influence while MDS perform meta-data area initialization after the file system start. Specify a **-c** option to avoid this influence.

F.2.15 sfcnode(1M) Adds, deletes, and alters node configuration information

SYNOPSIS

```
sfcnode -a host[:host],... raw_device_file
sfcnode -d [-h host,... | -i hostid,...] raw_device_file
```

DESCRIPTION

The **sfcnode** command adds, deletes, and changes GFS Shared File System node information.

To add or update node information, specify the **-a** option. If a specified host name already exists, **sfcnode** updates node and device information for the host. If the host does not exist, **sfcnode** adds the node and device information. Use a comma delimiter to specify multiple host names.

To delete node information, specify the **-d** option. For the **-h** or **-i** option, specify the host name (*host*) or host ID (*hostid*) of the host to be deleted. **sfcnode** deletes the node information of the host. If neither the **-h** nor **-i** option is specified, **sfcnode** deletes the node information for the host on which the **sfcnode** is executed. Use a comma delimiter to specify multiple host names or IDs. The **-h** and **-i** options cannot be specified together.

After the **sfcnode** command changes the GFS Shared File System node information, the entries in the file system configure information in a management partition are updated. The information retained in the partitions is also changed.

OPTIONS

The following options can be specified:

-a *host[:host],...*

Specifies the name of a host whose node information is to be added.

Use a colon delimiter to specify multiple host names. The multiplexed network path is used for communication path with AC and MDS, The host name of each node can be specified to **four** maximums.

-d

Deletes node information.

-h *host*,...

Specifies the name of a host whose node information is to be deleted. The host must be one that can be used by the GFS Shared File System.

-i *hostid*,...

Specifies the ID of a host whose node information is to be deleted. The host must be one that can be used by the GFS Shared File System.

OPERAND

The following operand can be specified:

raw_device_file

Specify a character-special device for the representative partition of the GFS Shared File System whose node information is to be updated.

EXAMPLES

Add new node that was added by **sfcsetup**(1M) to the sharing nodes of GFS Shared File System.

```
# sfcnode -a host1,host2 /dev/sfdsk/gfs/rdsk/volume1
```

Add or update node/device information for hosts by specifying host names.

```
# sfcnode -a host1, host2 /dev/sfdsk/gfs/rdsk/volume1
```

Delete node/device information for hosts by specifying host names.

```
# sfcnode -d -h host1,host2 /dev/sfdsk/gfs/rdsk/volume1
```

F.2 Management Commands Specific to the GFS Shared File System

Delete node/device information for a host by specifying its ID.

```
# sfcnode -d -i 80000000 /dev/sfdsk/gfs/rdisk/volume1
```

Delete node/device information for the local host.

```
# sfcnode -d /dev/sfdsk/gfs/rdisk/volume1
```

SEE ALSO

[mkfs_sfcfs\(1M\)](#), [sfcadd\(1M\)](#), [sfcadm\(1M\)](#), [sfcinfo\(1M\)](#), [sfcnewfs\(1M\)](#), [sfcsetup\(1M\)](#).

NOTES

- A file system can be changed only when all hosts that share the file system have started operation and GFS Shared File System has been installed. The reason is that the information for these hosts is required.
- *host* is necessary to be the host name of the node which constitutes the GFS common file system registered into the management partition by [sfcsetup\(1M\)](#).
- If **sfcnode** command output message "node full, can't sfcnode" and failed, exceeded the maximum number of sharing nodes when making GFS Shared File System by [sfcnewfs\(1M\)](#). Check registered node information by [sfcinfo\(1M\)](#) , and then add node after delete no need node information by **sfcnode -d** option.
- Cannot delete node information of host that placing primary MDS, and secondary MDS.

F.2.16 sfcproxybreak(1M) Cancels the relationship for high speed backup and restoration

SYNOPSIS

sfcproxybreak [-f] *rel_name*

DESCRIPTION

sfcproxybreak cancels the relationship between the backup source and destination for high-speed backup and restoration. By default, this command cannot cancel the relationship or fails when a real copy is being executed.

When -f option is specified, it cancels the relationship even when a real copy is being executed. But when a real copy is terminated by specifying -f option, the status of destination volume of GDS becomes data invalid (INVALID) status.

OPTION

The following option can be specified:

-f

Forcible specification. This option cancels the relationship even when a real copy is being executed between the file systems whose relationship is to be canceled.

OPERAND

The following operand can be specified:

rel_name

This command specifies the relationship to be canceled by using the relationship name *rel_name*

EXAMPLES

To cancel the relationship with the relationship name backup_for_volume0001, execute the following command.

```
# sfcproxybreak backup1_for_volume0001
```

SEE ALSO

[sfcadm\(1M\)](#), [sfcproxyinfo\(1M\)](#), [sfcproxyjoin\(1M\)](#), [sfcproxyop\(1M\)](#), [sfcproxyrestore\(1M\)](#)

NOTES

- **sfcproxybreak** does not delete the file system. When the file system of the backup destination is not required, delete file system with [sfcadm\(1M\)](#).
- Optional product is required for this function.

F.2.17 `sfcproxyinfo(1M)` Displays information about the relationship for high speed backup and restoration

SYNOPSIS

```
sfcproxyinfo  
sfcproxyinfo [ -f ] rel_name  
sfcproxyinfo -d raw_device_file
```

DESCRIPTION

sfcproxyinfo displays the information about the relationship between the backup source and destination for high-speed backup and restoration.

This command displays the following information:

- relation name:
The relationship name set up by [sfcproxyjoin\(1M\)](#) is displayed.
- type:
The type of operation pattern is displayed. "OPC" is displayed for the OPC operation, or "EC" is displayed for the EC operation.
- status:
join status, copy status and error status are displayed. "SEPARATE" is displayed for the OPC operation or for the separate status in the EC operation, or "JOIN" is displayed for the join status in the EC operation. "COPY" is displayed when a real copy is being executed subsequently. Or "INVALID" is displayed when a real copy failed.
- partition correspondence and copy status of copy source and destination
Backup source partition is displayed in the "master" item, and backup destination partition correspond to the source partition is displayed in the "proxy" item. The status of backup source partition is displayed in the "m_status" item, and the status of backup source partition is displayed in the "p_status" item; "COPY" is displayed in the status of copy destination partition when a real copy is being executed. "INVALID" is displayed in the status of copy destination partition when a read copy failed. Otherwise, "DONE" is displayed.

When no argument is specified, this command displays information of all relationships.

When a relationship name is specified for an argument, this command displays information on the relationship.

When a character special device is specified in the **-d** option, this command displays information on all relationships that include such character special device in the backup source or destination.

OPTIONS

The following options can be specified:

- d *raw_device_file*
This option displays information on all relationships that include such character special device *raw_device_file* in the backup source or destination.
- f
This option displays information only for partition correspondence in the format that can be used in case **-f** option of [sfcproxyjoin\(1M\)](#). This option can be specified only when *rel_name* is specified.

OPERAND

The following operand can be specified:

rel_name

This operand displays information on the relationship *rel_name*.

EXAMPLES

(Example 1)

This example displays information on the relationship backup1_for_volume0001. When a real copy is completed in some partition during backup in the OPC operation, the following is displayed.

```
# sfcproxyinfo backup1_for_volume0001
relation name: backup1_for_volume0001
type: OPC
status: SEPARATE, COPY
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdisk/volume0001 /dev/sfdsk/class0001/rdisk/volume0004 DONE COPY
/dev/sfdsk/class0001/rdisk/volume0002 /dev/sfdsk/class0001/rdisk/volume0005 DONE DONE
/dev/sfdsk/class0001/rdisk/volume0003 /dev/sfdsk/class0001/rdisk/volume0006 DONE COPY
```

(Example 2)

This example displays information on the relationship backup1_for_volume0001. When an equivalent copy is being executed after [sfcproxyjoin](#) or [sfcproxyop rejoin](#) in the EC operation. The following is displayed.

```
# sfcproxyinfo backup1_for_volume0001
relation name: backup1_for_volume0001
type: EC
status: JOIN, COPY
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdisk/volume0001 /dev/sfdsk/class0001/rdisk/volume0004 DONE COPY
/dev/sfdsk/class0001/rdisk/volume0002 /dev/sfdsk/class0001/rdisk/volume0005 DONE DONE
/dev/sfdsk/class0001/rdisk/volume0003 /dev/sfdsk/class0001/rdisk/volume0006 DONE DONE
```

(Example 3)

This example displays information on the relationship backup1_for_volume0001. When an equivalent copy is being executed after [sfcproxyrestore](#) in the EC operation. The following is displayed.

```
# sfcproxyinfo backup1_for_volume0001
relation name: backup1_for_volume0001
type: EC
status: JOIN, COPY
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdisk/volume0001 /dev/sfdsk/class0001/rdisk/volume0004 DONE DONE
/dev/sfdsk/class0001/rdisk/volume0002 /dev/sfdsk/class0001/rdisk/volume0005 COPY DONE
/dev/sfdsk/class0001/rdisk/volume0003 /dev/sfdsk/class0001/rdisk/volume0006 COPY DONE
```

(Example 4)

This example displays information on the relationship backup1_for_volume0001. When a real copy is not executed in the separate status in the EC operation, the following is displayed.

```
# sfcproxyinfo backup1_for_volume0001
relation name: backup1_for_volume0001
type: EC
status: SEPARATE
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdisk/volume0001 /dev/sfdsk/class0001/rdisk/volume0004 DONE DONE
/dev/sfdsk/class0001/rdisk/volume0002 /dev/sfdsk/class0001/rdisk/volume0005 DONE DONE
/dev/sfdsk/class0001/rdisk/volume0003 /dev/sfdsk/class0001/rdisk/volume0006 DONE DONE
```

F.2 Management Commands Specific to the GFS Shared File System

(Example 5)

This example displays information on all relationships that include the partition /dev/sfdsk/class0001/rdsk/volume0001 in the backup source or destination.

```
# sfcproxyinfo -d /dev/sfdsk/class0001/rdsk/volume0001
relation name: backup1_for_volume0001
type: OPC
status: SEPARATE
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdsk/volume0001 /dev/sfdsk/class0001/rdsk/volume0004 DONE DONE
/dev/sfdsk/class0001/rdsk/volume0002 /dev/sfdsk/class0001/rdsk/volume0005 DONE DONE
/dev/sfdsk/class0001/rdsk/volume0003 /dev/sfdsk/class0001/rdsk/volume0006 DONE DONE
relation name: backup2_for_volume0001
type: EC
status: JOIN, COPY
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdsk/volume0001 /dev/sfdsk/class0001/rdsk/volume0007 DONE COPY
/dev/sfdsk/class0001/rdsk/volume0002 /dev/sfdsk/class0001/rdsk/volume0008 DONE DONE
/dev/sfdsk/class0001/rdsk/volume0003 /dev/sfdsk/class0001/rdsk/volume0009 DONE DONE
```

(Example 6)

This example displays information on all relationships.

```
# sfcproxyinfo
relation name: backup1_for_volume0001
type: OPC
status: SEPARATE
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdsk/volume0001 /dev/sfdsk/class0001/rdsk/volume0004 DONE DONE
/dev/sfdsk/class0001/rdsk/volume0002 /dev/sfdsk/class0001/rdsk/volume0005 DONE DONE
/dev/sfdsk/class0001/rdsk/volume0003 /dev/sfdsk/class0001/rdsk/volume0006 DONE DONE
relation name: backup2_for_volume0001
type: EC
status: JOIN, COPY
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdsk/volume0001 /dev/sfdsk/class0001/rdsk/volume0007 DONE COPY
/dev/sfdsk/class0001/rdsk/volume0002 /dev/sfdsk/class0001/rdsk/volume0008 DONE DONE
/dev/sfdsk/class0001/rdsk/volume0003 /dev/sfdsk/class0001/rdsk/volume0009 DONE DONE
relation name: backup1_for_volume0010
type: OPC
status: SEPARATE
master      proxy      m_status  p_status
/dev/sfdsk/class0002/rdsk/volume0010 /dev/sfdsk/class0002/rdsk/volume0012 DONE DONE
/dev/sfdsk/class0002/rdsk/volume0011 /dev/sfdsk/class0002/rdsk/volume0013 DONE DONE
```

(Example 7)

This example displays information on the relationship backup1_for_volume0001. When a real copy failed in some partition in the OPC operation, the following is displayed.

```
# sfcproxyinfo backup1_for_volume0001
relation name: backup1_for_volume0001
type: OPC
status: SEPARATE, INVALID
master      proxy      m_status  p_status
/dev/sfdsk/class0001/rdsk/volume0001 /dev/sfdsk/class0001/rdsk/volume0004 DONE INVALID
/dev/sfdsk/class0001/rdsk/volume0002 /dev/sfdsk/class0001/rdsk/volume0005 DONE INVALID
/dev/sfdsk/class0001/rdsk/volume0003 /dev/sfdsk/class0001/rdsk/volume0006 DONE DONE
```

F.2.17 sfcproxyinfo(1M) Displays information about the relationship for high speed backup and restoration

(Example 8)

This example displays information for partition correspondence on the relationship backup1_for_volume0001 in the format which can be used in case -f option of [sfcproxyjoin\(1M\)](#).

```
# sfcproxyinfo -f backup1_for_volume0001
/dev/sfdsk/class0001/rdisk/volume0001=/dev/sfdsk/class0001/rdisk/volume0004
/dev/sfdsk/class0001/rdisk/volume0002=/dev/sfdsk/class0001/rdisk/volume0005
/dev/sfdsk/class0001/rdisk/volume0003=/dev/sfdsk/class0001/rdisk/volume0006
```

SEE ALSO

[sfcproxybreak\(1M\)](#), [sfcproxyjoin\(1M\)](#), [sfcproxyop\(1M\)](#), [sfcproxyrestore\(1M\)](#)

NOTE

- Optional product is required for this function.

F.2.18 sfcproxyjoin(1M) Sets up the relationship for high speed backup and restoration

SYNOPSIS

```
sfcproxyjoin separate -r rel_name [ -s ] master_raw_device_file=proxy_raw_device_file,...  
sfcproxyjoin separate -r rel_name [ -s ] -f param_file  
sfcproxyjoin equiv[alent] -r rel_name [ -s ] master_raw_device_file=proxy_raw_device_file,...  
sfcproxyjoin equiv[alent] -r rel_name [ -s ] -f param_file
```

DESCRIPTION

sfcproxyjoin sets up the relationship between backup source and destination for high-speed backup and restoration.

When a relationship is set up with this command, the partitions of the backup destination are set up for the corresponding partitions that compose the GFS Shared File System of the backup source.

And a relationship for GDS Snapshot is also set up between the corresponding partitions. The backup source is set as master, and the backup destination is set as proxy. Consequently, conditions of a relationship for GDS Snapshot need to be satisfied between the corresponding partitions. Main conditions are shown below. Refer to the manual of PRIMECLUSTER Global Disk Services for details.

- Master and proxy belong to different group of same disk class of GDS.
- Volume being related as master is not related as proxy.
- Volume being related as proxy is not related as master or proxy.

The partitions of the backup source and destination need to satisfy the following conditions.

- The backup source and destination have the same partition configuration. The number of partitions of the backup source is the same as that of the backup destination, and the size of a partition of the backup source is the same as that of the corresponding partition of the backup destination.
- When the file system is already created at the backup destination, the configuration of MDS nodes and nodes that are sharing each file system for the backup source is the same as that of the backup destination.
- When the OPC operation is performed, the partitions (that correspond to one another) between the backup source and destination exist on the same cabinet of the Fujitsu RAID device ETERNUS3000/GR720/GR730/GR740.
- When the OPC operation is performed, partitions of backup source and destination are not mirrored by GDS.

OPC In the case of OPC operation, specify **separate** in the first argument. In this case, Although One Point Copy is performed from the backup source to the backup destination, the backup source and destination remain in separate status. The copy operation is performed logically instantaneously but file system update operation at the backup source is temporary locked while a copy command is issued. When the **-s** option is not specified, this command returns without waiting for the real copy processing to be completed.

In the case of EC operation, specify **equivalent** (or **equiv**) in the first argument. In this case, the backup source and destination enter the join status and copy from the backup source to the backup destination is started. When the **-s** option is not specified, this command returns without waiting for the real copy processing to be completed. The file system of the backup destination that entered the join status and the partitions that compose that file system are disabled until they are separated by using [sfcproxyop part](#), or the relationship is canceled by using [sfcproxybreak](#).

Specify a relationship name as the name that corresponds to the relationship. After the relationship is set

F.2.18 sfcproxyjoin(1M) Sets up the relationship for high speed backup and restoration

up, execute all operations such as backup, restoration and relationship cancellation with the relation name specified.

The following restrictions on setting up the relationship applies.

- Although two or more backup destinations can be related to one backup source, different operation pattern cannot be mixed. For example, the file system already related as backup source for OPC operation cannot be related as backup source for EC operation.

When relationship is set up, this command fails in the following cases.

- The case that backup source or destination volume of GDS is copy status.
In this case, execute this command after the copy is completed.
- The case that the status of backup source volume of GDS is data invalid (INVALID) status.
In this case, recover from INVALID status according to the manual of PRIMECLUSTER Global Disk Services.
- The case that the partitions of backup destination are activated on some nodes other than the node on which this command is executed.
In this case, inactivate the volumes by using **sdxvolume** of GDS on those nodes.

