

User Guide

hp StorageWorks File System Extender

Product Version: 3.1

First Edition (January 2005)

Part Number: AA-RW17A-TE

This guide describes how to configure and operate HP StorageWorks File System Extender software. It details its hierarchical storage management principles, offers practical information on optimizing the operation of the software, and explains FSE commands.



© Copyright 2005 Hewlett-Packard Development Company, L.P.

Hewlett-Packard Company makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard. The information contained in this document is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Linux® is a U.S. registered trademark of Linus Torvalds.

UNIX® is a registered trademark of The Open Group.

Windows® is a U.S. registered trademark of Microsoft Corporation.

Hewlett-Packard Company shall not be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Hewlett-Packard Company products are set forth in the express limited warranty statements for such products. Nothing herein should be construed as constituting an additional warranty.

File System Extender User Guide
First Edition (January 2005)



- About this Guide.11**
- Overview. 11
 - Intended Audience 11
 - Prerequisites 11
 - Related Documents 12
- Getting Help 12
 - HP Technical Support 12
 - HP Storage Web Site 13
 - HP Authorized Resellers 13
- 1 Introducing HP StorageWorks File System Extender15**
- What is FSE? 15
 - FSE Basics 16
 - Infinite file system 16
 - Library—tape compatibility 16
 - Automatic error detection 17
 - NFS and CIFS support. 17
- FSE Architecture 18
 - FSE Components 18
 - HSM file system (HSM FS) 18
 - FSE partition 19
 - FSE media pool 20
 - FSE medium 20
 - FSE library. 21
 - FSE drive 21
 - FSE Daemons/Services and FSE Agents 21
 - FSE User Interface 23
 - FSE Installation Deployment 24
 - Consolidated implementation 24
 - Distributed implementation 25

FSE client processes	26
Client-server communication in a distributed FSE implementation	27
Configuring the communications settings in a distributed FSE implementation	28
FSE installation with multiple libraries	29
FSE Operations	30
Migration	30
Release	31
Recall	31
2 Configuring FSE	33
Before You Start	33
Configuration Basics	35
Configuration Files	35
FSE configuration file elements	36
Templates	37
Configuration Database	37
Resource Management Database	38
About FSE Commands	38
FSE Commands List	40
Using FSE Commands for Configuring Resources and Policies	41
Configuration Procedure	43
Step 1: Configuring FSE Libraries and FSE Drives	44
About SCSI identification	44
Library and drive configuration procedure	44
Library and tape drive configuration examples	47
Step 2: Configuring FSE Media Pools	52
Step 3: Preparing HSM File Systems	52
Preparing HSM file systems on Linux	52
Preparing HSM file systems on Windows	54
Step 4: Configuring FSE Partitions	55
Configuring an FSE partition for a Linux FSE client	55
Configuring an FSE partition for a Windows FSE client	56
Configuring a WORM FSE partition	58
Step 5: Preparing FSE Media	59
Step 6: Mounting HSM File Systems	60
Mounting on Linux	60
Mounting on Windows	61
Reconfiguring FSE Interprocess Communication	62

Reconfiguring Communication on a Consolidated FSE Implementation or on an FSE Server	64
No external FSE clients or ordinary LAN connection	64
Private network connection	64
Reconfiguring Communication on External Linux FSE Clients	65
Ordinary LAN connection	65
Private network connection	66
Reconfiguring Communication on External Windows FSE Clients	67
Ordinary LAN connection	67
Private network connection	67
3 Managing Media	69
FSE Media Pools	69
Media Pool Types and Their Characteristics	70
Configuring a Media Pool	71
Determining the Configured Media Pools	74
Checking the Status of a Particular Media Pool.	76
FSE Media	76
FSE Medium Volumes.	78
FSE Media Types.	78
Introducing Disk Media	79
Local and remote file systems as disk media	80
Disk media specifics	81
Adding Media to an FSE Media Pool	83
Formatting Media.	83
Initializing Media.	84
Duplicating Media	85
How it works	86
Procedure for duplicating media	86
Reorganizing Media.	88
The media reorganization process	89
Media reorganization parameters	92
Exceptions to the reorganizational rule	92
Scanning media for obsolete data	93
Checking the resulting statistics of a reorganizational scan job	94
Running the media reorganization job.	95
Reusing the recycled medium volumes.	97
Recreating Redundant Copies of Migrated Data	97
The redundant copy recreation process	98

Determining the unusable media	99
Starting redundant copy recreation	100
Removing the unusable medium from the media pool	101
Resolving situations with no alternative migrated copies	101
Checking the Status of Media	102
Checking the status of a particular medium	104
Changing the Condition Status of Media	105
Prevent writing to an FSE medium	105
Prevent reading from and writing to an FSE medium	105
Putting an FSE medium back into use	105
Closing Medium Volumes	106
Recreating Fast Recovery Information from Media	106
Removing FSE Media	107
4 Migration, Release, Recall, and Deletion	109
FSE Operation Basics	109
FSE Hierarchical Storage Manager Lists	111
About Policies	113
Migration	114
Migration Policy Parameters	115
Minimum file age for migration (MinFileAge)	115
Minimum wait time for migration (MinWaitTime)	115
Maximum wait time for migration (MaxWaitTime)	115
Minimum number of files for migration (MinNumMigFiles)	115
Maximum number of files for migration (MaxNumMigFiles)	116
Minimum size for migration (MinMigSize)	116
Maximum size for migration (MaxMigSize)	116
Example situations that trigger migration	116
Default migration policy	117
Release	117
Watermarks	118
Critical watermark	118
High watermark	120
Low watermark	120
Exclusion from Release	121
Filename pattern conventions	122
Configuring exclusion from release	124
Releasing excluded files	125
Release Policy Parameters	125

Critical watermark (CriticalWaterMark)	125
High watermark (HighWaterMark)	125
Low watermark (LowWaterMark)	125
Release interval (ReleaseInterval)	126
Migration retention time (MigRetentionTime)	126
Recall retention time (RecallRetentionTime)	126
Minimum file size for release (MinFileSize)	126
Files that are never released (ExcludeFromRelease)	127
Example situations that trigger release	127
Default release policy	127
Recall	128
Recall Timeout Parameter	129
Recall timeout (RecallTimeout)	129
Default recall policy	129
Deletion	129
Automatic Deletion Process	130
Deletion Policy Parameters	130
File expiration time (FileExpiration)	131
Directory paths (Path)	131
Examples of a Configured Deletion Policy	131
Starting the Deletion Job	133
Recalling Deleted Files	133
Resource Allocation	134
Resource Allocation Concepts	134
Central FSE resource manager	135
Media selection by media location	135
Diverse allocation parameters	136
Job priorities	136
System Allocation and Job Priority Policy	136
System Allocation and Job Priority Policy Parameters	136
Maximum number of drives for migration (SystemMaxNumDrivesMigration)	137
Maximum number of drives for recall (SystemMaxNumDrivesRecall)	137
Maximum number of drives for administration jobs (SystemMaxNumDrivesAdmin)	
137	
Maximum number of drives for recovery (SystemMaxDrivesRecovery)	137
Resource allocation threshold (ResourceAllocationThreshold)	138
Time step (TimeStep)	138
Phase factor (PhaseFactor)	138

Backup job priority (BackupJobPriority)	139
System administration job priority (SystemAdminPriority)	139
Default system allocation and job priority policy	139
Partition Allocation and Job Priority Policy	139
Partition Allocation and Job Priority Policy Parameters	140
Maximum number of drives (MaxNumDrives).	140
Maximum number of drives for migration (MaxNumDrivesMigration)	140
Maximum number of drives for recall (MaxNumDrivesRecall).	140
Maximum number of drives for recovery (MaxNumDrivesRecovery).	140
Partition priority (PartitionPriority).	140
Migration priority (MigrationPriority)	141
Recall priority (RecallPriority)	141
Recovery priority (RecoveryPriority)	141
Maintenance priority (MaintenancePriority).	141
Default partition allocation and job priority policy	142
Priority Calculation.	142
Multiple Copying	144
Parallel Copying.	145
Sequential Copying	145
HSM File System Access Modes	146
Limited Access Mode (LAM) and Full Access Mode (FAM).	146
5 Monitoring and Maintaining FSE	149
About Monitoring and Maintaining in FSE	149
Monitoring the FSE Installation	149
Managing FSE Partitions	150
Monitoring FSE Partition Status	150
Disabling an FSE Partition.	152
Enabling an FSE Partition	152
Removing an FSE Partition	153
Retrieving Detailed Information about Configured FSE Partitions.	154
Switching the Access Mode for an HSM File System.	154
Switching to Limited Access Mode.	154
Switching to Full Access Mode.	154
Monitoring FSE Jobs.	155
Checking the Consistency of the File System Catalog.	157
Consistency Check of FSC vs. HSM File System.	157
Consistency Check of FSC vs. FSE Media	160
Recalling Older Generations of a File	163