When a relationship is set up, a file system needs to have been created at the backup source but not at the backup destination. After the relationship is set up, processing up to the creation of file system resources is performed at the backup destination.

OPTIONS

The following options can be specified:

- r *rel_name*
Specify a character string of 255 or less characters as a name that is assigned to this relationship. However, these characters must not begin with "-".
- s
Synchronization specification. This option waits for the real copy processing to be completed.
- f *param_file*
Specify a file in which the correspondence between the backup source and destination is described. Describe the correspondence between partitions in the format "*master_raw_device=proxy_raw_device*" as in the case of the argument. For details, see the **OPERAND** items. Unlike the case of the argument, however, delimit each correspondence with a comma or line feed. When a line begins with "#", it is assumed to be a comment line, and content up to the end of that line is ignored.

OPERAND

The following operand can be specified:

- master_raw_device=proxy_raw_device,...*
Specify the partition to be a backup source in *master_raw_device* and the one to be a backup destination in *proxy_raw_device*, each as a character special device. These two partitions must be the same size.
When the file system of the backup source consists of two or more partitions, the partitions of the backup destination that correspond to all of these partitions need to be specified. To describe two or more correspondences, delimit them with a comma.

EXAMPLES

(Example 1)

Use a GFS Shared File System that consists of these two partitions as a backup source.

/dev/sfdsk/class0001/rdsk/volume0001

/dev/sfdsk/class0001/rdsk/volume0002

To relate each of the following partitions as a backup destination of this file system for the OPC operation,

/dev/sfdsk/class0001/rdsk/volume0003

/dev/sfdsk/class0001/rdsk/volume0004

Execute the following command.

```
# sfcproxyjoin separate -r backup1_for_volume0001
/dev/sfdsk/class0001/rdsk/volume0001=/dev/sfdsk/class0001/rdsk/volume0003 \
/dev/sfdsk/class0001/rdsk/volume0002=/dev/sfdsk/class0001/rdsk/volume0004
```

(Example 2)

Use a GFS Shared File System that consists of these two partitions as a backup source.

/dev/sfdsk/class0001/rdsk/volume0001

/dev/sfdsk/class0001/rdsk/volume0002

To relate each of the following partitions as a backup destination of this file system for the EC operation,

/dev/sfdsk/class0001/rdsk/volume0003

/dev/sfdsk/class0001/rdsk/volume0004

Execute the following command.

```
# sfcproxyjoin equivalent -r backup1_for_volume0001
/dev/sfdsk/class0001/rdsk/volume0001=/dev/sfdsk/class0001/rdsk/volume0003 \
/dev/sfdsk/class0001/rdsk/volume0002=/dev/sfdsk/class0001/rdsk/volume0004
```

SEE ALSO

[sfcproxybreak\(1M\)](#), [sfcproxyinfo\(1M\)](#), [sfcproxyop\(1M\)](#), [sfcproxyrestore\(1M\)](#)

NOTE

- Optional product is required for this function.

F.2.19 sfcproxyop(1M) Executes operation for high speed backup and restoration

SYNOPSIS

```
sfcproxyop backup [ -s ] rel_name  
sfcproxyop part rel_name  
sfcproxyop rejoin [ -s ] rel_name  
sfcproxyop mount rel_name mount_point  
sfcproxyop umount rel_name mount_point
```

DESCRIPTION

sfcproxyop executes operation for high-speed backup and restoration. Before this command is executed, the relationship between the backup source and destination must have been set up with the [sfcproxyjoin](#) command.

sfcproxyop includes the following subcommands:

- **backup** - Backup in OPC operation
- **part** - Snapshot creation by separation in EC operation
- **rejoin** - Rejoin of backup destination in EC operation
- **mount** - Mount the backup destination file system
- **umount** - Unmount the backup destination file system

The following explains the function of each subcommand.

[backup] - Backup in OPC operation

This subcommand executes One Point Copy of the file system by copy from the backup source to the backup destination specified by the relationship *rel_name* in the OPC operation. When this command is executed, the following conditions must be satisfied.

- The file system of the backup source must be unmounted in all nodes. Or, the file system must be mounted in the node on which this command is executed. If the file system is active and it is not mounted in the node on which this command is executed is inactive, the command fails.
- Real copy processing must not have been executed at the backup source or destination.
- The file system of the backup destination must be unmounted.
- The backup destination volumes of GDS must be stop status.
- The backup source volumes of GDS must not be data invalid (INVALID) status.

The copy operation is performed logically instantaneously but the file system update operation at the backup source is temporarily locked while a copy command is issued.

By default, this command returns without waiting for the real copy processing to be completed. When the -s option is specified, this command returns after the real copy processing is completed. Even in this case, the file system is locked only while the copy command is issued.

[part] - Snapshot creation by separation

In the EC operation, this subcommand instantaneously separates the backup from the backup source specified with the relationship *rel_name* and creates a file system snapshot at the backup destination. When this command is executed, the following conditions must have been satisfied.

- The file system of the backup source must be unmounted in all nodes. Or, the file system must be mounted in the node on which this command is executed. If the file system is active and it is not mounted in the node on which this command is executed is inactive, the command fails.

- The backup destination must have been joined in the backup source.

Although the separation operation is performed logically instantaneously, the file system update operation at the backup destination is temporarily locked while a separation command is issued.

After this command returns, the equivalent status with the backup source is cleared, making it possible to use the file system of the backup destination and the partitions that compose that file system. To return to the equivalent status after using the backup destination, use **rejoin** subcommand.

In the case of OPC operation, after restoration by using [sfcproxyrestore](#), this subcommand needs to be executed after copy processing is completed so that backup can be executed again.

[rejoin] - Rejoin of backup destination

In the EC operation, this subcommand starts copy from the backup source to the backup destination specified by the relationship *rel_name* and returns to the join status. When this subcommand is executed, the following conditions must be satisfied.

- Real copy processing must not have been executed at the backup source or destination.
- The file system of the backup destination must be unmounted.
- The file system of the backup source must have stopped its cluster service.
- The backup source and destination must be in separate status.
- The backup source volumes of GDS must not be data invalid (INVALID) status.

By default, this command returns without waiting for the real copy. When the **-s** option is specified, the command returns after the real copy processing is completed.

After this command returns, the backup destination becomes join status to the backup source. The file system of the backup destination and partitions that compose that file system cannot be used until the separation operation is performed with [sfcproxyop part](#) or the relationship is canceled with [sfcproxybreak](#).

[mount] - Mount the backup destination

Mount the GFS Shared File System created with **backup** subcommand or **part** subcommand at the *mount point* specified by *mount_point*.

In the EC operation, the backup source and destination must be in separate status with **part** subcommand.

To mount the file system created at the backup destination, use this subcommand all the time.

The following is different from normal file system and needs to be cared.

- The file system cannot be mounted on two or more nodes at the same time. However, it can be mounted on the node that is not defined as MDS node.

[umount] - Unmount the backup destination

Unmount the backup destination GFS Shared File System that mounted at the mount point *mount_point*.

OPTION

The following option can be specified:

-s

Synchronization specification. This option waits for the real copy processing to be completed.

OPERAND

The following operand can be specified:

rel_name

Specify an operation target relationship with the relationship name *rel_name*.

EXAMPLES

(Example 1)

To perform instant copy from the backup source to the backup destination specified with the relationship name `backup1_for_volume0001` in the OPC operation, execute the following command.

```
# sfcproxyop backup backup1_for_volume0001
```

(Example 2)

To separate the backup destination from the backup source specified with the relationship with the relationship name `backup1_for_volume0001` in the EC operation, execute the following command.

```
# sfcproxyop part backup1_for_volume0001
```

(Example 3)

To rejoin the backup destination specified with the relationship with the relationship name `backup1_for_volume0001` and put it in the equivalent status in the EC operation, execute the following command.

```
# sfcproxyop rejoin backup1_for_volume0001
```

(Example 4)

To mount the backup destination specified with the relationship with the relationship name `backup1_for_volume0001` at the mount point `/mnt`, execute the following command.

```
# sfcproxyop mount backup1_for_volume0001 /mnt
```

(Example 5)

To unmount the backup destination specified with the relationship with the relationship name `backup1_for_volume0001` at the mount point `/mnt`, execute the following command.

```
# sfcproxyop umount backup1_for_volume0001 /mnt
```

SEE ALSO

[sfcproxybreak\(1M\)](#), [sfcproxyinfo\(1M\)](#), [sfcproxyjoin\(1M\)](#), [sfcproxyrestore\(1M\)](#)

NOTES

- In the EC operation, if [sfcproxyop part](#) is executed during the execution of the real copy (equivalent copy) after [sfcproxyjoin equivalent](#) or [sfcproxyop rejoin](#) is executed, the execution waits until the equivalent copy is completed. Even in this case, the file system update operation at the backup source is locked only while the separation command is issued.
- Optional product is required for this function.

F.2.20 sfcproxyrestore(1M) Restores a file system at high speed

SYNOPSIS

sfcproxyrestore [-s] *rel_name*

DESCRIPTION

sfcproxyrestore executes high-speed restoration from the backup destination to the backup source specified by the relationship *rel_name*. Before this command is executed, the relationship between the backup source and destination must have been set up with [sfcproxyjoin\(1M\)](#).

When this command is executed, the following conditions must have been satisfied.

- The file system at the backup source must have stopped its service.
- The file system at the backup destination must have been unmounted.
- Copy processing must not have already been executed at the backup source or destination.
- The backup destination must have been separated from the backup source (must not be in join status).
- The backup source and destination volumes of GDS must not be data invalid (INVALID) status.

After this command is executed, the backup destination is put in the join status. In the OPC operation, separate the backup destination with [sfcproxyop part](#) after copy processing is completed so that backup can be executed again.

OPTION

The following option can be specified:

- s
Synchronization specification. This option waits for the real copy processing to be completed.

OPERAND

The following operand can be specified:

rel_name
Specify the relationship of a restoration target by using relationship name *rel_name*.

EXAMPLES

(Example 1)

To restore from the backup destination to the backup source specified with the relationship with the relationship name `backup1_for_volume0001` in the OPC operation, execute the following command.

```
# sfcproxyrestore -s backup1_for_volume0001
# sfcproxyop part backup1_for_volume0001
```

(Example 2)

To restore from the backup destination to the backup source specified with the relationship with the relationship name `backup1_for_volume0001` and put the backup source and destination in the equivalent status in the OPC operation, execute the following command.

```
# sfcproxyrestore backup1_for_volume0001
```

SEE ALSO

[sfcproxybreak\(1M\)](#), [sfcproxyinfo\(1M\)](#), [sfcproxyjoin\(1M\)](#), [sfcproxyop\(1M\)](#)

NOTE

- Optional product is required for this function.

F.2.21 sfcquota(1M) Displays the quota limit value and uses amount for each user

SYNOPSIS

sfcquota [-v] [*username*]

DESCRIPTION

sfcquota displays a warning for a GFS Shared File System whose quota function use amount is equal to or greater than the limit. **sfcquota** without options displays only warnings if the usage is above quota limits for mounted GFS Shared File Systems.

If an option is specified, **sfcquota** displays the current disk, the file use amount, limit, and time limit. The display details are as follows:

- Filesystem (file system amount point)
- usage (current disk use amount)
- files (current file use amount)
- quota (soft limit of the disk and file)
- limit (hard limit of the disk and file)
- timeleft (soft time limit of the disk and file)

Only a super user can specify operand *username* to display other user information.

OPTIONS

The following option can be specified:

-v

Displays the disk use amounts and limits of the users set in all mounted GFS Shared File Systems containing quotas files in the command execution node.

OPERAND

The following operand can be specified:

username

username can specify a login name or user ID. If *username* is omitted, information on the user executing this command is displayed.

EXAMPLES

```
# sfcquota user1
Block limit reached on /sfcs1

# sfcquota -v user1
Disk quotas for user1 (uid 24001):
Filesystem  usage quota limit timeleft files quota limit timeleft
/sfcs1      5  20  30          1  20  100

% sfcquota -v
Disk quotas for user1 (uid 24001):
Filesystem  usage quota limit timeleft files quota limit timeleft
/sfcs1      5  20  30          1  20  100
```

FILES

quotas

quota information file in the root directory of the file system

/etc/mnttab

Table of file system being mounted

SEE ALSO

[sfcquota\(1M\)](#), [sfcquotadm\(1M\)](#), [sfcquotaon\(1M\)](#), [sfcprepquota\(1M\)](#).

[mnttab\(4\)](#), [passwd\(4\)](#), [vfstab\(4\)](#) in the "*Solaris X Reference Manual Collection*."

NOTE

- **sfcquota** cannot display information on the GFS Shared File Systems NFS-mounted.

F.2.22 sfcquotadm(1M) Manages quotas files

SYNOPSIS

sfcquotadm -i *filesystem*
sfcquotadm -e *filesystem*
sfcquotadm -c *filesystem*
sfcquotadm -d *filesystem*

DESCRIPTION

sfcquotadm manages quotas files of the specified GFS Shared File System.

This command enables the following types of processing: copying or a comparison of user limitations between external and internal quotas files, and deletion of internal quotas files.

This command is used when an external quotas file is deleted, or overwritten by restoring. This command need not be used in general.

A file system to be specified by operand *filesystem* must be in use for the file system is in mounted status in the command execution node. In addition, the related root directory must have a quotas file (external quotas) owned by the root. *filesystem* can specify a mount point or block special file.

Only a super user can specify this command.

OPTIONS

The following options can be specified:

- i**
Reflects user limitations for the external quotas file in the internal quotas file.
- e**
Reflects user limitations for the internal quotas file in the external quotas file.
- c**
Compares user limitations for external and internal quotas files. For a super user (user ID: 0), this option compares time limitations for the number of blocks and files when the soft limit is exceeded. If user limitations between files do not match, this option displays user IDs and contents. A display example is shown below. < indicates the contents of the external quotas file and > indicates the contents of the internal quotas file.
--- uid = root(0) ---
< blocks time limit = 0 (default), files time limit = 0 (default)
> blocks time limit = 1.00 week, files time limit = 12.00 days
--- uid = 1000 ---
< block limits (soft = 0, hard = 0) file limits (soft = 0, hard = 0)
> block limits (soft = 0, hard = 5120) file limits (soft = 0, hard = 100)
--- uid = 2100 ---
< block limits (soft = 0, hard = 5120) file limits (soft = 0, hard = 100)
> block limits (soft = 3072, hard = 5120) file limits (soft = 80, hard = 100)
- d**
Deletes the internal quotas file. File deletion is possible only when the quota function is invalid.

OPERAND

The following operand can be specified:

filesystem

Specifies a mount-directory of the target file system or a block-special device of the representative partition.

EXAMPLES

```
# sfcquotadm -i /mnt
# sfcquotadm -e /mnt
# sfcquotadm -c /mnt
--- uid = root(0) ---
< blocks time limit = 0 (default), files time limit = 0 (default)
> blocks time limit = 1.00 week, files time limit = 12.00 days
--- uid = 1000 ---
< block limits (soft = 0, hard = 0) file limits (soft = 0, hard = 0)
> block limits (soft = 0, hard = 5120) file limits (soft = 0, hard = 100)
--- uid = 2100 ---
< block limits (soft = 0, hard = 5120) file limits (soft = 0, hard = 100)
> block limits (soft = 3072, hard = 5120) file limits (soft = 80, hard = 100)

# sfcquotadm -d /mnt
```

FILES**quotas**

quota information file in the root directory of the file system

/etc/mnttab

Table of file system being mounted

SEE ALSO

[sfcquota\(1M\)](#), [sfcquota\(1M\)](#), [sfcquotaon\(1M\)](#), [sfcprepquota\(1M\)](#),
[mnttab\(4\)](#), [vfstab\(4\)](#) in the "*Solaris X Reference Manual Collection*."

F.2.23 sfcquotaon(1M), sfcquotaoff(1M) Validates and invalidates the quota function

SYNOPSIS

```
sfcquotaon [ -v ] filesystem ...  
sfcquotaoff [ -v ] filesystem ...
```

DESCRIPTION

sfcquotaon validates the quota function of the GFS Shared File System specified by operand *filesystem*.

sfcquotaoff invalidates the quota function of the GFS Shared File System specified by the operand *filesystem*.

filesystem must be a mount point of the file system or a block special device of the representative partition of the GFS Shared File System.

To validate the quota function, the root directory of the file system must contain the file named *quotas* owned by the root. For details on this file, see [sfcquota\(1M\)](#).

The results obtained by changing the quota function when the specified file system is not used because of no the file system is unmounted are validated when the file system mounted in next time.

If the specified file system is mounted, the present quota function is changed. To indicate whether the quota function is valid or invalid, the *mntopts* field of the entry corresponding to */etc/mnttab* of each node is updated. As with UFS, **mount**(1M) can be used to check the quota function status.

OPTIONS

The following options can be specified:

sfcquotaon

-v

Displays a message of each file system whose quota function has been validated.

sfcquotaoff

-v

Displays a message of each file system whose quota function has been invalidated.

OPERAND

The following operand can be specified:

filesystem

Specifies a mount-directory of the target file system or a block-special device of the representative partition.