Viewing Logs	166
The FSE Event Log	166
Logging FSE media operations	167
The FSE Error Log	172
Retrieving Detailed Information	178
Examples of Command Outputs Generated using the --list --detail Options	179
6 Backup, Restore, and Recovery	191
Defining Your Backup Strategy	192
FSE System Components	192
FSE Backup Types	194
Methods of Retrieving Data	195
Backup	195
What is FSE Backup?	196
How Does it Work?	196
Backup Media Management	197
The fsebackup Command	200
Restore	207
What is FSE Restore?	207
Restore Procedure	208
Restore requirements	208
Executing FSE restore	209
Post-restore Steps	212
Recovery	212
File System Catalog Recovery	213
HSM File System Recovery	213
How it works	214
7 Troubleshooting	217
HSM File System Filter Problems	218
Wrong Kernel Version on the FSE Client	218
Checking HSM File System Kernel Messages	218
HSM File System Mounting Problems	219
HSM File System cannot be Mounted	219
Partition ‘ready’ or ‘mounted’ but cannot be Accessed	220
FSE Media Management Problems	222
Media Initialization Fails with Large Block Size	222
Media Format Fails with Message “illegal request - No MIC detected”	222
FSE Migration Problems	223

Migrations all Waiting for Disk Buffer	223
fse-hsm Goes Down if all I-nodes on the HSM FS are Used.	223
FSE Recall Problems	223
Recall Job does not Continue if Medium is Marked Unusable During the Job.	223
Device Problem is Reported During a Failed Recall	224
FSE Backup and Restore Problems	224
Backup is Aborted, but its Job Hangs	224
Backup does not Start because of Previous Unfinished Backup Jobs	225
FSE Start-up and Shut-down Problems	225
The fse --stop Command does not Finish and Leaves FSE Processes Running	225
Communication Problems	226
IP Address of an FSE Installation has Changed.	226
CORBA Cannot be Initialized	226
CORBA Naming Service Cannot be Accessed	227
Connection to Management Interface Fails	227
Wrong Network Configuration	227
fsefile Fails with Error Message “6004 Caught Corba exception”	228
Firebird Server Problems	228
SCSI Problems	229
Various SCSI Errors.	229
Blocked Access to a SCSI Device	230
Tape Library Problems	230
The Tape Library Reports the Same Barcode in Different Locations	230
FSE Troubleshooting Tools	231
Debugging FSE Processes	232
Debugging Options	232
Debugging Backup and Restore Jobs.	237
Potential Problems with Debugging.	237
Contacting Support.	238
Gathering Information About Your Problem.	238
Checklist for required information	238
Contact	239
FSE Error Codes.	240
A Directory Layout	249
B Command Reference	251
Introduction to the FSE system (fseintro).	252
fse	257

fsebackup	265
fsebak	270
fsecheck	275
fsedrive	278
fsefile	283
fsejob.	291
fselibrary.	296
fsemedium.	302
fsepartition	313
fsepool.	322
fserrecover	328
fserrestore.	332
fsesystem.	334
C FSE Configuration Templates	341
Template for FSE Libraries	342
Template for FSE Drives	343
Template for an FSE System Configuration.	344
Template for FSE Partitions—General Part	345
Template for FSE Partitions—Migration Part	346
Template for FSE Partitions—Release Part	347
Template for FSE Partitions—Recall and Allocation Part.	348
Template for FSE Partitions—Deletion Part	349
Template for an AIT Media Pool	349
Template for an SAIT Media Pool	350
Template for an LTO Media Pool	351
Template for a Disk Media Pool.	352
Glossary.	355
Index	377

Overview

The FSE User Guide provides information to help you:

- Understand the basic concepts of FSE.
- Learn typical tasks needed for operating FSE.

Intended Audience

All FSE users and administrators. Experience with the following is necessary for advanced configuration tasks:

- SUSE LINUX Enterprise Server 8 (SLES) or Red Hat Enterprise Linux 3 (RHEL) operating systems (abbreviated to Linux throughout this guide).
- Windows 2003, Windows 2003 Storage Server or Windows 2000 operating systems (abbreviated to Windows throughout this guide).

Prerequisites

Before you configure FSE, make sure you consider the items below.

- Installation environment:
Read the *FSE Installation Guide* and ensure that your FSE environment is correctly installed and functioning.
- Software requirements:
Check additional software requirements needed for setting up the FSE installation.
- Hardware requirements:
For information on hardware requirements see the *FSE Release Notes*.

Related Documents

In addition to this guide, HP provides corresponding information:

- *FSE Release Notes* (`RelNo.pdf` on the installation CD)
This document provides information on product requirements, new features, limitations, and recommendations.
- *FSE Installation Guide* (`InstG.pdf` on the installation CD)
This guide describes how to install the FSE software and is intended for system administrators. It details software and environment requirements and includes mandatory pre-installation and preparation procedures.
- *FSE Management Console Reference Guide* (`UIG.pdf` on the installation CD)
This guide describes how to use the add-on FSE Management Console application, an easy-to-use tool that enables you to perform many FSE monitoring and administration tasks.

Additionally, FSE commands with detailed description of their functionality and syntax are explained in the FSE command-line reference, which is installed as a part of the FSE package and available on Linux as UNIX man pages, and on Windows as Windows help. This information is also provided in this manual; see the “[Command Reference](#)” on page 251.

Getting Help

If you still have a question after reading this guide, contact an HP authorized service provider or access our web site: <http://www.hp.com>.

HP Technical Support

Technical support information can be found at the HP Electronic Support Center at the following web site: <http://www.hp.com/support/>.

Be sure to have the following information available before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Description of the steps being performed

HP Storage Web Site

The HP web site has the latest information on this product, as well as the latest drivers. Access storage at: <http://www.hp.com/country/us/eng/prodserv/storage.html>. From this web site, select the appropriate product or solution.

HP Authorized Resellers

For the name of your nearest HP authorized reseller:

- In the United States, call 1-800-345-1518
- In Canada, call 1-800-263-5868
- Elsewhere, see the HP web site for locations and telephone numbers:
<http://www.hp.com>.

Introducing HP StorageWorks File System Extender



This chapter includes the following topics:

- [What is FSE?](#), page 15
- [FSE Architecture](#), page 18
- [FSE Operations](#), page 30

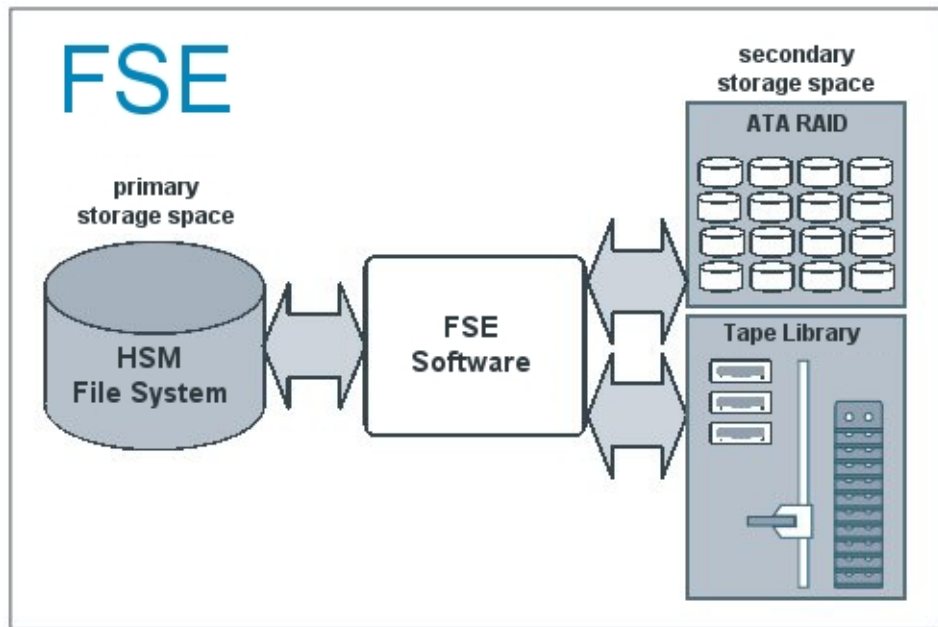
What is FSE?

HP StorageWorks File System Extender (FSE) is a mass storage oriented software product, based on client-server technology. For details on supported and required hardware and software items, see the *FSE Release Notes*.

FSE provides large storage space by combining disk storage and a tape library with high capacity tape media and by implementing Hierarchical Storage Management. It efficiently manages and stores extremely large quantities of data. Its main benefits are easy data management with fast data access, economical data storage, and data protection. Data management functions can be customized and automated for each user of an FSE installation.

The efficiency of the FSE installation is achieved with a large storage capacity that consists of primary and secondary storage space. These two types of storage space present a successful combination of the speed of disk storage and the capacity and flexibility of tape media.

FSE software manages both primary and secondary storage space, and it is implemented with a series of intercommunicating components that are responsible for data management. For detailed information on these components, see “[FSE Components](#)” on page 18.



Example of FSE storage spaces

FSE Basics

Infinite file system

The most significant characteristic of FSE is an infinite file system and a very transparent access to the data through the file system. You have access to essentially unlimited, virtual storage capacity. FSE constantly controls the files in use. The way FSE controls these files is defined by you with migration and release policies. Additionally, FSE ensures ultimate data availability and protection with data replication. It also enables multiple copying of migrated files, therefore satisfying the need for additional backup copies and protecting from failure of any single medium.

Library—tape compatibility

FSE software is a part of HP StorageWorks File System Extender, and can be used with supported SCSI tape libraries with supported drive types. For details see the *FSE Release Notes*.

FSE software supports LTO drive technologies for LTO Ultrium 1 and LTO Ultrium 2 media. Sony AIT-2 and AIT-3 WORM media is supported where a WORM type secondary media is required.

Automatic error detection

FSE automatically detects drive and media hardware errors. When FSE detects such an error, the following happens:

- The drive is marked as offline.
- If the error occurs during the migration job, the medium is marked as unreliable, which disables further writing to it.
- If the error occurs during the recall job, the medium is marked as unusable, which disables reading from and writing to it. Neither the marked drive nor medium are used for the operation until the administrator clarifies the situation and appropriately marks drive and medium. For more information, see “[Changing the Condition Status of Media](#)” on page 105.