EXAMPLES

```
# sfcquotaon /mnt  
# sfcquotaoff /mnt
```

FILES

quotas

quota information file in the root directory of the file system

/etc/mnttab

Table of file system being mounted
/etc/vfstab
Environment file of the file system

SEE ALSO

[sfcquota\(1M\)](#), [sfcquota\(1M\)](#), [sfcquotadm\(1M\)](#), [sfcprepquota\(1M\)](#),
mnttab(4), **vfstab(4)** in the "*Solaris X Reference Manual Collection*."

NOTES

- The GFS Shared File System ignores **rq** or **quota** specified in the mntopts field of **/etc/vfstab**.
- Assume that a file system is mounted in another node. In this case, if the file system is unmounted in the node that executes **sfcquotaon** or **sfcquotaoff**, the quota function cannot be neither validated nor invalidated.

F.2.24 `sfcrepquota(1M)` Displays the quota function limits and uses amounts of all users

SYNOPSIS

```
sfcrepquota [ -v ] filesystem ...
sfcrepquota [ -v ] -a
```

DESCRIPTION

sfcrepquota displays the quota function limit and disk use amount of the specified GFS Shared File System. This command outputs the current number of files and current number of blocks (in kilobytes) of each user together with the quota limit specified by [sfcedquota\(1M\)](#). The output details are as follows:

- User (user ID of each user)
- used (use amount of the current disk or file)
- soft (soft limit of the disk or file)
- hard (hard limit of the disk or file)
- timeleft (soft time limit of the disk or file)

The signs **+** and **-** are used to indicate maximum use of disk and file, respectively.

The file system specified by operand *filesystem* must have a `quotas` file owned by the root in the root directory.

Only a super user can reference other user information.

OPTIONS

The following options can be specified:

- a**
When the current quota function is valid, this option displays all GFS Shared File Systems being mounted in the command execution node.
- v**
Displays all quota information items even if a user is not using file system resources.

OPERAND

The following operand can be specified:

filesystem

Specifies a mount-directory of the target file system or a block-special file of the representative partition.

EXAMPLES

```
# sfcrepquota /mnt
/dev/sfdsk/gfs/rdsk/volume1 (/sfcs1):
      Block limits      File limits
User   used soft hard  timeleft  used soft hard  timeleft
user1  -- 2258 3072 5120      71  80  100
user2  -- 1345 3072 5120      38  80  100
user3  ++ 3424 3072 5120  7.0 days  81  80  100  7.0 days
```



```
# sfcprepquota -v /mnt
/dev/sfdsk/gfs/rdisk/volume1 (/sfcs1):
      Block limits      File limits
User   used soft hard  timeleft  used soft hard  timeleft
user1  -- 2258 3072 5120          71  80 100
user2  -- 1345 3072 5120          38  80 100
user3  ++ 3424 3072 5120  7.0 days  81  80 100  7.0 days
user4  --  0 3072 5120          0  80 100

# sfcprepquota -a
```

FILES

quotas

quota information file in the root directory of the file system

/etc/mnttab

Table of file system being mounted

/etc/vfstab

Environment file for the file system.

SEE ALSO

[sfcquota\(1M\)](#), [sfcquota\(1M\)](#), [sfcquotadm\(1M\)](#), [sfcquotaon\(1M\)](#).

NOTE

- **sfcprepquota** displays a warning for the GFS Shared File System whose quota function is invalid or setting now. Use [sfcinfo\(1M\)](#) to confirm the status of quota function.

F.2.25 `sfcrestore(1M)` Restores from the dump created with `sfcdump`

SYNOPSIS

`sfcrestore` *i|r|t|x*[*abdefhlmosvyLT*] [*archive_file*] [*factor*] [*e_opt*] [*dump_file*] [*n*] [*filename...*]

DESCRIPTION

The `sfcrestore` utility restores files from backup media created with the `sfcdump` command. `sfcrestore`'s actions are controlled by the key argument. The key is exactly one function letter (*i*, *r*, *t*, or *x*) and zero or more function modifiers (letters). The key string contains no SPACE characters. Function modifier arguments are listed on the command line in the same order as their corresponding function modifiers appear in the key string.

The `sfcrestore` also operates on a system other than the GFS Shared File System. For information about the operations to be performed when the dump created with `sfcdump` is restored on a file system other than GFS Shared File System or the relationships between `sfcrestore` and `ufsrestore`, see NOTES below.

OPTIONS

[Function letters]

One (and only one) of the following function letters is required:

i

Interactive. After reading in the directory information from the media, `sfcrestore` invokes an interactive interface that allows you to browse through the dump file's directory hierarchy and select individual files to be extracted. See Interactive commands, below, for a description of available commands.

r

Recursive. Restore the entire contents of the media into the current directory (which should be the top-level of the file system). To completely restore a file system, use this function letter to restore the level 0 dump, and again for each incremental dump. Although, this function letter is intended for a complete restore onto a clear file system, if the file system contains files not on the media, they are preserved.

t

Table of contents. List each filename that appears on the media. If no *filename* operand is given, the root directory is listed. This results in a list of all files on the media, unless the **h** function modifier is in effect. The table of contents is taken from the media or from the specified archive file, when the **a** function modifier is used.

x

Extract the named files from the media. If a named file matches a directory whose contents were written onto the media, and the **h** function modifier is not in effect, the directory is recursively extracted. The owner, modification time, and mode are restored (if possible). Existing files are overwritten and a warning is given. If no *filename* operand is given, the root directory is extracted. This results in the entire tape being extracted unless the **h** function modifier is in effect.

[Function modifiers]

a *archive_file*

Read the table of contents from *archive_file* instead of the media. This function modifier can be used in combination with the **t**, **i** or **x** function letters, making it possible to check whether files are on the media without having to mount the media. When used with the **x** and interactive (**i**) function letters, it prompts for the volume containing the file(s) before extracting them.

b *factor*

Blocking factor. Specify the blocking factor for tape reads. For variable length SCSI tape devices, unless the data was written with the default blocking factor, a blocking factor at least as great as that used to write the tape must be used; otherwise, an error will be generated. Note that a tape block is 512 bytes. Refer to the man page for your specific tape driver for the maximum blocking factor.

d

Turn on debugging output.

e e_opt

Specifies how to handle files that have extent attribute information. It's impossible to set the information onto files if the destination file system does not support file extension attribute information or it cannot satisfy the conditions required to set the attribute (e.g. lacks free extents appropriate to satisfy the extent attribute information requirements). This function modifier specifies the action in those case. Valid values for *e_opt* are:

force

Fails to restore files if extent attribute information could not be set.

ignore

Ignores to extent attribute information entirely.

warn

Issues a warning message if extent attribute information could not be set to a file. The extent attribute information of the file is erased. The default value for *e_opt* is **warn**.

f dump_file

Use *dump_file* instead of /dev/rmt/0 as the file to restore from. Typically *dump_file* specifies a tape or diskette drive. If *dump_file* is specified as '-', **sfcrestore** reads from the standard input. This allows [sfcdump\(1M\)](#) and **sfcrestore** to be used in a pipeline to copy a file system:

```
# sfcdump 0f - /dev/sfdsk/gfs/rdisk/volume1 | ( cd /home; sfcrestore xf - )
```

If the name of the *dump_file* is of the form *machine:device*, the restore is done from the specified machine over the network using **rmt(1M)**. Since **sfcrestore** is normally run by root, the name of the local machine must appear in the *.rhosts* file of the remote machine. If *dump_file* is specified as *user@machine:device*, **sfcrestore** will attempt to execute as the specified user on the remote machine. The specified user must have a *.rhosts* file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.

h

Extract or list the actual directory, rather than the files that it references. This prevents hierarchical restoration of complete sub trees from the tape.

l (lower-case l)

Autoload. When the end-of-tape is reached before the restore is complete, take the drive off-line and wait up to **two** minutes (the default, see the **T** function modifier) for the tape drive to be ready again. This gives autoloading (stackloader) tape drives a chance to load a new tape. If the drive is ready within **two** minutes, continue. If it is not, prompt for another tape and wait.

L string

The *string* as label that should appear in the header of the dump file. If the labels do not match, **sfcrestore** issues a diagnostic and exits. The tape label is specific to the [sfcdump](#) tape format, and bears no resemblance to IBM or ANSI-standard tape labels.

m

Extract by i-node numbers rather than by filename to avoid regenerating complete pathnames. Regardless of where the files are located in the dump hierarchy, they are restored into the current directory and renamed with their i-node number. This is useful if only a few files are being extracted.

o (lower-case o)

Offline. Take the drive off-line when the restore is complete or the end-of-media is reached and rewind the tape, or eject the diskette. In the case of some autoloading 8mm drives, the tape is removed from the drive automatically.

s n

Skip to the *n*'th file when there are multiple dump files on the same tape. For example, the command:

```
# sfcrestore xfs /dev/rmt/0hn 5
```

would position you to the fifth file on the tape when reading volume 1 of the dump. If a dump extends over more than one volume, all volumes except the first are assumed to start at position 0, no matter what "**s n**" value is specified. If "**s n**" is specified, the backup media must be at BOT (beginning of tape). Otherwise, the initial positioning to read the table of contents will fail, as it is performed by skipping the tape forward *n*-1 files rather than by using absolute positioning. This is because on some

devices absolute positioning is very time consuming.

T *timeout* [hms]

Sets the amount of time to wait for an autoload command to complete. This function modifier is ignored unless the **I** function modifier has also been specified. The default timeout period is **two** minutes. The time units may be specified as a trailing **h** (hours), **m** (minutes), or **s** (seconds). The default unit is minutes.

v

Verbose. **sfcrestore** displays the name of each file it restores, preceded by its file type.

y

Do not ask whether to abort the restore in the event of tape errors. **sfcrestore** tries to skip over the bad tape block(s) and continue as best it can.

[Interactive Commands]

sfcrestore enters interactive mode when invoked with the **i** function letter. Interactive commands are reminiscent of the shell. For those commands that accept an argument, the default is the current directory. The interactive options are:

add [*filename*]

Add the named file or directory to the list of files to extract. If a directory is specified, add that directory and its files (recursively) to the extraction list (unless the **h** function modifier is in effect).

cd *directory*

Change to *directory* (within the dump file).

delete [*filename*]

Delete the current directory, or the named file or directory from the list of files to extract. If a directory is specified, delete that directory and all its descendents from the extraction list (unless the **h** function modifier is in effect). The most expedient way to extract a majority of files from a directory is to add that directory to the extraction list, and then delete specific files to omit.

extract

Extract all files on the extraction list from the dump media. **sfcrestore** asks which volume the user wishes to mount. The fastest way to extract a small number of files is to start with the last volume and work toward the first.

help

Display a summary of the available commands.

ls [*directory*]

List files in *directory* or the current directory, represented by a **`.`** (period). Directories are appended with a **`.`** (slash). Entries marked for extraction are prefixed with a **``*** (asterisk). If the verbose option is in effect, i-node numbers are also listed.

marked [*directory*]

Like **ls**, except only files marked for extraction are listed.

paginate

Toggle the pagination of the output from the **ls** and **marked** commands. The **paginate** used is that defined by the **PAGER** environment variable, or **more(1)** if that envvar is not defined. The **PAGER** envvar may include white-space-separated arguments for the pagination program.

pwd

Print the full pathname of the current working directory.

quit

sfcrestore exits immediately, even if the extraction list is not empty.

setmodes

Prompts: set owner/mode for **`.`** (period). Type **y** for yes to set the mode (permissions, owner, times) of the current directory **`.`** (period) into which files are being restored equal to the mode of the root directory of the file system from which they were dumped. Normally, this is what you want when restoring a whole file system, or restoring individual files into the same locations from which they were dumped. Type **n** for no, to leave the mode of the current directory unchanged. Normally, this is what you want when restoring part of a dump to a directory other than the one from which the files were dumped.

setpager *command*

Sets the command to use for paginating output instead of the default or that inherited from the environment. The *command* string may include arguments in addition to the command itself.

verbose

Toggle the status of the **v** function modifier. While **v** is in effect, the **ls** command lists the i-node numbers of all entries, and **sfcrestore** displays information about each file as it is extracted.

what

Display the dump header on the media.

OPERAND

The following operand can be specified:

filename

Specifies the pathname of files (or directories) to be restored to disk. Unless the **h** function modifier is also used, a directory name refers to the files it contains, and (recursively) its subdirectories and the files they contain. *filename* is associated with either the **x** or **t** function letters, and must come last.

EXIT STATUS

The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred. Verbose messages are displayed.

FILES

/dev/rmt/0

The default tape drive.

/tmp/rst^{*}dir

File containing directories on the tape. The directory where this file is created can be specified by setting the environment variable **TMPDIR**.

/tmp/rst^{*}mode

Owner, mode, and timestamps for directories. The directory where this file is created can be specified by setting the environment variable **TMPDIR**.

./restoresymtable

Information passed between incremental restores.

ENVIRONMENT

PAGER

The command to use as a filter for paginating output. This can also be used to specify the options to be used. Default is **more(1)**.

TMPDIR

sfcrestore normally creates temporary files in the directory **/tmp**. You may specify another directory by setting the environment variable **TMPDIR** to your chosen directory. (If **TMPDIR** isn't a valid directory, then **sfcrestore** will use **/tmp**).

SEE ALSO

[sfcdump\(1M\)](#).

mkfs(1M), **mount(1M)**, **rmt(1M)** in the "*Solaris X Reference Manual Collection*."

DIAGNOSTICS

sfcrestore complains about bad option characters.

Read errors result in complaints. If **y** has been specified, or the user responds **y**, **sfcrestore** will attempt to continue.

If the dump extends over more than one tape, **sfcrestore** asks the user to change tapes. If the **x** or **i**

function letter has been specified, **sfcrestore** also asks which volume the user wishes to mount. If the **s** function modifier has been specified, and volume 1 is mounted, it is automatically positioned to the indicated file.

There are numerous consistency checks that can be listed by **sfcrestore**. Most checks are self-explanatory or can "never happen." Common errors are given below.

filename : not found on tape

The specified file name was listed in the tape directory, but was not found on the tape. This is caused by tape read errors while looking for the file, or from using a dump tape created on an active file system.

expected next file *inumber*, got *inumber*

A file that was not listed in the directory showed up. This can occur when using a dump tape created on an active file system.

Incremental tape too low

When doing an incremental restore, a tape that was written before the previous incremental tape, or that has too low an incremental level has been loaded.

Incremental tape too high

When doing incremental restore, a tape that does not begin its coverage where the previous incremental tape left off, or one that has too high an incremental level has been loaded.

media read error: invalid argument

Blocking factor specified for read is smaller than the blocking factor used to write data.

Tape read error while restoring *filename*

Tape read error while skipping over inode *inumber*

Tape read error while trying to resynchronize

A tape read error has occurred

If a file name is specified, then its contents are probably partially wrong. If an i-node is being skipped or the tape is trying to resynchronize, then no extracted files have been corrupted, though files may not be found on the tape.

Incorrect tape label. Expected *'foo'* got *'bar'*

The **L** option was specified, and its value did not match what was recorded in the header of the dump file.

resync restore, skipped *num*

After a tape read error, **sfcrestore** may have to resynchronize itself. This message lists the number of blocks that were skipped over.

NOTES

- **sfcrestore** can get confused when doing incremental restores from dump tapes that were made on active file systems.
- A "level 0" dump must be done after a full restore. Because **sfcrestore** runs in user mode, it has no control over i-node allocation. This means that **sfcrestore** repositions the files, although it does not change their contents. Thus, a full dump must be done to get a new set of directories reflecting the new file positions, so that later incremental dumps will be correct.
- Internal quotas file is not restored by **sfcrestore**. If the file system on which quota function is used, it's necessary to reflect the contents of external quotas file to internal quotas file by executing [sfcquotadm\(1M\)](#) with **-i** option. For details of the [sfcquotadm\(1M\)](#), see the reference manual.
- **sfcrestore** fails if the file larger than 2 gigabytes is restored on the file system which does not support files larger than 2 gigabytes.
- **sfcrestore** and **ufsrestore** are functionally compatible, though the backup formats are different. As a result, the **sfcrestore** cannot use the backup data created using the **ufsdump** for restoration. In the same way, **ufsrestore** cannot use the backup data created using [sfcdump\(1M\)](#) for restoration.
- If files are restored on the file system that does not support extent attribute information such as UFS, extent attribute information is erased. If files are restored on the file system that does not support ACLs, ACLs are erased.

F.2.26 sfcrcinfo(1M) Displays file system information

SYNOPSIS

```
sfcrcinfo -m [-a | -i fsid | special]
```

DESCRIPTION

The **sfcrcinfo** command outputs file system information for GFS Shared File Systems.

OPTIONS

The following options can be specified:

-a

Outputs information for all GFS Shared File Systems that have been created.

-m

Outputs the MDS and AC information for GFS Shared File Systems. If **-m** specified with **-a**, outputs information for all GFS Shared File Systems that have been created. If only specified **-m**, outputs information for GFS Shared File System which status is mounted.

-i *fsid*

Displays the information for the file system specified by *fsid*.

OPERAND

The following operand can be specified:

special

Displays information for the file system specified by *special*.