NFS and CIFS support

Windows and UNIX clients access the HSM file system over standard protocols. Network File System (NFS) is the standard protocol for UNIX and Linux. Common Internet File System (CIFS) is the standard protocol for Windows clients.

FSE supports mounting and accessing an FSE partition only via Network File System (NFS) on Linux.

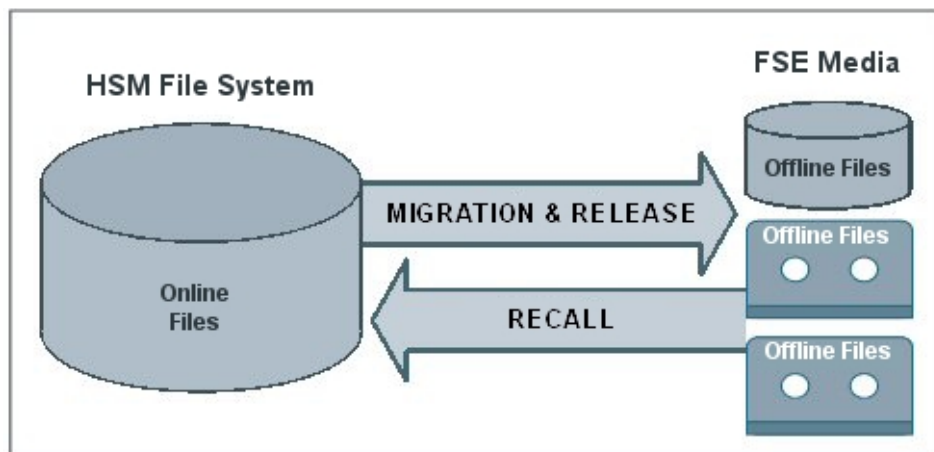
FSE supports mounting and accessing FSE partitions on a Windows machine through regular CIFS protocol (Windows shares) without any further requirements.

On Linux clients, CIFS protocol is available as a special add-on custom-built Samba package. Files that are written to or read from an HSM file system over a Samba connection are handled like all other files created or accessed locally on the HSM file system. The add-on package enables Samba to notify machines (especially Windows Explorer), running on Windows, when a file is offline, thus preventing unnecessary recalls of offline files. For further information on Samba integration, see the *FSE Release Notes*.

FSE Architecture

Efficiency is achieved by FSE exerting continuous control over the files in use. You define how FSE controls these files with different policies.

Basically, FSE copies new or modified file information from a file on an HSM file system to the FSE medium, releases rarely used files and makes them offline, and recalls offline files when necessary. These are the principal FSE jobs, called migration, release, and recall.



Main operations in FSE

FSE Components

HSM file system (HSM FS)

An HSM file system (HSM FS) is a hard disk area reserved for file storage, and managed by the FSE installation. The HSM file system is used to store the file metadata (name, attributes, permissions) and online files. On Linux, it is based on a Linux Ext3 file system; the HSM attributes are stored in Ext3 extended attributes. On Windows, file systems are based on NTFS.

In an FSE installation, an HSM FS represents the primary storage space, and the FSE media represent the secondary storage space.

An HSM file system can be configured either as a regular file system or as a WORM file system:

- Regular file system
Files and directory operations are supported as known in a native Ext3 file system on Linux and NTFS on Windows.
- WORM (Write-Once Read-Many) file system
File and directory operations are restricted to read-only after a configured period of time. The limitation does not include adding files to directories, which can be done as on a regular file system.
For more details on configuring an HSM file system as a WORM file system, see “[Step 4: Configuring FSE Partitions](#)” on page 55.

Hierarchical Storage Manager (HSM) is a component of the HSM file system. As a part of an FSE client, it handles HSM file system events and manages migration- and release candidate lists. Each mounted HSM file system has one FSE HSM.

In a distributed FSE implementation, the HSM file system, HSM, and the File System Event Manager are located on a separate FSE client host.

FSE partition

FSE uses FSE partitions to split the storage into multiple entities, meaning that one FSE installation can have more than one FSE partition. This way FSE provides the possibility for complex organization of data management.

Each FSE partition is related to a single HSM file system and has its own policies and its own File System Catalog (FSC). This enables you to configure individual policies (migration, release and deletion, and resource allocation) for a particular FSE partition, according to the characteristics of files and their intended use.

Each configured FSE partition is controlled by one Partition Manager (PM), which creates and manages FSE jobs for migration, recall, and deletion operations. The Partition Manager is located on the FSE server. It is included in the FSE server installation package.

File System Catalog

The File System Catalog (FSC) is a database related to a single FSE partition. It consists of a Data Location Catalog (DLC) and a Name Space Catalog (NSC).

- The Data Location Catalog contains information about location of files on the FSE media. It stores a full history of file locations.

- The Name Space Catalog (NSC) contains metadata of files on an HSM file system. It only stores the latest generation metadata, which includes directory structure, standard attributes and ownership on Linux, and ACL (Access Control List) and Alternate Data Streams on Windows. This data enables recovery of an HSM file system. For details, see [“Recovery”](#) on page 212.

The File System Catalog is located on:

- Linux: `/var/opt/fse/part/<PartitionName>/fsc`
- Windows: `%InstallPath%\var\part\<PartitionName>\fsc`
(where the default `%InstallPath%` is `Program Files\Hewlett-Packard\FSE`, but you can change this path by selecting Custom Setup during installation)

FSE media pool

An FSE partition must have at least one media pool assigned to it. An FSE media pool is configured to group media of the same type, which have the same characteristics.

Media pools are configured regarding their purpose and correspond to the specifics of FSE partition data and FSE media types, for example a WORM or regular disk media pool. Therefore one media pool cannot be assigned to multiple partitions in order to keep the files from different FSE partitions separated. An FSE partition can have more media pools assigned to it, but they have to be of type regular or WORM.

By assigning multiple media pools to one FSE partition, multiple copies of FSE data are made (as described in [“Multiple Copying”](#) on page 144). For more information on media pool types and their configuration, see [“FSE Media Pools”](#) on page 69.

FSE medium

FSE supports different types of tape media and also disk media for data storage. Each FSE medium, either a medium cartridge or disk medium has an entry and properties written in the Resource Management Database (RMDB). Each FSE medium is assigned to one FSE media pool. For more information on media types, see [“FSE Media Types”](#) on page 78.

An FSE medium is usually formatted and further divided into partitions that hold medium volumes. The FSE Resource Manager identifies the medium with a barcode. For media specifics regarding their type, see [“FSE Media Types”](#) on page 78 and [“Introducing Disk Media”](#) on page 79.

FSE library

The FSE library is any supported tape library attached to a host with FSE software, and configured as an FSE installation. An FSE installation can consist of one or more FSE libraries. Each FSE library is a repository of library slots, where medium cartridges are physically located, with a built-in changer device to move the media around. It has one or more connected drives.

Each configured FSE library has one FSE Library Agent running. The FSE Library Agent handles actions in the FSE library, such as loading and unloading media, and library inventory rescan.

FSE drive

An FSE drive is a tape drive inside the FSE library, configured in FSE. It is used to read to and write from FSE media.

Supported drive types are:

- LTO Ultrium 1, LTO Ultrium 2
- AIT-2 WORM, AIT-3 WORM

Each drive type requires a special drive firmware revision that enables the detection of WORM media.

FSE Daemons/Services and FSE Agents

There are several intercommunicating processes running on the FSE installation. These processes are either FSE daemons/services or FSE agents.

Generally, FSE daemons/services are the processes that are running most of the time FSE is operating. An FSE daemon/service is usually a server process with several sophisticated services. FSE agents are software agents, and are designed to perform defined tasks required by each specific job.

Binaries for FSE daemons/services and FSE agents are located on:

- Linux: `/opt/fse/bin`
- Windows: `%InstallPath%\bin`
(default: `%InstallPath%` is `Program Files\Hewlett-Packard\FSE`, but you can change this by selecting Custom Setup during installation)



Caution: You should not execute any of the following commands manually. Wrong use can cause data corruption.

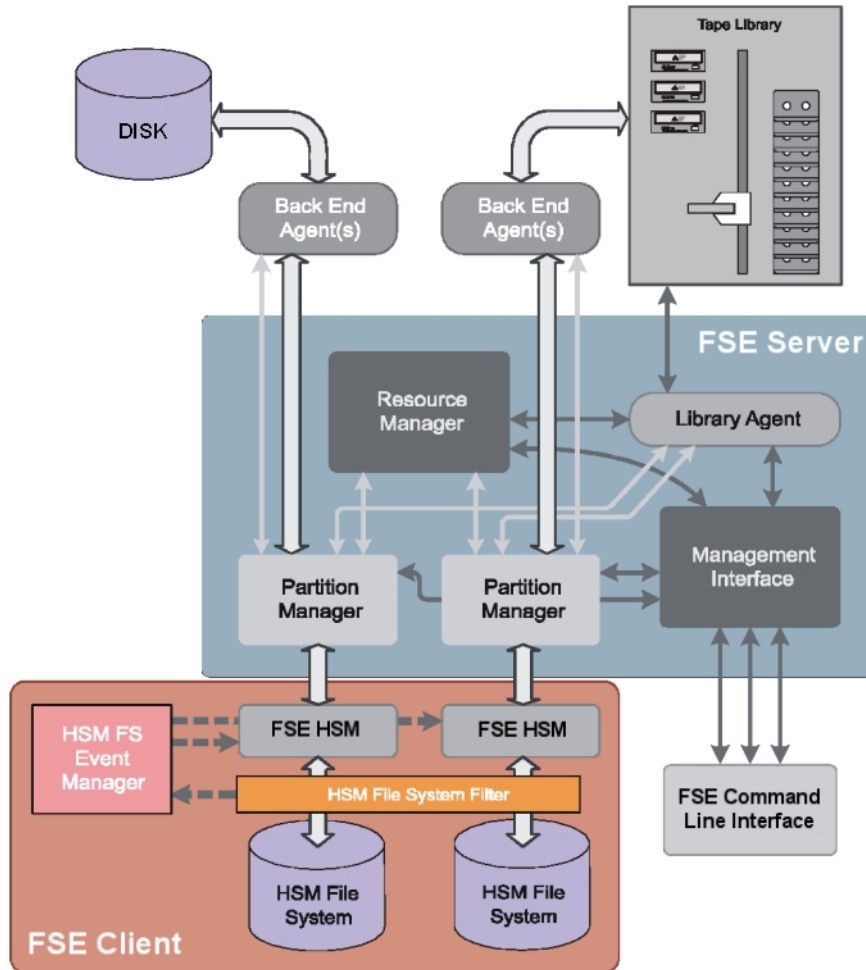
On Linux, you can execute only the commands in the `/opt/fse/sbin` directory.

On Windows, `fse.exe` can also be used for starting and stopping FSE services.

FSE daemon/service and FSE agent list

FSE daemon/service or FSE agent	Binary	Running on...	Description
File System Event Manager	<code>fse-fsevtmgr</code>	FSE client	Intercepts mount events for the HSM file system and starts Hierarchical Storage Manager (HSM)
Hierarchical Storage Manager	<code>fse-hsm</code>	FSE client	Receives file system events, manages FSE HSM lists, starts migration, release, recall, and deletion jobs
FSE Service	<code>fse-svc</code>	FSE client, FSE server	Launches other FSE daemons/services and FSE agents
FSE Windows Service	<code>fse.exe</code>	Windows FSE client, Windows FSE server	Launches other FSE services at Windows system start-up
Resource Manager	<code>fse-rm</code>	FSE server	Manages Resource Management Database, allocates FSE resources (FSE drives, FSE media)
Management Interface	<code>fse-mif</code>	FSE server	Handles user requests issued with FSE commands, manages FSE administration jobs
Partition Manager	<code>fse-pm</code>	FSE server	Manages FSE jobs for migration, recall, deletion, recovery, reorganization, and maintenance
Library Agent	<code>fse-la-s</code>	FSE server	Handles actions on FSE libraries, such as loading and unloading media, library inventory rescan, and so on
Back End Agent	<code>fse-bea</code>	FSE server	Handles data transfer between FSE drives and FSE media, executes FSE media formatting and initialization

The following figure demonstrates the interaction of the main FSE components.



FSE User Interface

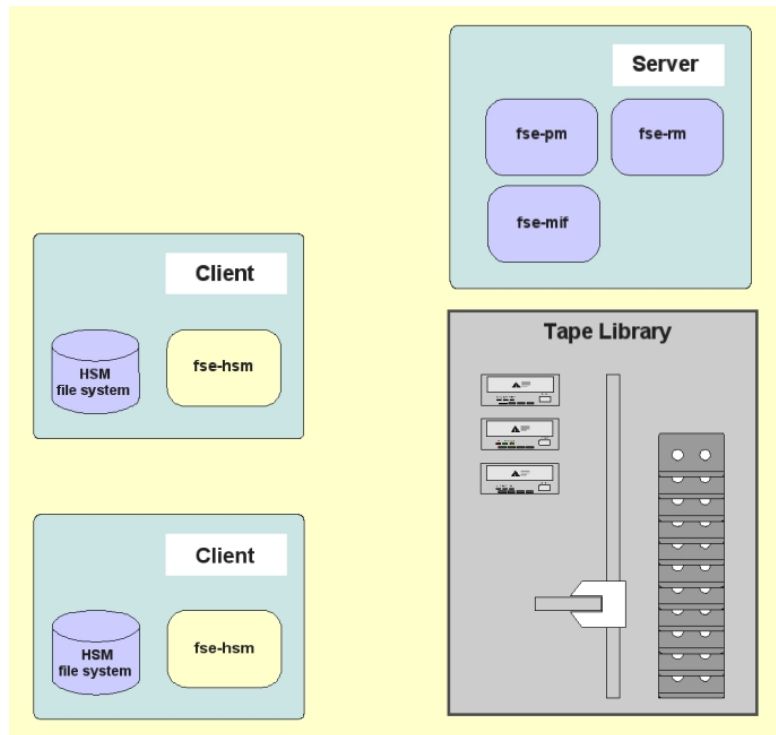
The FSE command-line interface provides commands to configure, monitor, and manage the FSE installation. The commands are passed to FSE through the FSE Management Interface, responsible for accepting, handling, and executing requests for user actions. FSE command-line reference is available on Linux as UNIX man pages and on Windows as Windows/HTML help. The commands are also explained in [“About FSE Commands”](#) on page 38.

FSE Installation Deployment

Due to its modular structure FSE allows several configurations. It can be set up as a consolidated implementation or as a distributed implementation of servers and clients, which can be either Linux or Windows. Additionally, it provides support for multiple tape libraries. The number of libraries that can be connected to one FSE server depends on the number of slots for Fibre Channel and SCSI cards on the machine running the FSE server.

Consolidated implementation

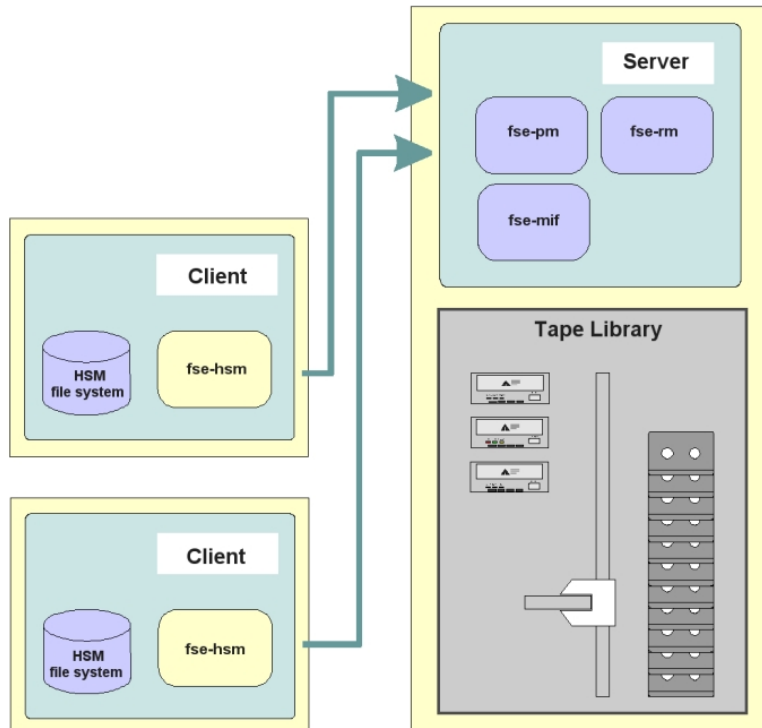
The simplest application is the “all in a box” configuration, where all FSE components reside on the same machine..



This consolidated FSE implementation integrates the functionality of both FSE server and FSE client in a single machine. It has a direct connection to the secondary media and hosts an arbitrary number of HSM file systems. User data from local HSM file systems are recorded on the media in the attached tape library or disk media.

Distributed implementation

In this, FSE clients connect to one FSE server and physically separated. External clients connect to the server through either a LAN or a private network, and host HSM file systems, while the FSE server does not have any HSM FS partitions.



The clients manage the HSM file systems and communicate with major services running on the server. User data from HSM file systems on clients is transferred to the FSE server and recorded on the corresponding tape media in the attached tape library, or on the secondary disk. This makes it possible to manage file systems on multiple file server machines without the need for a library and drives for each machine. It is also possible to configure a mixed implementation, which combines the consolidated implementation with external clients.



Caution: An FSE installation does not support any proxy server or firewall between FSE clients and the FSE server. FSE will only work if there is a direct network connection with a full-duplex transmission between FSE clients server.

For the latest information on supported TCP/IP connections, see the *FSE Release Notes*.