EXAMPLES

```
# sfcrcinfo -m /dev/sfdsk/gfs/rdsk/volume1
/dev/sfdsk/gfs/dsk/volume1:
FSID MDS/AC STATE S-STATE RID-1 RID-2 RID-N hostname
 1 MDS(P) run - 0 0 0 prime01
 1 AC run - 0 0 0 prime01
 1 MDS(S) wait - 0 0 0 prime02
 1 AC run - 0 0 0 prime02
 1 MDS(I) imag - 0 0 0 prime03
 1 AC run - 0 0 0 prime03
```

- MDS/AC
 - MDS(P) : The primary MDS on a definition
 - MDS(S) : The secondary MDS on a definition
 - MDS(I) : The other nodes to which MDS operates
 - AC : The node that mount the file system
- STATE/SUB-STATE of MDS
 - run : The primary MDS is running
 - wait : The secondary MDS is waiting for failure recovery of the primary MDS
 - imag : Waiting for failure recovery of the primary MDS or the secondary MDS
 - stop : MDS stopped
 - none : no relation with MDS
 - If SUB-STATE is other than "-", MDS is in state-transition.
- STATE/SUB-STATE of AC
 - run : file system is in mounted
 - stop : file system is in unmounted
 - If SUB-STATE is other than "-", AC is in state- transition.

F.2.27 sfcsetup(1M) Initializes the management partition; adds, deletes, displays path of the management partition

SYNOPSIS

```
sfcsetup -c [-f] raw_device_file  
sfcsetup -a raw_devece_file  
sfcsetup -d  
sfcsetup -p  
sfcsetup
```

DESCRIPTION

The **sfcsetup** command is used to initialize the management partition, and to add, delete, and display node information in the management partition, and display the path of management partition.

To initialize the management partition, specify the **-c** option and the character device. When the specified device seems to have been user for the management partition, perform initialization with the **-f** option specified. The size of a management partition needs to be 40 Mbyte or more.

To add node information, specify the **-a** option. When the specified character device has already been initialized for the management partition, node information is added to the management partition for which the command was executed.

To delete node information, specify the **-d** option. Note, however, that **sfcrmd** demon must be stopped in all node of the cluster foundation.

When the command is executed with **-p** options specified, display the path name of a management partition.

When the command is executed with no options specified, a listing of the node information recorded in the management partition is displayed.

When addition of node information and deletion are performed by the **sfcsetup** command, add/delete entries of management partition to/from the **sfcsrm.conf** file.

OPTIONS

The following options can be specified:

```
-c raw_device_file  
    Initializes the partition specified in raw_device_file for use as a management partition.  
-a raw_device_file  
    Adds node information.  
-d  
    Deletes node information.  
-f  
    Initialize a character special device that has been used as a management partition.  
-p  
    Display the path name of a management partition.
```

OPERAND

The following operand can be specified:

```
raw_device_file  
    Specifies the character special device to be used for the management partition.
```


EXAMPLES

Initializing a management partition.

```
# sfcsetup -c /dev/sfdsk/gfs/rdsk/control
```

When you display node information registered in the management partition.

```
# sfcsetup
HOSTID      CIPNAME      MP_PATH
80f1ab32    host1RMS     yes
80f1ab33    host2RMS     yes
```

Initializing a parturition that has previously been used as a management partition.

```
# sfcsetup -c -f /dev/sfdsk/gfs/rdsk/control
```

When sfcfrmd is not running in all node of the cluster, and adding information for a node.

```
# sfcsetup -a /dev/sfdsk/gfs/rdsk/control
```

When sfcfrmd is not running in the cluster, and deleting information for a node.

```
# sfcsetup -d
```

FILE

/var/opt/FJSVsfcs/sfcfrmd.conf

Environment file of sfcfrmd daemon

SEE ALSO

[sfcadm\(1M\)](#), [sfcfrmdstop\(1M\)](#).

NOTE

- Initialization of a management partition, addition and deletion of node information cannot be performed when the sfcfrmd demon has started.
- Node information used by a file system cannot be deleted. Please delete the file system configuration information related the node in a management partition by [sfcadm\(1M\)](#).

F.2.28 sfcstat(1M) Reports statistics on the GFS Shared File System

SYNOPSIS

sfcstat [**-abcdefghijklmnoqrstA**] [*interval* [*count*]] [*mount_point* ...]

DESCRIPTION

The **sfcstat** command displays statistics about GFS Shared File Systems.

sfcstat first reports statistical information recorded after the file system is mounted. The next time, it reports the information recorded since the last report.

When the parameter *interval* is specified, **sfcstat** reports statistical information for the interval that was specified in this parameter (in seconds). When no *interval* specified, it reports on statistical information only once.

When the parameter *count* is specified, **sfcstat** reports statistical information for the interval it takes to count to count. When no *count* specified, it keeps displaying statistical information.

When the parameter *mount_point* is specified, **sfcstat** reports statistical information on the specified file system. When no *mount_point* specified, **sfcstat** reports statistical information on all mounted GFS Shared File System.

OPTIONS

The following options can be specified:

If the options are omitted, the command reports all information.

-A

Reports all information.

-a

Reports AC information about read/write operations. Reports the number of RPC messages for the following purposes:

- To change file atime or mtime
- To change file attributes
- To access file attributes
- To access file size
- To acquire tokens when changing file size
- To change file size
- To purge tokens when changing file size
- To access information about extent allocations
- To change information about extent allocations

-b

Reports AC information about meta-data management buffers. Reports the number of buffer allocations, the number of dynamic buffer allocations, and the number of sleep occurred during buffer requests.

-c

Reports MDS information about meta-data management buffers. Reports the number of buffer allocations, the number of dynamic buffer allocations, and the number of sleep occurred during buffer requests.

-d

Reports AC information about direct I/O. Reports the following information:

- Number of direct I/O read operations
- Number of direct I/O write operations
- Number of blocks read
- Number of blocks written
- Number of read requests changed to normal I/O
- Number of write requests changed to normal I/O
- Number of page flush after reading
- Number of page flush after writing

- e** Reports MDS information about purging tokens. Reports the following information:
 - Number of purging tokens
 - Number of AC that purged tokens
 - Number of purging tokens (by token type)
 - Number of AC that purged tokens (by token type)

- f** Reports AC information about VFS operations. Reports the number of RPC messages for the following purposes:
 - To access file handle
 - To access file attributes
 - To purge tokens

- g** Reports the number of times MDS was received requests for tokens (by token type).

- h** Reports MDS information about the meta-daemon. Reports the number of times the meta-daemon was activated, the number of buffers in the meta-daemon I/O queues, the number of times the meta-daemon was activated because the disk was full, and the number of times the meta-daemon was activated because the meta-cache was full.

- i** Reports AC information about actual I/O. Reports the number of actual I/O reading, the number of actual I/O writing, the number of I/O blocks actually read, and the number of I/O blocks actually written.

- j** Reports MDS information about the log daemon. Reports the following information:
 - Number of times log daemon was activated
 - Number of sync requests
 - Number of fsync requests
 - Number of times log buffer full
 - Number of requests issued from meta-write daemon
 - Number of times sleep occurs because transaction limit exceeded
 - Number of times sleep occurs because log disk full
 - Number of times sleep occurs because of a preceding large transaction
 - Number of times sleep occurs because a preceding transaction enlarged
 - Number of requests for I/O operation issued from log daemon
 - Time required for writing
 - Number of blocks written
 - Number of valid blocks
 - Number of requests to log transactions
 - Number of parallel operation transactions to be logged
 - Average number of parallel operation transactions
 - Number of times huge buffer was used
 - Number of sub-transactions generated

- k** Reports MDS information about operations for free extents. Reports the number of search requests, the number of times blocks acquired, the number of acquired blocks, the number of times acquisition failed, the number of times blocks released, the number of released blocks, the number of times released blocks successfully linked, the number of division of free extent management, and the number of joining free extent management.

- l** Reports AC information about file and record locks. Reports the number of RPC messages for the following purposes:
 - To open file
 - To close file
 - To map file to address space
 - To read file
 - To write file

F.2 Management Commands Specific to the GFS Shared File System

- To set lock, clear lock, or get lock information
- To set mandatory lock, or clear mandatory lock
- m**

Reports MDS information about meta cache (by meta type). Reports the number of accesses, the number of cache hits, the number of updates, the number of read operations, the number of write operations, writing time, and average time.
- n**

Reports AC information about RNODE operations. Reports the number of RPC messages for the following purposes:

 - To be requested to purge tokens
 - To be requested to purge tokens of freed RNODE
 - To purge tokens to free RNODE
- o**

Reports AC information about VNODE operations. Reports the following information:

 - Number of RPC messages to open file
 - Number of RPC messages to close file
 - Number of access to file attributes
 - Number of RPC messages to access file attributes
 - Number of re-sent RPC messages to access file attributes
 - Number of RPC messages to access the number of blocks of regular file
 - Number of access to symbolic link
 - Number of RPC messages to access symbolic link
 - Number of RPC messages to move modified attributes to storage devices
 - Number of access to directory
 - Number of RPC messages to access directory
 - Number of re-sent RPC messages to access directory
- p**

Reports MDS information about file operation. Reports the following information:

 - Number of requests to operate files (by operation type)
 - Size of RPC messages to operate files (by operation type)
 - Number of successful replies (by operation type)
 - Size of successful replies (by operation type)
 - Number of failed replies (by operation type)
 - Time required for operating files (by operation type)
- q**

Reports MDS information about operations for allocated extents. Reports the number of search requests, the number of times blocks acquired, the number of acquired blocks, the number of times acquisition failed, the number of times blocks released, the number of released blocks, the number of times released blocks successfully linked, the number of division of free extent management, and the number of joining free extent management.
- r**

Reports AC information about file operation. Reports the following information:

 - Number of sending requests to operate files
 - Number of receiving replies
 - Number of timeouts while receiving replies
 - Number of requests waiting to be sent (by operation type)
 - Time for requests to wait to be sent (by operation type)
 - Number of sending requests (by operation type)
 - Time to wait to receive replies (by operation type)
 - Number of successful replies (by operation type)
 - Number of replies failed because no file existed (by operation type)
 - Number of replies failed because file attributes changed (by operation type)
 - Number of replies failed because file existed (by operation type)
 - Number of replies failed because of other error (by operation type)
 - Number of failure to receive replies (by operation type)
- s**

F.2.28 sfcstat(1M) Reports statistics on the GFS Shared File System

Reports MDS information about operations for reserving extents. Reports the number of requests to reserve extents, the number of reserved extents, the number of reserved blocks, the number of requests to reserve extents contiguously, and the number of extents failed to reserve.

-t

Reports the number of times MDS is request to purging tokens. Reports the following information:

- Number of changing information about extent allocations
- Number of extents whose allocation information were changed
- Number of blocks whose allocation information were changed
- Number of changing information about extent allocations partially
- Number of extents whose allocation information were changed partially
- Number of blocks whose allocation information were changed partially
- Number of purging tokens (by token type)

OPERAND

The following operand can be specified:

mount_point

Specifies the path name of the mount point where the file system has been mounted.

EXAMPLE

```
# sfcstat 5 /mnt
bufalloc dbufalloc dbufsleap
 0 0 0
Filesystem:/mnt Thu Mar 13 15:58:09 JST 2003
eviction normode vacated
 0 0 0
vget getattr vacated
 0 0 0
open close gattr gattrc gattrn gattrv
 0 0 0 0 0 0
rlink rlinkc fsync rdir rdirc rdirn
 0 0 0 0 0 0
open close map read write flock vacated
 0 0 0 0 0 0 0
utime uattr cattr csize truncg truncs truncv extg extv
 0 0 0 0 0 0 0 0 0
dirr dirw dirrblk dirwblk dirr-n dirw-n dirrflsh dirwflsh
 0 0 0 0 0 0 0 0
send recv timeout
 11 15 9
slist swait addsl
rwait rok moent rbverf rethis rerror rnores
NULL 0 0 0
GETATTR 0 0 0 0 0 0 0
SETATTR 0 0 0 0 0 0 0
LOOKUP 0 0 0 0 0 0 0
READLINK 0 0 0 0 0 0 0
CREATE 0 0 0 0 0 0 0
REMOVE 0 0 0
```

F.2 Management Commands Specific to the GFS Shared File System

RENAME	0	0	0	0	0	0	0
	0	0	0				
LINK	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
SYMLINK	0	0	0				
	0	0	0	0	0	0	0
MKDIR	0	0	0				
	0	0	0	0	0	0	0
RMDIR	0	0	0				
	0	0	0	0	0	0	0
READDIR	0	0	0				
	0	0	0	0	0	0	0
STATFS	0	0	0				
	0	0	0	0	0	0	0
GETLEASE	0	0	0				
	0	0	0	0	0	0	0
VACATED	0	0	0				
	0	0	0	0	0	0	0
EVICTION	0	0	0				
	0	0	0	0	0	0	0
FILELOCK	0	0	0				
	0	0	0	0	0	0	0
HEARTBEAT	0	0	0				
	0	0	0	0	0	0	0
KSTAT	0	0	11				
	20	11	0	0	0	0	0
MOUNT	0	0	0				
	0	0	0	0	0	0	0
ADDVOL	0	0	0				
	0	0	0	0	0	0	0
RESERVE	0	0	0				
	0	0	0	0	0	0	0
UNMOUNT	0	0	0				
	0	0	0	0	0	0	0
FUNMOUNT	0	0	0				
	0	0	0	0	0	0	0
MNTINFO	0	0	0				
	0	0	0	0	0	0	0
ADDVOLINFO	0	0	0				
	0	0	0	0	0	0	0
BARSYNC	0	0	0				
	0	0	0	0	0	0	0
RECOVER	0	0	0				
	0	0	0	0	0	0	0
RNOTIFY	0	0	0				
	0	0	0	0	0	0	0
FSCTL	0	0	0				
	0	0	0	0	0	0	0
FSHINT	0	0	0				
	0	0	0	0	0	0	0
QUOTACTL	0	0	0				
	0	0	0	0	0	0	0
QUOTASET	0	0	0				
	0	0	0	0	0	0	0
QUOTAGET	0	0	0				

F.2.28 sfcstat(1M) Reports statistics on the GFS Shared File System

QUERYBLK	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
UPDATEBLK	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
EVICTONBLK	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
FREEZE	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
SNAPSHOT	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
EVICTEDALL	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
COPYBLK	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
SYNC	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
VGET	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
ITRACECTL	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	
		readcnts	readblks	writcnts	writeblks			
data volume[00]:	0	0	0	0	0			
metavolume[0]								
metatype	access	cachehit	mod	R-I/O	W-I/O	W-MS	I/Oave	
DBMAP	0	0	0	0	0	0	0	
IBMAP	0	0	0	0	0	0	0	
NBMAP	0	0	0	0	0	0	0	
LBMAP	0	0	0	0	0	0	0	
SBMAP	0	0	0	0	0	0	0	
IEX	0	0	0	0	0	0	0	
DIR	0	0	0	0	0	0	0	
SYM	0	0	0	0	0	0	0	
NODE	0	0	0	0	0	0	0	
LEAF	0	0	0	0	0	0	0	
SLIST	0	0	0	0	0	0	0	
DINODE	1	1	1	0	1	10	10	
SBLOCK	0	0	0	0	0	0	0	
VOLINF	0	0	0	0	0	0	0	
RTERM	0	0	0	0	0	0	0	
ACL	0	0	0	0	0	0	0	
QUOTA	0	0	0	0	0	0	0	
RLEAF	0	0	0	0	0	0	0	
bufalloc	dbufalloc	dbufsleep						
1	0	0						
fsearch	falloc	fnalloc	fealloc	ffree	fnfree	ffcomb	ftsplits	ftcomb
0	0	0	0	0	0	0	0	0
asearch	aalloc	analloc	aealloc	afree	anfree	afcomb	atsplits	atcomb
0	0	0	0	0	0	0	0	0
mwakeup	mnbufl	mlogful						
5	1	0						
METAFULL	DBMAP	IBMAP	NBMAP	LBMAP	SBMAP	IEX		
0	0	0	0	0	0			
DIR	SYM	NODE	LEAF	SLIST	DINODE			
0	0	0	0	0	0			
SBLOCK	VOLINF	RTERM	ACL	QUOTA	RLEAF			

F.2 Management Commands Specific to the GFS Shared File System

	0	0	0	0	0	0			
l wakeup	l sync	l fsync	l bfull	l meta	l tfull	l dfull	logbig	logbigx	
	10	0	1	0	0	0	0	0	0
Inwrite	l ms	l nblks	l occu	l ntrans	l para	l ave	l bigbuf	l subtrn	
	1	0	2	2	1	0	0	0	0
		recv	recvb	okreply	okreplyb	ngreply		r_ms	
NULL		0	0	0	0	0	0	0	
GETATTR		1	96	1	172	0	0		
SETATTR		0	0	0	0	0	0		
LOOKUP		1	100	1	188	0	0		
READLINK		0	0	0	0	0	0		
CREATE		0	0	0	0	0	0		
REMOVE		0	0	0	0	0	0		
RENAME		0	0	0	0	0	0		
LINK		0	0	0	0	0	0		
SYMLINK		0	0	0	0	0	0		
MKDIR		0	0	0	0	0	0		
RMDIR		0	0	0	0	0	0		
READDIR		0	0	0	0	0	0		
STATFS		0	0	0	0	0	0		
GETLEASE		1	100	1	108	0	0		
VACATED		2	240	2	72	0	0		
EVICTION		0	0	0	0	0	0		
FILELOCK		0	0	0	0	0	0		
HEARTBEAT		0	0	0	0	0	0		
KSTAT		11	1044	11	8616	0	10		
MOUNT		0	0	0	0	0	0		
ADDVOL		0	0	0	0	0	0		
RESERVE		0	0	0	0	0	0		
UNMOUNT		0	0	0	0	0	0		
FUNMOUNT		0	0	0	0	0	0		
MNTINFO		0	0	0	0	0	0		
ADDVOLINFO		0	0	0	0	0	0		
BARSYNC		0	0	0	0	0	0		
RECOVER		0	0	0	0	0	0		
RNOTIFY		0	0	0	0	0	0		
FSCTL		0	0	0	0	0	0		
FSHINT		0	0	0	0	0	0		
QUOTACTL		0	0	0	0	0	0		
QUOTASET		0	0	0	0	0	0		
QUOTAGET		0	0	0	0	0	0		
QUERYBLK		0	0	0	0	0	0		
UPDATEBLK		0	0	0	0	0	0		
EVICTONBLK		0	0	0	0	0	0		
FREEZE		0	0	0	0	0	0		
SNAPSHOT		0	0	0	0	0	0		
EVICTEDALL		0	0	0	0	0	0		
COPYBLK		0	0	0	0	0	0		
SYNC		0	0	0	0	0	0		
VGET		0	0	0	0	0	0		
ITRACECTL		0	0	0	0	0	0		
	evict	nevict							
	0	0							
token	tevict	tnevict							
DREAD	0	0							
DWRITE	0	0							

SREAD	0	0				
SWRITE	0	0				
TREAD	0	0				
TWRITE	0	0				
NREAD	0	0				
AREAD	0	0				
DELETE	0	0				
token getlease						
DREAD	1					
DWRITE	0					
SREAD	1					
SWRITE	0					
ROPEN	1					
WOPEN	0					
RMMAP	0					
WMMAP	0					
TWRITE	1					
ext	next	blk	npext	nnext	nrbk	
0	0	0	0	0	0	0
token request						
SIZE	0					
ATIME	0					
MTIME	0					
CREAD	1					
CWRITE	0					
CMREAD	0					
CMWRITE	0					
CMLOCK	0					
CCALL	1					
CFONLY	0					
ABSATIME	1					
ASBMTIME	0					
SYNC	0					
NEEDNTF	0					
rqst	next	nblk	ncrqst	empty		
0	0	0	0	0		

SEE ALSO

sar(1) in the "Solaris X Reference Manual Collection."