FSE client processes

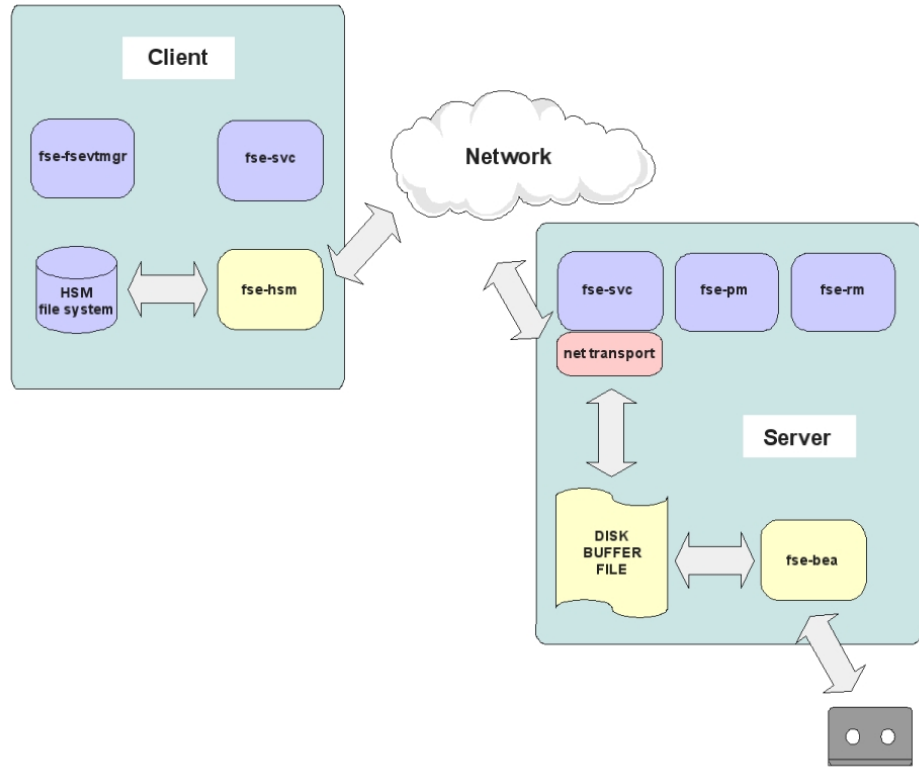
FSE supports external Linux and Windows clients. Both types of clients have the same features and requirements, but are different as regards installation. For installation specifics, see the *FSE Installation Guide*.

Once the FSE client packages are installed on the client machine, the following FSE daemons/services and programs exist on the client:

- HSM File System (`hsmfs`)
- FSE HSM Database (`hsmdb`)
- FSE Hierarchical Storage Manager (`fse-hsm`)
- FSE File System Event Manager (`fse-fsevtmgr`)
- FSE Service (`fse-svc`)
- FSE Windows Service (`fse.exe`) (on Windows systems only)

Client-server communication in a distributed FSE implementation

An external FSE client communicates with an FSE server through the network using CORBA services. An external client can connect to only one FSE server.



The network must provide a full-duplex connection between the FSE server and each of the external FSE clients. You can verify the network connection using the FSE tool `fse_net`. For more information about `fse_net`, see “[FSE Troubleshooting Tools](#)” on page 231.

You can check whether a network connection is full-duplex by pinging the server from the client and vice versa. If there is no response to one of the ping commands, the connection is half duplex only. In this case, FSE cannot perform its basic operations.

Half duplex connections are provided, for example, if NAT (Network Address Translation) is used between FSE clients and the FSE server.

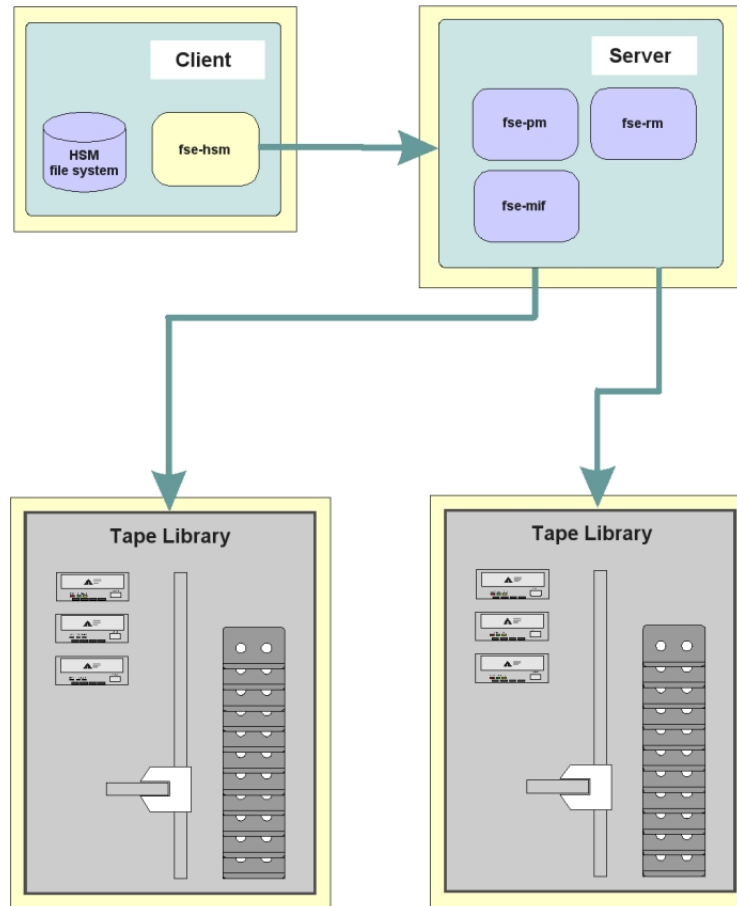
Configuring the communications settings in a distributed FSE implementation

In a distributed FSE implementation, an FSE client needs to know to which FSE server it is related. To provide this information, you need to modify the FSE interprocess (client-server) communications settings according to your particular FSE set-up. These settings are affected by your choice of ordinary LAN connections or private network connections for FSE client-server communication. You can retrieve the parameters that you will use for network configuration using the FSE *fse_net* tool.

If you are about to configure the interprocess communication in a newly installed FSE implementation, the configuration must be performed before the first start-up of FSE. Similarly, reconfiguring the communications in an already employed FSE implementation is allowed only while all FSE-related processes (including the omniNames service) on all FSE hosts are stopped. For instructions on how to perform the configuration, see [“Reconfiguring FSE Interprocess Communication”](#) on page 62.

FSE installation with multiple libraries

Attaching multiple libraries provides an option for a scalable FSE installation. If storage space is running out, another library can be added to the installation. Media inside the new library can then be added to existing media pools.



If multiple copies on different libraries are needed, the solution is using multiple libraries. Media can then be assigned to separate pools dedicated in a one-to-one relationship to each library.

FSE Operations

FSE main operations are migration, release, and recall. They are configured with user-defined policies. For more information on configuring policies, see [“Migration, Release, Recall, and Deletion”](#) on page 109.

All these operations can be triggered manually using `fsefile` command. For details, see [“fsefile”](#) on page 283.

Sometimes it is necessary to change a job’s priority or to gently terminate an FSE job. This is possible using the `fsejob` command. A job’s priority might be changed, for example, because someone on the board urgently needs certain documents to be recalled, but other jobs currently have a higher priority; or the library ran out of media and therefore additional media need to be added and initialized. In this case, migration jobs may have been queued for a long time and thus received a high priority. The administration job used to initialize the new media now needs to be set above the priority of those migration jobs, in order to get the media available and migrations running again. All jobs currently in the queue can also be listed using `fsejob`. For details, see [“Monitoring FSE Jobs”](#) on page 155, and [“fsejob”](#) on page 291.

Migration

Migration is a process of copying files from the HSM file system to FSE media. If a file is not modified for a configured period of time, it passes the so-called file age check and becomes a candidate for migration (migration candidate).

Migration is done in two steps:

1. Files from the migration candidate list are copied from the HSM file system to a temporary location on the FSE server, called a disk buffer. The disk buffer is located on:
 - Linux: `/var/opt/fse/diskbuf/`
 - Windows: `%InstallPath%\var\diskbuf`
2. Data from this temporary location is copied to the FSE media. To have redundant copies of the file data made during the migration job, the respective FSE partition must have multiple media pools assigned to it. For more information, see [“Multiple Copying”](#) on page 144.

Once the migration job is completed, the temporary location is cleared. The file entry in the FSE HSM is moved from the migration candidate list to the release candidate list. For further information on migration job, see [“Migration, Release, Recall, and Deletion”](#) on page 109.

Release

Release is a process following the migration or recall, during which the file data is removed from the HSM file system, leaving only the file's header. FSE starts the release according to the watermark parameters. For further information, see [“Watermarks”](#) on page 118.

Once the file is copied to the FSE media, it becomes a release candidate. FSE checks the release candidates for changes. If a file on the release candidate list was not modified, and a high watermark is reached on the HSM file system, the content (file data) of the file is removed from the HSM file system to free disk space. The same thing happens if a critical watermark is reached, but in this case all migrated files are released, regardless of how much time has passed since their last migration. For detailed information on release parameters, see [“Release”](#) on page 117.

Once the release operation is completed, the file is offline and its file entry is removed from the release candidate list.

Recall

Recall is a process of copying file data from the FSE medium back to the HSM file system. It is the reverse process to migration.

Recall is done in two steps:

1. Data is copied from the FSE media to a temporary location (disk buffer).
2. Data from this temporary location is copied back to the HSM file system, thus recreating file contents.

Once the recall job is completed, the temporary location is cleared, and the offline file is again online. Its file entry in the HSM file system is added to the current release candidate list. For further information on recall job, see [“Migration, Release, Recall, and Deletion”](#) on page 109.

FSE stores older versions of files on its media, referred to as older generations, which can be recalled manually. For details, see [“Recalling Older Generations of a File”](#) on page 163.

Configuring FSE

2

This chapter includes the following topics:

- [Before You Start](#), page 33
- [Configuration Basics](#), page 35
- [About FSE Commands](#), page 38
- [Configuration Procedure](#), page 43
- [Reconfiguring FSE Interprocess Communication](#), page 62

Before You Start

FSE provides effective management of your data with its extended virtual disk storage, which is controlled by user-defined policies. Although operating FSE is transparent, you need to think ahead and analyze your needs in order to configure your environment most efficiently.

The following list introduces the most common choices that you need to make in advance.

- Analyze your environment: determine which file systems need to be under FSE control.
- Analyze the types of data stored in your environment and sort the data logically according to the following characteristics:
 - Which files need to be migrated (archived)?
 - Will such files be used again for write operations or will they be treated as read-only?
 - Which of these files need additional (redundant) migrated copies?
 - What will be the approximate or average size of files stored on an FSE partition?
 - What will be the estimated maximum number of files that will be stored on an FSE partition?

- Will certain files become redundant after a period of time and can therefore eventually be deleted to free space on the file system?
- Will certain files have to be deleted after some period of time due to legal requirements?

Note that these characteristics and how you apply them in your policies can significantly influence the performance of the FSE installation. For example, data that needs frequent archiving will put a significant load on your system because of the constant migrations. On the other hand, data that you will only use for read-only purposes demands only one migration and many recalls and releases, resulting in a reduced amount of media, and reduced impact on FSE system performance.

- Regarding file expiration, consider the following points:
 - Will certain files need to be deleted from FSE partitions after a period of time, due to legal requirements?
 - How many migrated file generations will be treated as important, or how long will they need to be stored on FSE media?
- Determine the number of available drives. In a setup with tape as the only secondary media, the number of available drives will limit the number of migrations and recalls that can be performed in parallel on the FSE installation. It will also limit the number of copies that can be made in parallel.
- Decide on the number and individual characteristics of your FSE partitions; consider the previously determined data characteristics for each partition. Note that each FSE partition has its own media pools assigned and is located on a separate HSM file system.
- Create a special media pools for the backup up of the FSE installation. The backup media pool requires enough media to hold snapshots of the HSM file systems and FSE databases; you will have to decide how many backup copies you want to keep for data protection.
- Consider supported media types (tape, disk) according to your data characteristics. Note that duplicating (tape) media can significantly ease your vaulting policy.
- Consider your resource policies. Your system and partition policies need to be synchronized and well balanced if you want to achieve good performance.

- Define your migration and release policies regarding the characteristics of files on each individual file system, such as how much are the files in use, are these “working” files or files that simply need to be retained safely, how often do you want these files to be migrated, and so on.
- Consider the effect of your policies on your FSE installation and tune it accordingly.

Configuration Basics

All FSE resources, namely FSE library, FSE drive, FSE media pool, FSE medium, and FSE medium volume, must be configured before you can start using FSE. You configure FSE resources using configuration plain text files. See also the *FSE Management Console Reference Guide* for a graphical interface to some of these tasks..

Configuration Files

The plain text file format, used for configuration files, has the following advantages:

- Simple editing and modification
To edit and modify your configuration files, you can use any tool that can save files as plain text.
- Similar syntax for each component
You configure your resources using corresponding syntax, considering the requirements of a particular resource.
- Parsing
You can collect data from existing, already applied configuration files using parsing utilities and scripts.
- Use of comments
You can add your own comments to configuration files. Comments are introduced with a hash character (#) in order to be marked and preserved.

FSE configuration file elements

Each configuration file consists of particular elements, which can be grouped as follows:

Element	Description
type	Identification of configurable object type
header	Contains version number, date of application, user name
sections	Related parameters are logically grouped to sections
variables	Used to set parameters
comments	User's comments; they are marked with a # sign
qualifiers	Amount of data: KB, MB...; time units: m, h...; percentage: %

The following example shows the elements as they are represented in the configuration file

```
[FSE Pool] <----- type
[Version = 1] <----- header
[Date = "2002/08/05 15:00:12"]
[User = "MaxMeier@MyCompany.com"]

# This is where actual parameters start <----- comment
Name          = "My_Little_Pool" <----- main section

MediaFamily    = LTO          # AIT | LTO
PoolType       = Regular     # Regular | WORM | Backup

[DefaultFormat] <----- subsection

BlockSize      = 32KB <----- qualifier
NumOfVolumes   = 1
SysVolLocation = NONE       # NONE | FIRST | MIDDLE | LAST
SizeOfSysVol   <- = 512MB <----- variable
```

Templates

For each resource type, a default configuration template is available so that you do not need to create configuration files from scratch. The templates already contain most of the required parameters with remarks on the appropriate use of configuration variables, and details of maximum and minimum values. When configuring a new FSE resource, make a copy of the appropriate template, modify the copy according to the characteristics of the resource, and then configure the resource using the copy.

Templates are located in the following directories for consolidated FSE systems or the FSE server:

- Linux: `/opt/fse/newconfig/`
- Windows: `%InstallPath%\newconfig`

To configure your resources and your policies, use FSE commands as described in “[Using FSE Commands for Configuring Resources and Policies](#)” on page 41. For more information on FSE commands, see “[About FSE Commands](#)” on page 38.

See “[FSE Configuration Templates](#)” on page 341 for details on default settings in FSE configuration files and for maximum and minimum parameter values.

Configuration Database

The FSE configuration is stored in the Configuration Database. This database maintains a configuration history for each configured FSE resource thus providing multiple revisions of configuration files. When you reconfigure your resources, the new configuration is marked with new revision number.

Additionally, you are able to revert from the current to any other configuration revision at any time. You only need to retrieve the desired revision from the configuration database, and then reapply it. The applied configuration is again saved as new configuration revision.

Note: Do not alter the contents of the Configuration Database, as it can cause failures. Use the FSE command-line interface to change the configuration as described in section “[Using FSE Commands for Configuring Resources and Policies](#)” on page 41.

The Configuration Database is located in the following directory:

- Linux: `/var/opt/fse/cfg/`
- Windows: `%InstallPath%\var\cfg`

Resource Management Database

Once FSE resources are configured, the data from the configuration files are written to the Resource Management Database (RMDb). This database keeps all relevant information on FSE resources, such as the current status, usage information, and characteristics.

When an FSE resource is removed from the configuration, all data referring to it is removed from Resource Management Database accordingly.

The Resource Management Database is located in the following directory:

- Linux: `/var/opt/fse/rmdb/`
- Windows: `%InstallPath%\var\rmdb`

About FSE Commands

FSE commands are used to perform standard FSE tasks. They communicate with FSE through the FSE Management Interface daemon/service.

Note: To use FSE commands:

On Linux systems: you have to be logged on to the system as “root”.

On Windows systems: you must have administrator rights. These privileges are granted only if you are locally logged to the system as Administrator. Remote Desktop cannot be used for this task.

FSE command option syntax is based on standard Linux command syntax. Typically, on Windows, options start with a ‘/’ character, while FSE command syntax demands a ‘--’ string as the option prefix.

All FSE commands include online help that lists available options and their use. For example, if you want to see help for the `fsepartition` command, you would use:

Linux:

```
# fsepartition --help
```

Windows:

```
C:\Program Files\Hewlett-Packard\FSE\bin>fsepartition --help
```

FSE commands have the following characteristics:

- One command per FSE component

FSE commands are based on a principle of one command per FSE component, for example `fsepartition`, `fsepool`, `fsemedium`, `fselibrary`, `fsedrive`, and so on.

- Common options

All commands share similar options, for example `--help`, `--add`, `--modify`, `--remove`, `--list`, `--show`, `--status`, and so on.

- Short and long options

FSE commands accept short (`-o`) and long (`--option`) formats of options. For example, the `--a` and `-add` options have the same effect. (Note that some options do not have short variants.)

- Location

On Linux, FSE commands are located in the `/opt/fse/sbin` directory. On Windows, they are located in the `%InstallPath%\bin` directory. (`%InstallPath%` is `Program Files\Hewlett-Packard\FSE` by default, but you can change this path by choosing Custom setup during FSE installation.)

- FSE command-line reference

FSE command-line reference is available on Linux as UNIX man pages, and on Windows as Windows/HTML help. Both contain the same detailed description of all FSE commands, including their options, arguments, and invocation syntax. They are also included in this guide in “[Command Reference](#)” on page 251.

- Man pages on Linux

To use FSE man pages, add the following line to the corresponding `.bashrc` file:

```
export MANPATH=$MANPATH:/opt/fse/man
```

Use `man` followed by the name of the FSE command to display the man page for the selected command.

For a list of all FSE commands and files used by FSE (such as a list of log files) see the “`fseintro`” man page.

- Windows/HTML help

You can find FSE help under the submenu of **Hewlett-Packard FSE (Start > Program Files > Hewlett-Packard > FSE > Docs)**.

For a list of all FSE commands and files used by FSE (such as a list of log files), see `fseintro.html`.

FSE Commands List

The following table lists all FSE commands with a brief description. For a detailed description of command functionality and invocation syntax, see “[Command Reference](#)” on page 251.

Command	Description	See page
fsebackup	Performs a backup of the FSE installation, including HSM file systems and databases	page 265
fsebak	Helper tool for controlling FSE databases during backup and restore	page 270
fsecheck	Checks File System Catalog (FSC) consistency with HSM file systems and FSE media	page 275
fsedrive	Used for configuration of drives	page 278
fsefile	Used for management of files on HSM file systems, for example, recalling files (including older file generations), triggering migration and release of files, triggering deletion of files	page 283
fsejob	Shows running/queuing jobs, changes priority of jobs, and aborts jobs	page 291
fselibrary	Used for configuration of libraries, shows library inventory, rescans library inventory	page 296
fsemedium	Used for configuration of FSE media, formatting and initializing, checking and modifying media status, reorganizing medium volumes, and performing maintenance tasks on FSE media	page 302
fsepartition	Used for configuration of FSE partitions, shows partition status	page 313
fsepool	Used for configuration of media pools and displaying the reorganizational scan results for media pools	page 322
fsererecover	Recovers File System Catalog (FSC) from FSE media, and HSM file systems from the Name Space Catalog	page 328
fserestore	Performs a restore of the FSE installation from backup media	page 332
fsesystem	Used for configuration of system options and shows overall system status	page 334

Using FSE Commands for Configuring Resources and Policies

You configure FSE resources and your policies using the command-line commands. You can use each command with similar options, such as `--list`, `--add`, `--remove`, `--show`, `--modify`, depending on your needs. You can repeat the commands if needed, for example to add multiple media to a media pool.