F.2.29 sfctune(1M) Tunes the GFS Shared File System

SYNOPSIS

sfctune [*specific_options*] *special* | *filesystem*

DESCRIPTION

The **sfctune** command changes dynamic parameters that have an effect on the block allocation scheme of the specified GFS Shared File System.

Before using **sfctune**, set the file system to unmounted in all node.

Before **sfctune** can be used, the file system must be described in the `/etc/vfstab` file.

Specify the parameters to be changed with the options described below.

If the file system is configured with multiple partitions, specify the path name of the representative partition as a special file name.

OPTIONS

specific_options

The following options can be specified for sfctune:

-m minfree

Specifies the percentage of the area that ordinary users are not allowed to use.

-o opt

Specifies how the file system is to be optimized.

space: Space

time: Access time

-a maxcontig

Specifies the maximum number of contiguous blocks allocated before a forcible rotation wait occurs.

-M mminfree

Specifies the percentage of the V-data area that ordinary users are not allowed to use.

OPERANDS

The following operands can be specified:

special

Specifies the representative partition of the target file system.

filesystem

Specifies the mount-point of the target file system.

EXAMPLE

```
# sfctune -m 5 /dev/sfdsk/gfs/dsk/volume1
```

FILE

/etc/vfstab

Environment file of the file system

SEE ALSO

[mkfs_sfcfs\(1M\)](#).

vfstab(4) in the "Solaris X Reference Manual Collection."

F.2.30 sfcumntgl(1M) Unmounts the GFS Shared File System from all nodes

SYNOPSIS

```
sfcumntgl special | mount_point
sfcumntgl special mount_point
```

DESCRIPTION

The **sfcumntgl** unmounts all nodes sharing the same file system according to the specified operands. sfcfsd and MDS will be stopped.

The **sfcumntgl** can be executed in any nodes which are specified in the GFS configuration node data made by **sfcsetup**(1M).

Only a super user can perform unmount processing.

When unmount processing goes wrong, error information is outputted to the console of each node. Please perform management according to the outputted error information.

OPERANDS

The following operands can be specified:

special

Specifies the block special device name for a representative partition of the file system to be mounted.

mount_point

Specifies the name of the directory to be mounted.

EXAMPLES

```
# sfcumntgl /dev/sfdsk/gfs/dsk/volume2 /mount_point
# sfcumntgl /dev/sfdsk/gfs/dsk/volume2
# sfcumntgl /mount_point
```

FILES

/etc/mnttab

Table for the currently mounted file system.

/etc/vfstab

Environment file for the file system.

SEE ALSO

[sfcumntgl\(1M\)](#), [sfcsetup\(1M\)](#), [umount_sfcfs\(1M\)](#).

umount(1M), **mnttab(4)**, **vfstab(4)** in the "Solaris X Reference Manual Collection."

NOTES

- sfcfrmd daemon must be running to enable the execution of this command.
- When both *special* and *mount_point* are specified, they must also be described in **/etc/vfstab** in all node which shared a file system.

F.3 General Commands Specific to the GFS Shared File System

F.3.1 sfccp(1) Copies files with extent attribute information

SYNOPSIS

```
sfccp [-fip] [-e ext_action] source_file target_file  
sfccp [-fip] [-e ext_action] source_file... target  
sfccp -r|-R [-fip] [-e ext_action] source_dir... target
```

DESCRIPTION

The **sfccp** command copies both files with file extension attribute information or files without file extension attribute information on the GFS Shared File System. If necessary, the **sfccp** command will operate on a file system other than the GFS Shared File System, but in this case, using **cp**(1) is strongly recommended.

In format 1 shown in SYNOPSIS, neither *source_file* nor *target_file* must be a directory file, and they must not be specified with the same name. The **sfccp** utility copies the contents of the file indicated by *source_file* and the extent attribute information of the GFS Shared File System to the destination path indicated by *target_file*. If *target_file* already exists, **sfccp** overwrites the contents of the file. The mode (ACL if it is valid), owner, and group of the file, however, remain unchanged. For information about file extension attribute information, see the explanation of the **-e** option. The latest update time in *target_file* and the latest access time in *source_file* are set as the copy time. If *target_file* file is not found, **sfccp** creates a file named *target_file*. The mode of this file is the same as that of *source_file*. However, a sticky bit is not specified unless the user that issued this command is a super user. For the owner and group of *target_file*, the owner name and the group name of the owner are used. If *target_file* indicates a link to another file having a link, the other link remains and *target_file* assumed to be a new file.

In format 2, **sfccp** copies one or more files indicated by *source_file* to a directory indicated by *target*. For each *source_file*, a new file is created in *target*. in the same mode (the same ACL if it is valid). In this case, this system uses the owner and group of the user that issues this command. When the type specified in *source_file* is directory, *target* is not found, and *target* is not directory, processing fails.

In format 3, **sfccp** copies one or more directories indicated by *source_dir* to the directory indicated by *target*. Either **-r** or **-R** must be specified. **sfccp** copies all files and subdirectories in each directory indicated by *source_dir*.

OPTIONS

The following options can be specified:

-e *ext_option*

Specifies an operation for file extension attribute information. Explanation of *ext_option*

AWE

Append **W**rite **E**xtent (default).

When *target* and *target_file* already exist with file extension attribute information, this option does not copy the file extension attribute information. Otherwise, this option copies extent attribute information.

IWE

Interactive **W**rite **E**xtent.

When *target* and *target_file* already exist with file extension attribute information, this option displays a confirmation message asking whether the file extension attribute information should be overwritten. If the response is **y**, the file extension attribute information is copied. If the response is another character, the information is not written.

NWE

Not Write Extent.

This option does not copy file extension attribute information in the same way as **cp(1)**.

OWE

OverWrite Extent.

This option forcibly copies the file extension attribute information of *source* to *target* and *target_file*. Even though the *source* does not contain file extension attribute information, it is forcibly copied to *target* and *target_file*.

-f

Releases a link. If a file descriptor is not obtained for the destination file, this option releases the link of the destination file and attempts to continue processing.

-i

Indicates interactive mode. When *target* and *target_file* specified, as destination files already exist, this option displays a confirmation message asking whether the data should be overwritten. If the response is **y**, copy processing is continued. If the response is another character, the data is not overwritten. When both the **-i** option and **-e IWE** option are specified together, they are processed independently in interactive mode.

-p

Holds data. **sfccp** copies the contents of the file indicated by *source_file*, and also specifies an owner and group ID, access right mode, update time, and access time of the output file. If ACL is valid, the input file value is set as the ACL value. This command does not fail when it is not possible to specify an update time, access time, or access right mode. When the owner and group ID cannot be specified, **sfccp** does not fail, and clears **S_ISUID** and **S_ISGID** of the output file. If these bits cannot be cleared, **sfccp** outputs a diagnostic message to the standard error output and returns a termination status other than 0.

To store the owner, group ID, access right mode, update time, and access time, the user must have access permission to the appropriate file. In other words, the user must have the same owner ID as that of the *target* file.

-r

Recursive. **sfccp** copies to *target* the entire directory (that is, all files in the directory), and also copies all subdirectories in the directory and all files in the subdirectories.

-R

The same as **-r**, except that a named pipe file is copied, not read.

OPERANDS

The following operands can be specified:

source_file

Path name of an ordinary file to be copied

source_dir

Path name of a directory to be copied

target_file

Path name of the file used as the output when one file is copied regardless of whether the file already exists.

target

Path name of a directory to which a copied file is to be written

EXAMPLE

1. Examples in which one file is copied

```
% sfccp goodies goodies.old
% ls goodies*
goodies goodies.old
```

2. Examples in which one file group is copied to a directory

```
% sfccp ~/src/* /dst
```

F.3 General Commands Specific to the GFS Shared File System

3. Examples in which a specific directory is copied first to a new directory, and then to an existing directory

```
% ls ~/bkup
/usr/example/fred/bkup not found
% sfccp -r ~/src ~/bkup
% ls -R ~/bkup
x.c y.c z.sh
% sfccp -r ~/src ~/bkup
% ls -R ~/bkup
src x.c y.c z.sh
src:
x.c y.c z.sh
```

4. Example in which data is forcibly copied on the basis of source information when *target* has extent attribute information

```
% sfccp -e OWE source target
```

ENVIRONMENT

For information about environment variables **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**, which affect the execution of **sfccp**, see **environ(5)**.

EXIT STATUS

The following termination status information is returned:

- 0 All files have been copied successfully.
- >0 An error has occurred.

SEE ALSO

[sfccpio\(1\)](#), [sfcgetext\(1\)](#), [sfcgetfat\(1\)](#), [sfcmv\(1\)](#), [sfcsetext\(1\)](#), [sfcsettime\(1\)](#), [sfcsetvolume\(1\)](#).

cp(1), **chmod(1)**, **chown(1)**, **setfacl(1)**, **utime(2)**, **attributes(5)**, **environ(5)**, **largefile(5)** in the "Solaris X Reference Manual Collection."

NOTES

- The access right mode of *source_file* is stored when copy is executed.
- Users can clearly specify the end of a command line option with two consecutive hyphens (--). When this method is used and a file name beginning with a single hyphen (-) is specified as the filename argument, **sfccp** recognizes that the character string indicates a file name. If both -- and - are written on the same line, **sfccp** interprets the second hyphen as specifying a file name.
- **sfccp** can copy data between different file systems, and data with file extension attribute information from/to GFS Local File System. If a file on the GFS Shared File System is copied to the other file system (for example, UFS), however, file extension attribute information is deleted.
- **sfccp** will operate on a file system other than the GFS Shared File System and GFS Local File System, but its operation will be equivalent to the conventional **cp(1)**.

F.3.2 sfccpio(1) Extracts or restores files from an archive with extent attribute information

SYNOPSIS

```
sfccpio -i[bBcdfkmPrsStuvV] [-C bufsize ] [-E file ] [-I file [-M message ]] [-R userid ] [pattern ...]
sfccpio -o[aABcLPvV] [-C bufsize ] [-O file [-M message ]]
sfccpio -p[adILmPuvV] [-R userid ] directory
```

DESCRIPTION

The **sfccpio** command copies both files containing file extension attribute information or files not containing file extension attribute information from the GFS Shared File System to the sfccpio archive and vice versa.

The **sfccpio** command also operates on a system other than the GFS Shared File System. In this case, using **cpio**(1) is strongly recommended. For information about the relationships between the **sfccpio** and **cpio** commands or the operations to be performed when the sfccpio archive is restored to a system other than the GFS Shared File System, see NOTES below.

The **-i** , **-o** , and **-p** options are used to select the processing to be executed. The type of processing for each option is explained in detail. (These types of processing are mutually exclusive.)

[Restore mode]

sfccpio -i (copy in) extracts a file using the output previously created by **sfccpio -o** as the standard input. Only files having the name specified in *pattern* are selected. For information about *pattern*, see **OPERANDS** and **sh**(1) below. The extracted file is created as defined by the conditions, and copied to the current directory on the basis of the options described below. The file access right is the same as that of the previous output of **sfccpio -o**. The owner and group are the same as the owner and group of the current user unless the current user is super-user. If the current user is a super-user, the owner and group are the same as the owner and group obtained from the output file previously processed with **sfccpio -o**. If a file to be created by **sfccpio -i** already exists and the version of the file is the same or a newer version, a message is displayed and the file is not replaced. (The **-u** option is used to replace an existing file regardless of the conditions.)"

[Storage mode]

sfccpio -o (copy out) reads a path name list from the standard input and copies their files to the standard output together with path names, status information, and file extension attribute information. When data is output, the output data is blocked on **512**-byte boundaries by default. If the **-B** or **-C** option is used, output data can be blocked using the user-specified block size or with the block size, depending on a device.

[Pass Mode]

sfccpio -p (pass) reads a path name list for file from the standard input, and creates a file according to the conditions, and copies it to a destination *directory* on the basis of the options below.

When data is output to a character-type device (**-o**) or input from a character-type device (**-i**), if the end of the medium (for example, end of a tape) is reached without the **-O** and **-I** option, **sfccpio** displays the following message:

To continue, type device/file name when ready.

To continue the processing, replace the medium, enter a character-type special device name (for example, `/dev/rmt/0m`), and press **RETURN** key. This function is used to instruct the **sfccpio** command to use different devices and to continue the processing. For example, when there are two tape units, this function is convenient for switching the current device to another device so that the **sfccpio** command can continue processing during replacement of a tape. (The **sfccpio** command terminates processing when the **RETURN** key is pressed.)

OPTIONS

The following options can be specified:

- i**
(copy in) **sfccpio -i** extracts a specified file from the standard input.
- o**
(copy out) **sfccpio -o** reads a path name list from the standard input and copies the file to the standard output.
- p**
(pass) **sfccpio -p** reads a path name list for a specified file from the standard input. After the **-o**, **-i**, or **-p** option is specified, the following options can be specified in any order.
- a**
After copying a file, this option resets the input file access time. If **sfccpio -pla** (cannot be specified together with **-m**) is specified, the access time of a linked file is not reset.
- A**
Adds a file to an archive. This option must be paired with the **-O** option. This option is valid only for files, floppy disks, or hard disk partition archives.
- b**
Reverses the byte order of each word. (Can only be used together with the **-i** option.)
- B**
Block data as **5120** bytes for data input/output. When the **-B** and the **-C** options are not used, the default buffer size is **512** bytes. The **-B** option is not used for the **-p** option. This is because the **-B** option is valid for input from or output to a `/dev/rmt/0m` a character-type special device.
- c**
Reads and writes header information in ASCII mode, taking into account portability. There are no restrictions on user ID or group ID for this header format. The **-c** option means using extension device numbers supported only in the SVR4-based systems.
- C bufsize**
Block input/output is in *bufsize* units. *bufsize* is replaced with a positive integer. When the **-B** and the **-C** options are not used, the default buffer size is **512** bytes. (The **-C** option is not used for the **-p** option; This is because the **-C** option is valid only for input to or output from a character-type special device such as `/dev/rmt/0m`.)
- d**
Create a directory as required.
- E file**
Specifies an (one filename per line) input file (*file*), including a file name list to be extracted from archive.
- f**
Extracts all files excluding those specified in *pattern*. (For information about *pattern*, see **OPERANDS** below.)
- i file**
Read the contents as an input archive. If the *file* is a character-type special device and it finishes reading all of the current medium, replace the medium and press the **RETURN** key to continue processing. This option can be used only with the **-i** option.
- k**
Skips a damaged file header or input-output error. To copy a file from a damaged medium or media that are out of sequence, this option allows only files having a normal header to be read. (When an **sfccpio** archive contains another **sfccpio** archive and an error occurs, the **sfccpio** command terminates. The **sfccpio** command searches for the next normal header. When a smaller archive is found, this command reads data until it encounters its trailer and then terminates processing.) The **-k** option can be used only with the **-i** option.
- l**
As much as possible, links files instead of copying them. (This option can be used only with the **-p** option.)
- L**
Traces a symbolic link. The default does not trace a symbolic link.