The following examples show how to configure a new FSE component, how to re-configure your current configuration and revert to the previous FSE configuration file revision.

Example of configuring a new component in FSE

You want to configure a new partition on Linux. You will use an existing template, modify it and then apply it to configure a new FSE partition. You can use the described commands in a similar way when configuring any of the FSE resources. Note that the procedure for the new partition configuration is generally the same for both operating systems, except for the location of the templates, which can be found on Linux in the `/opt/fse/newconfig` directory, and on Windows in `%InstallPath%\newconfig` directory.

1. Locate the wanted template in the `newconfig` directory and copy it to the selected directory, in this example named `MyConfiguration`, using the following commands:

Linux:

```
# cd MyConfiguration
# cp /opt/fse/newconfig/fse-partition.cfg
  /PartitionCfgFileName
```

Windows:

```
# cd MyConfiguration
# copy C:\Program Files\Hewlett-Packard\FSE\newconfig\
  fse-partition.cfg PartitionCfgFileName
```

2. Edit the configuration file and modify the parameter values relevant for the new FSE partition.
3. To configure a new partition, apply the modified template using the following:

```
# fsepartition --add PartitionCfgFileName
```

You have configured a new FSE partition. This FSE partition configuration is stored in the configuration database as the first respective configuration revision.

Example of modifying an FSE configuration file

You want to change the parameter values of your migration policy. Since these parameters are written in the FSE partition configuration file, you will use the `fsepartition` command followed by the relevant command options, which you can use in a similar way when re-configuring any of the FSE resources.

1. Check the names of the configured partitions in order to get the exact name of the partition for which you want to modify a migration policy, using the following command:

```
# fsepartition --list [--detail]
```

2. Copy the parameters of the selected partition (in this example named `PartitionName`) to a file, named for example `MyFile` in the current directory:

```
# fsepartition --show PartitionName > MyFile
```

3. Edit the created configuration file and modify the parameter values relevant for your migration policy using any text editor.
4. Apply the new configuration using the following command:

```
# fsepartition --modify PartitionName MyFile
```

This FSE partition configuration is stored in the configuration database with a new revision number.

Example of reverting to the previous FSE configuration file revision

You want to revert from the current to the previous configuration revision 4. You will retrieve the chosen fourth revision from the configuration database, and then reapply it.

1. Check the names of the configured partitions in order to get the exact name of the partition for which you want revert to the previous revision, using the following command:

```
# fsepartition --list [--detail]
```

2. Check the configuration history for the target partition. Use the following commands to get the list of all configuration revisions for the target partition in the configuration database.

```
# fsepartition --show PartitionName --history
```

3. Copy the contents of the chosen configuration revision (in our case 4) for FSE partition to a file, named for example `MyFile` in the current directory:

```
# fsepartition --show PartitionName 4 > MyFile
```

4. Apply the chosen configuration revision using the following command:

```
# fsepartition --modify PartitionName MyFile
```

This FSE partition configuration is stored in the configuration database with a new revision number.

Configuration Procedure

The general procedure for FSE configuration consists of the following:

- Editing the configuration file in plain text format—entering parameter values.

Note: To avoid difficulties like missing or inappropriate parameters, it is recommended that you refer to FSE configuration templates. They already contain all required parameters and provide remarks on appropriate use of configuration variables.

- Applying the configuration file with the appropriate command.

Different FSE resources are related to each other and therefore have to be configured in the appropriate order, as follows:

1. [Step 1: Configuring FSE Libraries and FSE Drives](#), page 44 (related to the FSE library)
2. [Step 2: Configuring FSE Media Pools](#), page 52
3. [Step 3: Preparing HSM File Systems](#), page 52
4. [Step 4: Configuring FSE Partitions](#), page 55 (related to the FSE media pools)
5. [Step 5: Preparing FSE Media](#), page 59 (related to a single FSE media pool, FSE partition and FSE library)
6. [Step 6: Mounting HSM File Systems](#), page 60 (related to a single FSE partition)

Once the media are initialized, medium volumes are configured automatically. They are related accordingly to the medium, media pool and partition.

Note: To use FSE commands:

On Linux systems: you have to be logged on to the system as “root”.

On Windows systems: you must have administrator rights. These privileges are granted only if you are locally logged to the system as Administrator. Remote Desktop cannot be used for this task.

Step 1: Configuring FSE Libraries and FSE Drives

About SCSI identification

For the FSE library and drive configuration, you need to identify the SCSI addresses of libraries and drives that will be used by the FSE installation. Use the `fse_scsi --scan` command to find addresses. The format of the identification strings that are displayed depend on the operating system where the `fse_scsi` command is run.

Linux

FSE on Linux uses a SCSI generic (`sg`) driver for accessing SCSI libraries and drives. The `sg` driver is an upper level SCSI subsystem device driver that enables user applications to send SCSI commands to relevant devices. It is capable of addressing up to 256 SCSI devices.

When configuring libraries and drives, you need to specify their control device file names: each device file name corresponds to one attached SCSI device. Control device file names are located in the `/dev` directory. Therefore, the specified control device for library and drive takes the form `dev/sgX`, where `X` is a number between 0 and 255.

Windows

Before configuring FSE drives, ensure that on a consolidated FSE implementation or FSE server host the drivers of all tape drives to be used as FSE drives are disabled. For more information on disabling the tape drive drivers, see the *FSE Installation Guide*.

FSE on Windows identifies attached SCSI devices (libraries and drives) by specifying port, bus, target, and LUN in the form of `id:P:B:T:L`, where:

- P* SCSI port number (HBA number) to which the device is connected
- B* SCSI bus/channel number on the HBA to which the device is connected
- T* Target ID to which the device is connected
- L* Logical unit number (LUN) of the device

Library and drive configuration procedure

1. Before you start configuring FSE libraries and drives, you must determine their control device files. On Linux, you can check the `/proc/scsi/scsi` file for a list of available SCSI devices. Use the `fse_scsi` tool to get the desired output:

Linux

```
# /opt/fse/sbin/tools/fse_scsi --scan
** /dev/sg0          id:0:0:0:0

SCSI device ID: port 0, ch 0, id 0, lun 0, type Disk 0
Device: ACARD      MirrorSmart 38G  3.1M ser #:589C589C04CB15BE
std :2

** /dev/sg1          id:1:0:0:0

SCSI device ID: port 1, ch 0, id 0, lun 0, type Disk 0
Device: IFT        IFT-7200          0145 ser #: 3293971          std
:4

** /dev/sg2          id:3:0:1:0

SCSI device ID: port 3, ch 0, id 1, lun 0, type Tape 1
Device: SONY       SDX-700C          0102 ser #:0006882752 std :2

** /dev/sg3          id:3:0:2:0

SCSI device ID: port 3, ch 0, id 2, lun 0, type Tape 1
Device: SONY       SDX-500C          0203 ser #:0004894242 std :2

** /dev/sg4          id:3:0:3:0

SCSI device ID: port 3, ch 0, id 3, lun 0, type Tape 1
Device: SONY       SDX-500C          0203 ser #:0003830095 std :2

** /dev/sg7          id:3:0:6:0

SCSI device ID: port 3, ch 0, id 6, lun 0, type Medium Changer 8
Device: GRAU       ITL-340i          1.5  ser #:0000000000 std :2

Done!
```

Windows

```
C:\Program Files\Hewlett-Packard\FSE\bin>fse_scsi --scan

** \\.\Scsi2:          id:2:0:2:0
```

```
SCSI device ID: port 2, ch 0, id 2, lun 0, type Tape 1
Device: SONY      SDX-500C      0203 ser #: 0000906652 std : 2
```

```
** \\.\Scsi2:      id:2:0:3:0
```

```
SCSI device ID: port 2, ch 0, id 3, lun 0, type Tape 1
Device: SONY      SDX-500C      0203 ser #: N/A std : N/A
```

```
** \\.\Scsi3:      id:3:0:6:0
```

```
SCSI device ID: port 3, ch 0, id 6, lun 0, type Medium Changer
8
Device: GRAU      ITL-340i      1.5 ser #: N/A std : N/A
```

```
Done!
```

Note: On Linux systems, if the Library Agent is running, the device for the library used by the agent is not seen in the list.

2. Enter the values you get from the output of the above command to the configuration files for library and drives, and check that all other parameters match the characteristics of your library and drives.
See “[Library and tape drive configuration examples](#)” on page 47 for reference. You will configure your media pool according to your drive configuration.

Note: To use disk media, no library and drive configuration is needed.

3. Configure the FSE library using the command:

```
# fselibrary --add LibraryCfgFile
```
4. Once the library is configured, a new Library Agent is started. Media that are physically located in the library are added to the FSE configuration automatically, but they are not yet assigned to the media pool.
5. Configure the new FSE drive using the command:

```
# fsedrive --add DriveCfgFile
```


The following are examples of configuring a library and supported tape drives. The examples present configurations for both supported operating systems, despite the fact that the differences in configuration are minor (in these particular examples, the difference is the *ControlDevice* file).