- m**
Fetches the previous file update time. This option is invalid if a directory is being copied. (This option cannot be used with the **-a** option.)
- M message**
Defines a *message* for replacement of a medium. When a character-type special device is specified with the **-O** or **-I** option, this option defines a message that is displayed at the end of a medium. One **%d** can be used to display the sequence number of the next medium.
- O file**
Changes an **sfccpio** output destination to *file*. When *file* is a character-type special device and the current medium is full, replace the medium and press the **RETURN** key to continue processing. This option can be used only with the **-o** option.
- P**
Holds ACL. When this option is used for output and ACL exists, ACL is written to the standard output together with other attributes. ACL is created as a special file having a special file type. When this option is used for input, ACL is restored from the standard input together with other attributes. This option can identify a special file type. This option cannot be used with the **-c** option because ACL is not necessarily supported on all systems. Also, portability is not assured. To take into account portability, use an ASCII header.
- r**
Changes a file name in interactive mode. When only the **RETURN** key is pressed, the file is skipped. If "." is entered, the original path name is used. (This option is invalid for **sfccpio -p**.)
- R userid**
Changes the owner and group information for each file to a *user ID* (id must be a valid login ID in the */etc/passwd* file). Only a super user can use this option.
- s**
Substitutes bytes for each half word.
- S**
Substitutes a half word for each word.
- t**
Displays the entered character strings. No file is then created. (This option cannot be used with **-V** option.)
- u**
Executes copying regardless of the condition. (An old file is not ordinarily replaced with a new one.)
- v**
Displays detail information and lists file names. When the **-v** option is used together with the **-t** option, data is output the same way as for the **ls -l** command (see **ls(1)**).
- V**
Special verbose. Print a dot for each file read or written. Useful to assure the user that **sfccpio** is working without printing out all file names.

OPERANDS

The following operands can be specified.

directory

Path name of an existing directory to be processed by **sfccpio -p**.

pattern

Formula using a representation method for pattern matching. This formula is similar to the one (see **sh(1)**) the shell uses for file name pattern matching or regular representation. The following meta-characters are defined:

*

Matches all character strings, including an empty character string.

?

Matches any 1 character.

[...]

Matches one of the characters enclosed with braces. When two characters are delimited by a

F.3 General Commands Specific to the GFS Shared File System

hyphen '-', they match all characters (including those two characters) between two characters according to the default collation sequence of the system. If the character following the first bracket '[' is a '!', the results is not assured.

!

Indicates a negation. For example, the !abc* is specified, the pattern matches a file name that does not begin with character string abc.

In *patterns*, meta-characters ?, *, and [...] match a slash (/) and a backslash (\) is an escape character. More than one *pattern* can be specified. If no character is specified in *pattern*, the default is * (that is, all files are selected).

Each pattern must be enclosed in double quotation marks (") If it is not, a file in the current directory may be used.

USAGE

A file of two gigabytes or more cannot be stored. Also, an archive of eight gigabytes or more cannot be created.

EXAMPLES

The following are three **sfccpio** examples:

When the standard input is transferred to **sfccpio -o**, via a pipe, files are grouped and collected as one archive file (../newfile). The **-c** option ensures compatibility with a different machine from the machine with the archive file. To transfer a file name list to **sfccpio -o**, via a pipe, **find(1)**, **echo(1)**, and **cat(1)**, can be used instead of **ls(1)**. If necessary, the output destination can be changed from a file to a device.

```
% ls | sfccpio -oc &gt; ../newfile
```

sfccpio -i extracts a file that matches a specified pattern (memo/a1, memo/b*), using the output of **sfccpio -o** (in the example below, transferred to a pipe using **cat**). This command also creates a directory under the current directory (**-d** option) as required, and stores the file in the appropriate directory. The **-c** option is available when an input file is created with a compatible header. If no pattern is specified, all files in newfile are stored in the directory.

```
% cat newfile | sfccpio -icd "memo/a1" "memo/b*"
```

sfccpio -p reads file names via a pipe and copies or links (**-l** option) those files to another directory (in the example below, newdir). The **-d** option means that a directory is created as required. The **-m** option fetches a modification time. To create a path name required to transfer data to **sfccpio**, use the **-depth** option of **find(1)** command. With this function, the user can eliminate a problem that occurs when an attempt is made to create files under a read-only directory. The copy destination directory newdir is necessary.

```
% find . -depth -print | sfccpio -pdlmv newdir
```

When **sfccpio** is used together with **find** and **-L** option is specified in the **sfccpio** option, use **-follow** in the **find** option and vice versa. If **-follow** is not used, an unexpected result is displayed.

Note that for multi-tape archives, unload the old tape, load the new one, and continue to the next tape by typing the name of the next device (probably the same as the first tape). To stop, type a **RETURN** and **sfccpio** will end.

ENVIRONMENT

For information about environment variables **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, **LC_TIME**, **TZ**, and **NLSPATH**, see **environ(5)**.

TERMINATION STATUS

The following termination status information is returned:

0 Terminated normally.

>0 An error occurred.

SEE ALSO

[sfccp\(1\)](#), [sfcgetext\(1\)](#), [sfcgetfattr\(1\)](#), [sfcmv\(1\)](#), [sfcsetext\(1\)](#), [sfcsettime\(1\)](#), [sfcsetvolume\(1\)](#), [ar\(1\)](#), [cat\(1\)](#), [cpio\(1\)](#), [echo\(1\)](#), [find\(1\)](#), [ls\(1\)](#), [setfacl\(1\)](#), [sh\(1\)](#), [tar\(1\)](#), [vold\(1M\)](#), [archives\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#) in the "Solaris X Reference Manual Collection."

NOTES

- The path name is limited to **256** bytes (default) in binary notation; otherwise, limited to **1,024** bytes.
- If a file has too large a user ID or group ID for a selected header format, an error message is displayed. To create an archive that can support all the user ID or group ID values, use the **-c** option.
- Only a super user can copy a special file.
- **One** block is **512** bytes.
- When the access right of a specific file is **000**, the file contains character data; and the user is not root, the file is not stored.
- While the volume management daemon is operating, in some cases a floppy disk unit cannot be accessed using a normal device name such as `/dev/rdiskette`. For details, see [vold\(1M\)](#).
- To fetch and copy a file from a tape to hard disk, use the same blocking factor as that used to copy the file from hard disk to the tape. Therefore, specify the **-B** option.
- **sfccpio** can read only an archive in the **sfccpio** format. An archive in the **cpio** format cannot be read.
- **sfccpio** can operate between different file systems, and can copy with extent attribute information from/to GFS Local File System. If a file on the GFS Shared File System is copied to the other file system (for example, UFS), extent attribute information is erased.
- **sfccpio** will operate on a file system other than the GFS Shared File System and GFS Local File System, but its operation will be equivalent to the conventional **cpio(1)**.

F.3.3 sfcgetext(1) Displays extent attribute information

SYNOPSIS

sfcgetext [-iH] *file...*

DESCRIPTION

The **sfcgetext** command displays extent attribute information about a specified file on the GFS Shared File System. The details of the extent attribute information are listed below. Note, however, that this information is not displayed when the specified file is not on the GFS Shared File System or its file attribute is not ordinary file or directory.

- File system block size (in kilobyte units)
- Initial number of blocks
- Incremental number of blocks
- Allocation flag
- File name

For information about the initial number of blocks, increment number of blocks, and allocation flag, see [sfcsetext\(1\)](#).

OPTIONS

The following options can be specified:

- H**
Displays normal data after a column header.
- i**
This option disables display when no extent attribute information is defined for a specified file and lists only files that have extent attribute information.

OPERAND

The following operand can be specified:

- file...**
Specifies a general file to target for the transaction or path name of the directory.

EXAMPLES

```
% sfcgetext /mnt/file1 /mnt/file2 /mnt/file3 /mnt/file4 /mnt/file5
 8    1024    512  contig    /mnt/file1
 8    2048     0  noextend  /mnt/file2
 8    512     0  contig,noextend /mnt/file3
 8    512     0  -          /mnt/file4
 8     -     -  -          /mnt/file5

% sfcgetext -i /mnt/file1 /mnt/file2 /mnt/file3 /mnt/file4 /mnt/file5
 8    1024    512  contig    /mnt/file1
 8    2048     0  noextend  /mnt/file2
 8    512     0  contig,noextend /mnt/file3
 8    512     0  -          /mnt/file4

% sfcgetext -iH /mnt/file1 /mnt/file2 /mnt/file3 /mnt/file4 /mnt/file5
Blocksize(KB) init-blocks inc-blocks  flags    filename
 8    1024    512  contig    /mnt/file1
 8    2048     0  noextend  /mnt/file2
 8    512     0  contig,noextend /mnt/file3
```

8	512	0	-	/mnt/file4
---	-----	---	---	------------

SEE ALSO

[sfcsetext\(1\)](#), [sfccpio\(1\)](#), [sfccp\(1\)](#), [sfcmv\(1\)](#).

statvfs(2) in the "Solaris X Reference Manual Collection."

F.3.4 sfcgetfattr(1) Reports extent attribute information of a file

SYNOPSIS

sfcgetfattr [-iHDev] *file*

DESCRIPTION

sfcgetfattr displays the extended attribute information of file in a GFS Shared File System.

Reports the following information: However, when the specified files are files other than a GFS Shared File System, and when a file is usually except a regular file and a directory, file extended attribute information is not displayed.

Extent attribute information:

- block size of file system (kilo byte unit)
- Initial number of blocks
- Incremental number of blocks
- Allocation flag

Ignore update time attribute information:

- Ignore update last access time attribute
- Ignore update last modify time attribute

file data allocation attribute information:

- partition number
- Special file name (when **-D** options is specified)

For details of the initial number of blocks, incremental number of blocks, and Allocation flag, see [sfcsetext\(1\)](#).

When no option is given, **sfcgetfattr** displays all extended attribute information with column header. Which the same operation as the case where **-iHDev** are specified.

OPTIONS

The following options can be specified:

- D**
Report the disk partition, when **-v** option is specified.
- e**
Reports the extent attribute information of *file*.
- H**
Print column header.
- i**
Suppress the message printed on the standard error when *file* has no the extended attribute information.
- t**
Reports the ignore update time attribute information of *file*.
- v**
Reports the file data allocation attribute information of *file*.

OPERAND

The following operand can be specified:

file

A path names of a regular file or a directory whose a information to be written.

EXAMPLES

```
% sfcgetfattr /mnt/extentfile
Blocksize(KB) init-blocks inc-blocks  flags
      8      1024      512  contig
% sfcgetfattr -iHe /mnt/extentfile
Blocksize(KB) init-blocks inc-blocks  flags
      8      1024      512  contig
% sfcgetfattr -ie /mnt/extentfile
      8      1024      512  contig
% sfcgetfattr -e /mnt/extentfile
      8      1024      512  contig
% sfcgetfattr -iHt /mnt/notimeupdatefile
atime      mtime
noatime    -
% sfcgetfattr -iHDv /mnt/allocationfile
vol  devname
  2  /dev/sfdsk/gfs/rdisk/volume03
```

SEE ALSO

[sfccp\(1\)](#), [sfccpio\(1\)](#), [sfcgettext\(1\)](#), [sfcmv\(1\)](#), [sfcsettext\(1\)](#), [sfcsettime\(1\)](#), [sfcsetvolume\(1\)](#).

F.3.5 sfcmv(1) Moves files with extent attribute information

SYNOPSIS

sfcmv [-fi] *source target_file*
sfcmv [-fi] *source... target_dir*

DESCRIPTION

The **sfcmv** command moves a file with or without file extension attribute information on the GFS Shared File System. The **sfcmv** command also operates on the other file system, but in this case, using **mv**(1) is strongly recommended.

In format 1 above, the **sfcmv** utility moves the file indicated by the *source* operand to the destination indicated by the *target_file*. The same name cannot be specified for *source* and *target_file*. If *target_file* is not found, **sfcmv** creates a new file with the same name. If *target_file* is found, its contents are overwritten. When the last operand does not indicate an existing directory, **sfcmv** is assumed to be in format 1.

In format 2, the **sfcmv** utility moves each file specified in the *source* operand to each file in an existing directory indicated in the *target_dir* operand. In the destination path name for a *source*, a slash (/) and the last path name of *source* are added after the destination directory name. When the last operand indicates an existing directory, **sfcmv** is assumed to be in format 2.

When *target_file* is identified in write-disabled mode, **sfcmv** displays a mode (see **chmod**(2)), requests a response, and reads one line from the standard input. If the response is positive and **sfcmv** has the appropriate permission, it moves the file. Otherwise, this command terminates. When *target_file* makes ACL valid, the access right is not completely displayed in mode. If the parent directory of *source* is in write-enabled mode with a sticky bit set, one or more of the following conditions must be true:

- The user must own the file.
- The user must own directories.
- The file must be in user-write-enabled mode.
- The user must be a privileged user.

When *source* indicates a file and *target_file* indicates a link to another file having a link, the other link remains, and *target_file* changes to a new file.

OPTIONS

The following options can be specified:

-f

sfcmv moves a file without displaying a prompt even when it writes data to an existing *target*. Note that this option is the default if the standard input is not a terminal.

-i

sfcmv always displays a confirmation prompt when an existing *target* is overwritten in a move operation. If the response is y, the move processing continues. If the response is another character, **sfcmv** does not overwrite *target*.

When both the **-f** and **-i** options are specified, they are not assumed to be illegal. Instead, the **-f** option is valid.

OPERANDS

The following operands can be specified:

source

Path name of a file or directory to be moved

target_file

A new path name of a file or directory to be moved
target_dir
Path name of an existing directory used as the destination of the move file operation

ENVIRONMENT

For information about environment variables **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**, which affect the execution of **sfcmv** see **environ(5)**.

EXIT STATUS

The following termination status information is returned:
0 All input files have been moved correctly.
>0 An error has occurred.

SEE ALSO

[sfccp\(1\)](#), [sfccpio\(1\)](#), [sfcgetext\(1\)](#), [sfcgetfattr\(1\)](#), [sfcsetext\(1\)](#), [sfcsettime\(1\)](#), [sfcsetvolume\(1\)](#).

cp(1), **cpio(1)**, **ln(1)**, **mv(1)**, **rm(1)**, **setfacl(1)**, **chmod(2)**, **attributes(5)**, **environ(5)**, **largefile(5)** in the "Solaris X Reference Manual Collection."

NOTES

- When *source* and *target_dir* are on different file systems, **sfcmv** copies the file and deletes the originals; In this case, all links to other files are deleted.
- Since users can clearly specify the end of a command line option with two consecutive hyphens (--), **sfcmv** can identify a file name argument that begins with a single hyphen (-). For a move operation in BSD, **sfcmv** accepts a single hyphen (-) as a synonym of two hyphens (--). Note, however, that the BSD move function may be deleted in future releases. If both two consecutive hyphens (--) and a single hyphen (-) are specified on the same line, the latter is interpreted as specifying a file name.
- When *target* is specified, use [sfccp\(1\)](#) to store extent attribute information for *target* (with **-e NWE** specified).
- **sfcmv** can move a file between different file systems, and a file with extent attribute information from/to GFS Local File System. If a file on the GFS Shared File System is moved to the other file system (for example, UFS), however, extent attribute information is deleted.
- **sfcmv** will operate on a file system other than the GFS Shared File System and GFS Local File System, but its operation will be equivalent to the conventional **mv(1)**.

F.3.6 sfcsetext(1) Specifies extent attribute information

SYNOPSIS

sfcsetext -a *init-blocks*[,*inc-blocks*] [-c] [-f *flags*] *file*...

DESCRIPTION

The **sfcsetext** command specifies extent attribute information for a specified *file* on the GFS Shared File System.

After creating the specified *file*, this command specifies extent attribute information according to the specified options. In this case, the initial area allocated by the number of blocks specified in *init-blocks* is allocated.

However, extent attribute information cannot be specified when the specified *file* already exists or does not belong to the GFS Shared File System.

After the extent attribute information is set to a directory, files and subdirectories created in the directory inherit the file data allocation attribute information of the directory.

OPTIONS

The following options can be specified:

-a *init-blocks*[,*inc-blocks*]

init-blocks specifies the number of blocks allocated to the file as the initial area. **Zero (0)** cannot be specified.

inc-blocks specifies the increment (in blocks) when write data exceeds the initial area. If **0** is specified for *inc-blocks* or *inc-blocks* is not specified, the normal allocation is used.

When **noextend** is specified using the **-f** option, either specify **0** for *inc-blocks* or do not specify the option.

init-blocks and *inc-blocks* must be specified in the unit of blocks (eight kilobytes) used for the file system.

-c

Set extent attribute only. New file is specified without **-c** option, it will be created with the parent directory's policy attributes and extent attribute. When a directory specified with **-c**, other policy attributes that was set to it are cleared, and only extent attribute is set. When a directory specified without **-c**, it will have existing policy attribute and extent attribute.

-f *flags*

This option specifies the data area allocation method. The combination of the *flags* below can be specified for flags. To specify more than one flag, use a comma (,) as the delimiter.

contig

Allocates a contiguous area for the number of blocks specified in *init-blocks* and *inc-blocks*. Depending on the number of blocks specified in *inc-blocks*, the *init-blocks* area is allocated as shown below.

If 0 is specified:

Allocates a contiguous area for the number of blocks specified in *init-blocks*. If it cannot be allocated, then error information is returned.

If less than *init-blocks*:

Searches for a contiguous space that is equivalent to the number of blocks specified in *init-blocks*. If it cannot be found, the system allocates a contiguous area for *init-blocks* using the blocks specified in *inc-blocks*. If the specified value in *inc-blocks* is not a multiple of *init-blocks*, the system allocates a contiguous area for the number of blocks specified in *inc-blocks*, and then allocates a contiguous area for the remaining (including fractional) blocks. If it cannot be allocated, then error information is returned.

If greater than *init-blocks*:

Allocates a contiguous area for the number of blocks specified in *init-blocks*. If it cannot be allocated, then error information is returned.

noextend

If the write data exceeds the number of blocks specified in *init-blocks* or the file size is changed, additional area is not allocated, and error information is returned.

If the **-f** option or **contig** is not specified, the system does not necessarily allocate a contiguous area for the number of blocks specified in *init-blocks* or *inc-blocks*.

OPERAND

The following operand can be specified:

file...

Specifies a general file to target for the transaction or path name of the directory.

EXAMPLES

```
% sfcsetext -a 1024,512 -f contig /mnt/file1
% sfcsetext -a 2048 -f noextend /mnt/file2
% sfcsetext -a 512 -f contig,noextend /mnt/file3
% sfcsetext -a 512 /mnt/file4
```

SEE ALSO

[sfccp\(1\)](#), [sfccpio\(1\)](#), [sfcgetext\(1\)](#), [sfcmv\(1\)](#), [sfcsetvolume\(1M\)](#),
[statvfs\(2\)](#) in the "Solaris X Reference Manual Collection."

NOTES

- If **contig** is specified using the **-f** option, even though the file system has sufficient space, area allocation fails because of the specified option values.
- For *init-blocks* and *inc-blocks*, think about the value to be specified as the required number of blocks. If the number of blocks is unnecessarily large, used area increases, causing disk resources to be used inefficiently.

F.3.7 sfcsettime(1) Sets or deletes ignore update time attribute information

SYNOPSIS

sfcsettime {+|-}{a|m|am} *file*

DESCRIPTION

sfcsettime set the ignore update time attribute information to file in the GFS Shared File System.

After the ignore update time attribute information is set to a directory, files and subdirectories created in the directory inherit the attribute information of the directory.

The ignore update time attribute information cannot set, when file except a regular file, a directory, and a file in the GFS Shared File System are specified.

OPTIONS

The following options can be specified:

- +a** Set the ignore update last access time attribute information to file.
- a** Delete the ignore update last access time attribute information from file.
- +m** Set the ignore update last modify time attribute information to file.
- m** Delete the ignore update last modify time attribute information from file.
- +am** Set the ignore update last access time and modify time attribute information from file.
- am** Delete the ignore update last access time and modify time attribute information from file.

OPERANDS

The following operand can be specified:

- file*** A path name of a regular file or a directory that the ignore update time attribute information to be set.

EXAMPLES

```
% sfcsettime +a /mnt/accesstime
% sfcsettime +m /mnt/modifytime
% sfcsettime +am /mnt/accessmodifytime
% sfcsettime -m /mnt/accessmodifytime
```

SEE ALSO

[sfccp\(1\)](#), [sfccpio\(1\)](#), [sfcmv\(1\)](#), [sfcgetfattr\(1\)](#).

find(1), **ls(1)**, **touch(1)**, **stat(2)** in the " *Solaris X Reference Manual Collection* "

F.3.8 sfcsetvolume(1) Sets or deletes file data allocation attribute information

SYNOPSIS

```
sfcsetvolume -D file  
sfcsetvolume -v raw [ -c ] [-f mandatory] file
```

DESCRIPTION

sfcsetvolume set the file data allocation attribute information to file in the GFS Shared File System.

The file data allocation attribute information cannot set, when file except a regular file, a directory, and a file in the GFS Shared File System are specified.

After the file data allocation attribute information is set to a directory, files and subdirectories created in the directory inherit the file data allocation attribute information of the directory.

OPTIONS

The following options can be specified:

-D

Delete the file data allocation attribute information from file.

-c

Set file data allocation attribute only. New file is specified without **-c** option, it will be created with the parent directory's policy attributes and file data allocation attribute. When a directory specified with **-c**, other policy attributes that was set to it are cleared, and only file data allocation attribute is set. When a directory specified without **-c**, it will have existing policy attribute and file data allocation attribute.

-f *mandatory*

Set the file data allocation policy to mandatory. Data blocks are allocated only from file data partition specified in the **-v** option. Return the error when a data block cannot be allocated for a file from specified file data partition.

The default is advisory when **-f** option not specified. When a data block cannot be allocated from specified file data partition, a data block will be allocated from a file data partition that have space.

-v *raw*

Specify the character-special device of the file data partition that is allocated data block for file.

OPERANDS

The following operand can be specified:

file

A path name of a regular file or a directory that the file data allocation attribute information to be set.

EXAMPLES

```
% sfcsetvolume -v /dev/sfdsk/c1/rdsk/v1 /mnt/file1
```

SEE ALSO

[sfcset\(1\)](#), [sfcsettext\(1\)](#), [sfcgetattr\(1\)](#), [sfcmv\(1\)](#), [sfcsettext\(1\)](#), [sfcsettext\(1\)](#).

Appendix G Trouble shooting

This appendix explains messages for which emergency action is required.

G.1 Action for a Panic

Please gather the crash dump file as trouble investigation material when the node does the panic and the crash dump is gathered.

When the node starts after the panic, the crash dump is preserved as a file of the couple named **unix.X** and **vmcore.X**. The directory that crash dump file stored is ***"/var/crash/node name"*** in default. Please confirm the directory with **dumpadm(1M)** preservation ahead.



Note

For more information about crash dump, see "Solaris X System Administrator Guide" of Solaris OE.

G.2 Action for Error Messages

This section explains error messages that require urgent response in the GFS Shared File System.



For more information on how to manage command, daemon, and kernel messages, see [Appendix E](#) in this manual.

For more information on how to respond to PRIMECLUSTER error, see "11. The message and code of CF" in the *"PRIMECLUSTER Configuration and Administration Guide"*

G.2.1 Action for I/O errors

G.2.1.1 If message "sfdfs...:read error" or "sfdfs...:write error" is output

The "sfdfs...:read error" message indicates that an I/O error occurred while file system data was being read from a disk. The "sfdfs...:write error" message indicates that an I/O error occurred while file system data was being written to a disk.

If the messages are displayed, recovery processing is required because an I/O path or device is faulty.

Perform the recovery processing using the following procedure:

1. Locate the file system and partition in which the error occurred.

To do so, enter the GFS Shared File System management command `sfinfo(1M)` using the device number in the error message as key.

```
# sfinfo -d device-number
```

2. If an `sfdfs...:write error` occurs

This error indicates that file system information updated in memory cannot be reflected on the disk. If this happens, files in the file system should be backed up before unmounting to prevent the updated information from being lost by unmounting.

3. Stop the file system service in which the error occurred.
Stop the file system service.

4. Correct the file system in which the error occurred.

Make a copy of the file system data in the partition where the error occurred, replace the partition, then use `fsck_sfdfs(1M)` to restore the consistency of the file system. Issue `fsck_sfdfs(1M)` while specifying to perform a full check.

```
# fsck -F sfdfs -o nolog special-device-name
```

5. If the file system in which the error occurred cannot be corrected

If the file system cannot be corrected with step 4, replace the faulty part, re-create the file system, and restore the backed up data.

G.2.2 Collecting Investigation Material

Collect following material for investigation when you request troubleshooting to local Customer Support and if your manual instructs you to do so.

1. Material acquired in `/opt/FJSVsnap/bin/fjsnap -a filename` (Need to be executed with root access privilege)
2. `/var/crash/*/unix.*`
3. `/var/crash/*/vmcore.*`
4. Output of `fsck -F sfcfs -N -o nolog raw_device` * If the file system needs to be recovered.

If the investigation material must urgently be sent, prepare the followings at least. This material is a part of material collected by `fjsnap` and required for the initial investigation of GFS Shared file system.

- Material collected by `/etc/opt/FJSVsfscfs/bin/sfcsnap.sh -a -d dirname` (Collect by executing with a root access privilege)
- `/var/adm/messages*`



Note

- All material above should be collected from all nodes where the GFS Shared file system is shared.
- This material should be collected as soon as the symptom occurred. Some material might not be collected after rebooting the system.
- For the directory `dirname` designated in `sfcsnap.sh`, specify the existing directory. Material is stored in the sub-directory.
- Above-mentioned material is only for initial investigation. Be sure to collect all what are described in No.1 to 4.

G.3 GFS Shared File System Management View Abnormality

See “*PRIMECLUSTER Installation Guide*” for details of the GFS Management View abnormality.

Glossary

64-bit file system interface

This interface can handle files that exceed 2 gigabytes.

For details, see **largefile(5)** in the "*Solaris X Reference Manual Collection*" or the online manual.

AC

Access client. Module which operates in the kernel layer (file system layer) on each node sharing GFS Shared File System. An AC communicates with the MDS if a user application operating on the node sends an access request in a system call. File operations are performed using meta-data sent from the MDS and direct access to the real.

ACL

ACL is a file security system that specifies file access authority for specific users or groups in addition to the owner and owner group. For details, see **setfacl(1)** in the "*Solaris X Reference Manual Collection*" or the online manual.

alternate file system

The file system that made on the shared disk device is exclusive used in active node and the standing node in RMS Cluster.

area management data

Area management data includes information on i-nodes, V-data, and file data allocation information. It is stored in the meta-data area.

array disk

Disk file unit in which two or more physical disks are grouped as a single virtual disk. By dividing access between two or more physical disks, performance improves and high reliability can be implemented. Ordinarily, two or more access methods, which are called redundant array of inexpensive disks (RAID) levels, are provided. The major RAID levels are RAID 0 (striping), RAID 1 (mirroring), RAID 5 (striping with parity), and RAID 0+1 (striping set mirroring).

copy of partition structure information

If a block failure on a disk device destroys partition structure information, a copy is available for continued processing of the file system.

cylinder group

A cylinder group consists of a single disk cylinder or multiple contiguous disk cylinders that have i-node slots for files.

data

The term data refers to file data.

data area

The term data area refers to a file data area.

data partition

The term data area refers to a file data partition.

direct I/O

Direct I/O refers to a mode of data input/output that does not use kernel buffering.

EC

The abbreviation for Equivalent Copy. The hardware function of the disk array equipment (ETERNUS3000/GR720/GR730/GR740) of our company. EC is the function to hold the mirror state between disks within hardware. The duplicate of a disk is created by separation operation of a mirror state. In the case of EC employment in high-speed backup restoration of the GFS Shared File System by **sfcproxyop(1M)** and **sfcproxyrestore(1M)**, when possible, this hardware function is used.

EC employment

The employment pattern that incorporates (shift to an equivalent state), creates a snapshot by separation in high-speed backup restoration of the GFS Shared File System which used **sfcproxyop(1M)** and **sfcproxyrestore(1M)**, and performs backup restoration.

extent

An extent is a group of contiguous data blocks of the file system that are handled as a single unit. Each extent is defined by its start block and length.

extent attribute information

The information data of file include data that allocate a data area beforehand to a file, extend a data area in a certain proportion, and guarantee data area continuity.

extent-base

The extent base is a method of managing file system data based on extents.

fibre channel

One of the interfaces using optical fibre cables as transmission media. In a cluster system, a fibre channel is used to connect a array disk and the computer.

file data

File data is the data part of a file.

file data allocation attribute information

Property information used to specify allocation from a specific file data partition and to place a block as a normal file data.

file data area

The file data area in disk partitions contains file data. It is also called data area. The file data area is allocated in the representative partition and the file data partitions.

file data partition

The file data partition is a partition that consists of a file data area (data area).

The file data partitions are added to an existing file system in order to extend it. The file data partition will be specified in file data allocation attribute information.

file lock

File lock is a means for synchronization among processes that concurrently access data in the same file. There are two types of file locks: mandatory locks and advisory locks.

file system driver

Software driver to control a GFS Shared File System, located in a kernel like AC for data exchange between commands and AC.

GDS

GDS(Global Disk Services).

Volume management software for protecting data by managing a disk storage device totally within a SAN environment. For details, refer to "*PRIMECLUSTER Global Disk Services Configuration and Administration Guide.*"

GFS Local File System

A highly reliable and efficient file system utilizing the log mechanism and extent-base area management. For details, refer to "*Part 2 Global File Services Local File System*" in this manual.

GFS Management View

Software which allows various operations including GFS Shared File System configuration setting and operation management using the Web screen.

GLS

GLS(Global Link Services) software provides High Reliability of network communications by duplexing network communication paths using redundant NIC(Network Interface Card)s. For details, refer to "*PRIMECLUSTER Global Link Services Configuration and Administration Guide.*"

i-node

The i-node contains information on the file locations on the disk, on file sizes, and on the time when the respective files were last. It also includes other information for identification.

i-node area

The i-node area contains an i-node. It is located in the meta-data area.

instant snapshot

A snapshot of a master instantly created by cooperation with the OPC (One Point Copy) function provided by a GR700 series disk array.

log partition

Same as the update log partition

management partition

Required information is managed when employing the GFS Shared File System. Moreover, this is used in order for sfcfrmd daemon to maintain informational adjustment. The following information is included in the management partition.

- 1.Information on each node that constitutes the GFS Shared File System.
- 2.Correspondence information on common node of each GFS Shared File System, and shared device.

3.MDS arrangement information on each GFS Shared File System.

MDS

Meta-data server. Program that centrally manages the control information of a file system (meta-data). This program operates on the primary server and secondary server. Each AC communicates with the MDS and performs file operations according to the meta-data sent from the MDS.

MDS node

The node with the possibility that MDS performs. MDS performs in either node that it is defined as MDS node in nodes that share the file system. The default MDS node is 2 for the primary and secondary MDS.

meta area

Same as the meta-data area

meta-data

Various types of management data for the file system are collectively referred to as meta-data. Meta-data includes information on free areas, i-nodes, indirect extent blocks, directories, symbolic links, data of quota and data of ACL.

meta-data area

The meta-data area contains meta-data. It is allocated in the representative partition.

meta-data partition

Partition composed only of a meta-data area to use file system extension.

MPHD disk

A virtual device with multiple access paths for the disk array unit. An access path failure will not halt operation.

node

A node is a computer system that runs under one operating system.

node identifier

The node identifier is a universal name that is assigned to each node in the PRIMECLUSTER system. For each node, the node identifier is normally the same as the node name. However, if the node name was assigned to the standby node such as in hot standby mode, the correspondence between physical name and node name is temporarily disrupted. In this event, the node can be managed via the node identifier, which is the name for the physical device. The node identifier is displayed as each node name using PRIMECLUSTER Management View.

node name

The node name is the name of a node that is displayed with the "uname -n" command and identified in the communication network. For details, see **uname(1)** in the "*Solaris X Reference Manual Collection*" or the online manual.

OPC

The abbreviation for One Point Copy. The hardware function of the disk array equipment (ETERNUS3000/GR720/GR730/GR740) of our company. It is the function which the copy of a disk can carry out within hardware at high speed. Copy operation is completed in an instant and the emulation of the completion state of a copy is carried out about a non-copied domain. It is a function indispensable in the case of the OPC employment in high-speed backup restoration of the GFS Shared File System by **sfcproxyop(1M)** and **sfcproxyrestore(1M)**.

OPC employment

The employment pattern that performs backup restoration by the instant copy in high-speed backup restoration of the GFS Shared File System which used **sfcproxyop(1M)** and **sfcproxyrestore(1M)**. When using this employment pattern, it will be the requisite that the OPC function of our company disk array equipment ETERNUS3000/GR720/GR730/GR740 can be used.

partition

The partition is a physical or logical disk slice.

partition configuration data

This information refers to the partitions that constitute the GFS Shared File System. It includes information about the size of the meta-data area, update log area, file data area (data area) and on their layout in the configuration partition.

primary management server

Node on which the management server daemon of the Web-Based Admin View operates during normal operation. The primary management server uses the menus of the GFS management view to accept operation and display requests from the client and to transfer the requests to the monitored nodes.

primary server

Node on which the MDS operates during normal operation in a cluster system. This server should be set during file system registration.

PRIMECLUSTER

Software to configure the Cluster System. For details, refer to "*PRIMECLUSTER Concept Guide*."

quota function

quota is a function for limiting disk space or the number of i-nodes that can be used by the user.

representative partition

The representative partition is a GFS Shared File System partition that consists of meta-data, update log, and file data (data) areas. It may also consist of meta-data and file data (data) areas.

Remote console connection unit

A device used to connect an RS-232C interface to a LAN interface. With this device, users can use the Telnet function to access the TTY console function from other devices (personal computers) connected through a LAN.

RMS

Abbreviation for Reliant(R) Monitor Software. Software that maintains high availability of user-specified resources by providing monitoring and switchover capabilities.

round robin allocation

Round robin allocation is a file data allocation mode for allocating separate file data to multiple file data areas.

scope

A range of node set that can share an object when a disk class is shared.

secondary management server

Node on which the Web-Based Admin View management server daemon operates if an error occurs on the primary management server. A secondary management server is set when the Web-Based Admin View operating environment is set. If an error occurs on the primary management server, the management server function of the Web-Based Admin View passes to the secondary management server.

secondary server

Server that takes over the MDS function if an error occurs on the primary server in a cluster system. The secondary server must be set during file system registration before it can assume the MDS function when an error occurs on the primary server. The file system status is restored to assure contiguous access from each AC.

sfcfrmd

The demand from the commands or sfcfsd daemon is received and the communication way for realizing commands sfcfsd, commands sfcprmd, and communication between nodes is offered. Management of a composition database and a management partition I/O function are offered.

sfcfsd

Employment start and employment stop of the GFS Shared File System, taking over employment (failover) of MDS, an on-line node addition and deletion, file system composition cut return in the definition state (my gray SHON), etc. offer the function which controls MDS and AC.

sfcprmd

Starting and stop, down surveillance of the process which constitutes a GFS Shared File System is performed. When a composition process starts and executes other demons and commands, a demand is transmitted to sfcprmd and sfcprmd executes a demon and a command according to a demand.

The process which transmits a demand is called client process and the process started from sfcprmd according to a demand is called target process.

A target process notifies the situation to a client process, when it is supervised by sfcprmd and a process carries out a normal end and unusual end.

shared disk

One of operation modes of a shared disk device. Applications in each node can simultaneously access a shared disk device on their own. (Note that the terms shared disk device and shared disk have different meanings. To distinguish between the two terms, a "shared disk" may be referred to as a "shared disk for simultaneous access.")

shared disk device

Disk device connected to multiple nodes. FDU/FUs and disk arrays are available as PRIMECLUSTER shared disk devices. Shared disk devices can be designated as shared disks or switched disks according to whether multiple nodes are allowed to access them simultaneously. (Note that the terms shared disk device and shared disk has different meanings.)

shared node

The nodes shared the data of GFS Shared File System. Up to four nodes are allowed.

shared objects

Object set which belongs to shared disk class. Each object may also be referred to as shared class, shared group and shared disk.

slice

A slice is an alias of a partition.

standby MDS node

MDS node other than primary or secondary MDS node.

super block

The super block contains information on the type of the file system, dates of creation and updating, label information, information on size and layout of the file system, and summary information on free areas.

super block copy

A copy of the super block is created so that file system processing can continue even if the original super block is damaged by a block fault in the disk device.

system disk

Disk on which the operating Solaris OE is installed. The entire disk containing a slice currently operating as the file system (or swap area) of one of the following:

- /
- /usr
- /var
- swap area

update log

The update log contains the history of updates made to the file system structure. The history information is used to make a high-speed recovery of a file system possible. The area that contains the update log is called the update log area. Data in this area is overwritten cyclically.

update log area

The update log area contains update log. It is normally allocated in the representative partition. In a multi-partition configuration in which the update log area is allocated to a separate partition, the update log area is not located in the representative partition, located instead in the separate update log partition.

update log buffer

Update log data that was loaded into memory.

update log partition

This partition consists of the update log area. It is used to allocate an update log area that is separate from the representative partition so that update log replay can be performed at high speeds.

update log replay

Update log replay updates meta-data in the update log area for which an update had been interrupted. This feature is used to recover the file system in the case of failures.

V-data

Directory blocks, symbolic link path names, and indirect blocks are collectively referred to as V-data. This data is stored in the V-data area that was allocated in the meta-data area.

V-data area

The V-data area contains V-data. It is allocated in the meta-data area.

Web-Based Admin View

Software for system operation and management using the Web screen. You can use it by installing ESF(Enhanced Support Facility).

Index

- 6**
64-bit file system interface 196, 713
- A**
AC..... 221, 223, 234, 236, 396, 678, 713
AC (Information messages) **409**
AC (Panic messages) **396**
AC (Warning messages) **398**
ACL..... 10, 196, 713
alternate file system 713
area management data 22, **26**, 214, **215**, 713
array disk 713
- B**
Backing up an entire file system 368
Backing up file by file 368
- C**
cache-hit ratio 204
checking a file system for consistency 339
cli_sfxfs 38, **120**, 190
copy of partition structure information 713
cpio 368, 371, 392, 394
Create File System Wizard 279, 280, 281, 293
cylinder group 713
- D**
data area 713
data block size 303, 312
data partition 713
dd..... 239, 360, 368, 371, 618, 639
defragmentation 41, 86, 134
defragmentation function 74
df_sfcfs 350, 445, **598**
df_sfcfs (Messages)..... **445**
df_sfxfs 40, 86, **121**, 190
direct I/O 10, 130, 168, 196, 609, 678, 713
directory block..... 15, 26, 83, 204, 215
- E**
EC..... 713
EC employment **375**, 376, 377, 381, 384, 385, 713
extent 83, 215, 350, 606, 644, 679, 680, 713
extent attribute information . 10, **208**, 342, 671, 688, 691, 696, 698, 700, 702, 714
extent attribute information (Free extent) 598
extent-base 215, 390, 714
extent-based management 10, 14
external quotas file..... **63**, **236**
- F**
failure recovery **225**, 226
ff_sfxfs 38, **123**, 190
fibre channel 714
file data 215, 223, 714
file data allocation attribute information..... **208**, **344**, 698, 714
file data area 10, 11, 14, 22, **27**, 28, 35, 36, 86, 128, 154, 203, 214, **216**, 223, 304, 310, 598, 635, 643, 714
file data area (Add) ... 217, 219, 275, 289, 298, 322, 325, **612**, 635
file data area (Separate) 217, 218, 220, 308, 309, 605
file data partition 306, 310, 322, 324, 325, 350, 605, 643, 644, 714
file extension attribute information **342**, 671, 688, 691
file lock..... 10, 196, 714
file system (Create)..... 272, 278, 296, **302**
file system (Delete) 276, 294, 301, 333
file system (Re-create) 300, **329**
file system (Tuning)..... 287, **321**
file system driver 221, 714
file system driver (Information messages) **414**
file system status 316
fragmentation 41, 74, 134
fs_sfxfs 185
fsck_sfcfs 201, 215, 235, 339, 449, **601**, 709
fsck_sfxfs 13, 25, 37, 38, **124**, 190
fstyp_sfcfs **604**
fstyp_sfxfs **126**, 190
- G**
GDS 374, 714
GFS Management Screen 260
GFS Management View 269, 271, 286, 294, 711, 714
GFS Management View (Error messages) **589**
GFS Management View (Exit)..... 256
GFS Management View (Information messages) **592**
GFS Management View (Start) 254
GFS Management View (Warning messages) **590**
GFS Shared File System Management View 589
GLS..... 236, 714
- H**
high speed backup (cancel) 649
high speed backup (display) 650
high speed backup (operation)..... 657
high speed backup (relationship) 654
high speed backup (restore) 660
high speed file system recovery 25, 78
host name 279
- I**
ignore update time attribute information..... **343**, 698
i-node 83, 224, 390, 598, 601, 644, 714
i-node area..... **26**, **215**, 714
instant snapshot..... 714
internal quotas file..... **63**, **236**
- L**
last accessed time non-modify attribute information .. 208
last modify time non-modify attribute information..... 208
- M**
management partition **242**, 676, 714
maximum directory size 210
maximum file size 210
maximum file system capacity 210
maximum number of contiguous blocks..... 686
maximum number of i-nodes..... 210
maximum number of partitions.... 36, **210**, 238, 303, 310, 606, 644

maximum number of sharing nodes... **210**, 238, 606, 644
 maximum size of the file data area 238, **310**, 606, 644
 MDS... 221, **223**, 226, 228, 233, 236, 390, 616, 678, 715
 MDS (Error messages).....**419**
 MDS (Information messages).....**440**
 MDS (Warning messages).....**429**
 MDS node.....715
 MDS operational information 228, 229, 233, 311, 330,
 606, **643**, 647
 MDS operational information (Change) 300, **331**, 614
 Meta area.....715
 meta-data15, 17, 22, 28, 35, 36, 38, 40, 44, 78, 154, 204,
 214, 233, 339, 388, 644, 715
 meta-data area . 17, **26**, 35, 36, 154, **215**, 303, 304, 598,
 606, 635, 644, 715
 meta-data area (Change size) 310, 606
 meta-data partition 325, 715
 meta-data partition (Add) 218, 219, 220
 minimum percentage of free area **303**, 605, 644
 mkfs_sfcs..... 449, **605**, 643, 645
 mkfs_sfcs (Messages).....**449**
 mkfs_sfxfs..... **127**, 190
 mount.....313
 mount information284
 mount point.....280
 mount_sfcs 354, 458, **609**
 mount_sfcs (Messages).....**458**
 mount_sfxfs **130**, 190
 MPHD disk.....715
 multi-partition 16, 28, 35, 44, 598, 605, 612, 643
 multi-partition configuration..... 28, 35, 44, 203, 216
 multiple LAN support 202, 236, 353

N

ncheck_sfxfs..... 38, **132**, 190
 node..... 327, 330, 340, 715
 node identifier 715
 node name.....279, 292, 715
 number of disk blocks598
 number of i-nodes..... 303

O

OPC 374, 715
 OPC employment **374**, 376, 377, 384, 385, 715

P

partition215, 234, 239, 281, 289, 305, 317, 322, 350, 367,
 368, 372, 388, 389, 604, 614, 616, 635, 643, 686, 715
 partition configuration data.. **24**, 214, **215**, 220, 340, 614,
 715
 partition configuration data copy 615
 policy 75
 primary management server 715
 primary MDS 223, 226, 280, 292
 primary server 715
 purging token679

Q

quota function 10, 83, 196, **236**, 287, 318, 394, 716
 quota function (Backing up and restoring user limit
 values).....373

quota function (Disable) 273, 284, 288, 297, **320**, 666
 quota function (Enable)273, 284, 288, 297, 310, **320**, 666
 quota function (User limit time)319
 quota function (User limit values).....318, 319
 quotas file.....63, 65, 69, 71
 quotas files (Create).....318

R

Remote console connection unit.....716
 representative partition35, 36, 37, 39, 282, 304, 340, 366,
 598, 601, 604, 605, 614, 635, 643, 647, 716
 restore..... 53, 86, **359**, 368, 618, 639, 641
 Restoring an entire file system.....372
 Restoring File by file371
 RMS.....87, 354, 716
 round robin allocation.....203, 716

S

scope716
 secondary management server.....716
 secondary MDS 223, 226, 280, 292
 secondary server716
 sfcadd 291, 310, 322, 323, 325, 326, 468, **612**, 644
 sfcadd (Messages).....**468**
 sfcadm321, 333, 340, 367, 390, 478, 607, **614**, 619, 641,
 645
 sfcadm (Messages).....**478**
 sfccp 342, 565, **688**
 sfccp (Messages).....**565**
 sfccpio..... 239, 342, 368, 371, 570, **691**
 sfccpio (Messages)**570**
 sfccdr 486, **618**, 639
 sfccdr (Messages)**486**
 sfcd u 488, **620**
 sfcd u (Messages).....**488**
 sfcdump 239, 361, **622**
 sfcdquota 318, 319, 345, 490, **627**
 sfcdquota (Messages).....**490**
 sfcfrmd 221, **223**, 416, 439, 716
 sfcfrmd (Error messages)**416**
 sfcfrmd (Information messages).....**439**
 sfcfrmstart493, **629**
 sfcfrmstart (Messages)**493**
 sfcfrmstop496, **630**
 sfcfrmstop (Messages).....**496**
 sfcs_mount.....424, 438
 sfcs_mount (Error messages)**424**
 sfcs_mount (Warning messages).....**438**
 sfcsd 221, **223**, 422, 716
 sfcsd (Error messages)**422**
 sfcsmsg 223, 419, 429, 440
 sfcsmsg (Error messages)**419**
 sfcsmsg (Information messages)**440**
 sfcsmsg (Warning messages).....**429**
 sfcgetconf497, **631**
 sfcgetconf (Messages).....**497**
 sfcgetext342, 586, **696**
 sfcgetext (Messages).....**586**
 sfcgetfattr**698**

-
- sfcquota526
 - sfchnsd221
 - sfchost500, **633**
 - sfchost (Messages).....**500**
 - sfinfo340, 351, 368, 505, 612, **635**, 709
 - sfinfo (Messages).....**505**
 - sfcmbkup.....239, 512, 619, **639**
 - sfcmbkup (Messages)**512**
 - sfcmrstr513, **639**
 - sfcmrstr (Messages).....**513**
 - sfcmntgl514, **642**
 - sfcmntgl (Messages).....**514**
 - sfcmv342, 565, **700**
 - sfcmv (Messages).....**565**
 - sfcnews215, 279, 283, 302, 304, 310, 311, 324, 326, 329, 388, 518, **643**
 - sfcnews (Messages)**518**
 - sfcnode327, 330, 340, 478, 644, **647**
 - sfcnode (Messages).....**478**
 - sfcpcnd221, 427
 - sfcpcnd (Error messages).....**427**
 - sfcprmd.....221, **223**, 415, 428, 444, 716
 - sfcprmd (Error messages)**428**
 - sfcprmd (Information messages)**444**
 - sfcprmd (Panic messages)**415**
 - sfcproxybreak508, **649**
 - sfcproxybreak (Messages).....508
 - sfcproxyinfo.....508, **650**
 - sfcproxyinfo (Messages)508
 - sfcproxyjoin.....508, **654**
 - sfcproxyjoin (Messages)508
 - sfcproxyop374, 508, **657**
 - sfcproxyop (Messages).....508
 - sfcproxyrestore508, **660**
 - sfcproxyrestore (Messages).....508
 - sfcquota345, **662**
 - sfcquota (Messages).....**526**
 - sfcquotadm345, 528, **664**
 - sfcquotadm (Messages).....**528**
 - sfcquotaoff320, 345, 532, **666**
 - sfcquotaoff (Messages).....**532**
 - sfcquotaon345, 532, **666**
 - sfcquotaon (Messages).....**532**
 - sfcprepquota537, **668**
 - sfcprepquota (Messages)**537**
 - sfcrestore239, 361, **670**
 - sfcrcsinfo.....540, **675**
 - sfcrcsinfo (Messages).....**540**
 - sfcsetext.....342, 587, **702**
 - sfcsetext (Messages)**587**
 - sfcsettime.....**704**
 - sfcsetup542, **676**
 - sfcsetup (Messages).....**542**
 - sfcsetvolume**705**
 - sfcstat388, 389, 390, 554, **678**
 - sfcstat (Messages).....**554**
 - sfctune321, 555, 598, **686**
 - sfctune (Messages).....**555**
 - sfcumntgl560, **687**
 - sfcumntgl (Messages).....**560**
 - sfxadd24, 36, 86, 94, 128, **133**, 154, 192
 - sfxadm24, 39, 44, **135**, 192
 - sfxcp**172**, 190
 - sfxcpio.....**175**, 190
 - sfxddr**137**, 192
 - sfxdefrag**139**, 192
 - sfxdump**142**, 192
 - sfxedquota64, **147**
 - sfxgetext**180**, 192
 - sfxinfo39, 43, 52, **149**, 188, 192
 - sfxmbkup.....**151**, 192
 - sfxmrstr**151**, 192
 - sfxmv**181**, 190
 - sfxnews13, 26, 34, 35, 36, 78, 86, 89, **154**, 190
 - sfxnode24, 39, 44, 89, 98, 99, 128, 154, **156**, 192
 - sfxquota64, **158**, 191
 - sfxquotadm64, 71, **159**, 192
 - sfxquotaoff64, **161**, 191
 - sfxquotaon64, **161**, 191
 - sfxrepquota64, **162**, 191
 - sfxrestore**163**, 192
 - sfxsetext.....**183**, 192
 - sfxstat**168**, 192
 - sfxtone**171**, 192
 - shared disk.....322, 324, 716
 - shared disk device716
 - shared node.....276, 717
 - shared node information (Add).....327
 - shared node information (Change)276, 291, 299, **327**
 - shared node information (Delete).....328
 - shared objects.....717
 - single partition configuration216
 - standby MDS node717
 - super block.....**23**, 24, 28, 214, 220, 601, 604, 614, 717
 - super block copy**23**, **214**, 614, 717
 - system disk717
- T**
- tar.....368, 371, 392, 394
 - token223, **224**
 - type of optimization**303**, 607, 686
- U**
- umount_sfcfs.....463, **611**
 - umount_sfcfs (Messages).....**463**
 - update log13, 38, 201, 226, 228, 339, **388**, 389, 601, 717
 - update log area13, **25**, 34, 35, 36, 78, 128, 155, 201, 214, **215**, 223, 233, 286, 304, 389, 598, 605, 635, 643, 717
 - update log area (Change size)..274, 287, 303, 310, 321, 605, 606, **615**, 644
 - update log area (Separate)216, 217, 218, 219, 220, 305, 307
 - update log buffer388, 717
 - update log partition305, 310, 350, 605, 614, 643, 717
 - update log replay602, 717

Index

V

V-data 224, 598, 717
V-data area **26, 215**, 606, 644, 686, 717

W

Web-Based Admin View **252**, 253, 717

Web-Based Admin View (Exit) 257

Web-Based Admin View (Messages) 594

Web-Based Admin View Tool Menu 253