Library and tape drive configuration examples

Example of a library configuration file on Linux:

```
[FSE Library]
[Version = 1]
[Date = "2003/04/11 15:24:32"]
[User = "MaxMeier@MyCompany.com"]

#
# Specify a name for the tape library and a host name, where it
# resides. Use full host name.
#
Name          = "HP MSL5000 #2 LTO"
Host          = "fse01.MyCompany.com"

#
# Control device is identification of library's
# (media changer's) on the specified system.
#
# Tip: use fse_scsi --scan to list available SCSI devices
#
# Linux: /dev/sgN
# Windows: id:P:B:T:L
#
ControlDevice = "/dev/sg0"
LibType       = SCSI
# AIT - Sony AIT Family
# LTO - LTO Ultrium
MediaFamily   = AIT
#
# Specify the slots that will be used with this library.
# You can specify ranges of slot numbers and individual
# slot numbers.
#
# Example:
# Slots = (1-15, 20, 22, 24, 26-30)
#
[Slots]
Slots = (1-35)
```

Example of a library configuration file on Windows:

```
[FSE Library]
[Version = 1]
[Date = "2003/04/11 15:24:32"]
[User = "MaxMeier@MyCompany.com"]

#
# Specify a name for the tape library and a hostname, where it
# resides. Use full host name.
#
Name           = "GTL340i-AIT"
Host           = "fse01.MyCompany.com"

#
# Control device is identification of library's
# (media changer's) on the specified system.
#
# Tip: use fse_scsi --scan to list available SCSI devices
#
# Linux: /dev/sgN
# Windows: id:P:B:T:L
#
ControlDevice = "id:0:0:0:0"

LibType        = SCSI

# AIT - Sony AIT Family
# LTO - LTO Ultrium
MediaFamily    = AIT
#
# Specify the slots that will be used with this library.
# You can specify ranges of slot numbers and individual
# slot numbers.
#
# Example:
# Slots = (1-15, 20, 22, 24, 26-30)
#
[Slots]
Slots = (1-35)
```

Example of a regular AIT tape drive configuration file on Linux:

```
[FSE Drive]
[Version = 1]
[Date = "2003/04/11 15:30:12"]
```

```
[User = "MaxMeier@MyCompany.com"]

Name          = "AIT2-01"
Library       = "GTL340i-AIT"
MediaFamily   = AIT
DriveIndex    = "1"

Capabilities = ()

Host "fse01.MyCompany.com"
{
    ControlDevice = "/dev/sg1"
}
```

Example of a regular AIT tape drive configuration file on Windows:

```
[FSE Drive]
[Version = 1]
[Date = "2003/04/11 15:30:12"]
[User = "MaxMeier@MyCompany.com"]

Name          = "AIT2-01"
Library       = "GTL340i-AIT"
MediaFamily   = AIT
DriveIndex    = "1"

Capabilities = ()

Host "fse01.MyCompany.com"
{
    ControlDevice = "id:0:0:2:0"
}
```

Example of an SAIT tape drive configuration file on Linux:

```
[FSE Drive]
[Version = 1]
[Date = "2003/04/11 15:30:12"]
[User = "MaxMeier@MyCompany.com"]

Name          = "SAIT-01"
Library       = "GTL340i-SAIT"
MediaFamily   = SAIT
DriveIndex    = "1"

Capabilities = ()
```

```
Host "fse01.MyCompany.com"
{
    ControlDevice = "/dev/sg5"
}
```

Example of an SAIT tape drive configuration file on Windows:

```
[FSE Drive]
[Version = 1]
[Date = "2003/04/11 15:30:12"]
[User = "MaxMeier@MyCompany.com"]

Name          = "SAIT-01"
Library       = "GTL340i-SAIT"
MediaFamily   = SAIT
DriveIndex    = "1"

Capabilities = ()

Host "fse01.MyCompany.com"
{
    ControlDevice = "id:4:0:2:0"
}
```

Example of an AIT-WORM tape drive configuration file on Linux:

```
[FSE Drive]
[Date = "2003/04/11 15:30:12"]
[User = "MaxMeier@MyCompany.com"]

Name          = "WORM 01"
Library       = "GTL340i-AIT"
MediaFamily   = AIT
DriveIndex    = "1"

Capabilities = (WORM)

Host "fse01.MyCompany.com"
{
    ControlDevice = "/dev/sg3"
}
```

Example of an AIT-WORM tape drive configuration file on Windows:

```
[FSE Drive]
[Date = "2003/04/11 15:30:12"]
[User = "MaxMeier@MyCompany.com"]
```

```

Name          = "WORM 01"
Library       = "GTL340i-AIT"
MediaFamily   = AIT
DriveIndex    = "1"

Capabilities = (WORM)

Host "fse01.MyCompany.com"
{
    ControlDevice = "id:1:0:4:0"
}

```

Example of an LTO Ultrium tape drive configuration file on Linux:

```

[FSE Drive]
[Date = "2003/04/11 15:30:12"]
[User = "MaxMeier@MyCompany.com"]

Name          = "LTO 01"
Library       = "GTL340i-LTO"
MediaFamily   = LTO
DriveIndex    = "1"

Capabilities = ()

Host "fse01.MyCompany.com"
{
    ControlDevice = "/dev/sg4"
}

```

Example of an LTO Ultrium tape drive configuration file on Windows:

```

[FSE Drive]
[Date = "2003/04/11 15:30:12"]
[User = "MaxMeier@MyCompany.com"]

Name          = "LTO 01"
Library       = "GTL340i-LTO"
MediaFamily   = LTO
DriveIndex    = "1"

Capabilities = ()

Host "fse01.MyCompany.com"
{
    ControlDevice = "id:0:2:0:0"
}

```

}

Step 2: Configuring FSE Media Pools

For detailed information on creating a specific media pool type, see “[Configuring a Media Pool](#)” on page 71. Generally, you configure a FSE media pool as follows:

1. Set the proper values of FSE media partitioning parameters in the media pool configuration file. Check that all other parameters match the characteristics of your media.
2. Add the FSE media pool to FSE using the command:

```
# fsepool --add MediaPoolCfgFile
```

Step 3: Preparing HSM File Systems

You need to create and prepare one HSM file system for each FSE partition that you plan to configure.

Apart from newly-created HSM file systems, you can also include the existing non-HSM file systems in the FSE system. All data on such file systems is preserved and put under the FSE system’s control. On Linux systems, existing Ext3 file systems residing on LVM volumes can be used, provided that they are remounted as HSM file systems after the FSE partitions that belong to them are configured. On Windows systems, you can use existing disk volumes (partitions); you only need to dismount the Windows native NTFS, and configure the corresponding FSE partitions afterwards.

Preparing HSM file systems on Linux

On Linux, HSM file systems must be created on the LVM volumes.



Caution: Use the LVM-command set with caution, as certain commands can destroy existing file systems. Make sure you specify the correct device or partition.

LVM configuration procedure overview:

1. Define and initialize LVM physical volumes. LVM commands for managing LVM physical volumes begin with letters `pv` (physical volume) and are placed in `/sbin` directory.

2. Once physical volumes are configured, you have to create and initialize the LVM logical volume groups; these use the space on physical volumes. Commands for managing LVM logical volume groups begin with letters `vg` (volume group) and are placed in `/sbin` directory.
3. Create and initialize LVM logical volumes for the file systems you are going to use inside the FSE installation. Commands for managing LVM logical volumes begin with letters `lv` (logical volume) and are located in the `/sbin` directory.

For detailed instructions on LVM use, see the LVM man pages and web site <http://tldp.org/HOWTO/LVM-HOWTO/>.

Example of creating a file system on an LVM volume

After the LVM logical volumes have been successfully initialized, you need to create file systems on them with the `mkfs` command.

To create the file systems with `mkfs`, use the options `-t ext3` for Ext3 file system creation (with journal).

Tip: If you want to check which properties your file system will have once it is created, but do not want to actually set one up, run the `mkfs` command with the `-n` switch.

- Example output of checking the `example_fs` file system values:

```
# mkfs -t ext3 -n /dev/fse_vg/example_fs
mke2fs 1.27ea (14-Mar-2002)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
456064 inodes, 911680 blocks
45584 blocks (5.00%) reserved for the super user
First data block=0
28 block groups
32768 blocks per group, 32768 fragments per group
16288 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736
```

Note: The number of inodes in the `mkfs` output corresponds to the expected maximum number of files on the file system. If this number is not satisfactory, you can explicitly tell `mkfs` to reserve a certain number of inodes with the `-N` option.

Note that upper limit for the number of inodes is affected by two factors: the file system size and the file system block size. On a file system with size F_s and block size B_s the maximum size of inodes I_n is determined by the equation $F_s = B_s * I_n$. If you specify a number bigger than I_n , the `mke2fs` command creates I_n inodes.

For example, to create one million inodes, you have to specify the `-N 1000000` option. With a block size of 4096 bytes, the file system size must be equal to or greater than 3.8 GB (4,096,000,000 bytes) for this number of inodes to be created:

```
# mkfs -t ext3 -N 1000000 /dev/vg_fse/example_fs
```

Creating HSM file systems on Linux

To create an HSM file system, proceed as follows:

1. Use the following command to create an HSM file system on top of the LVM logical volume `fsefs_01`. In the given example, the HSM file system will store a maximum of 10,000,000 files. You should use the LVM logical volume name according to your actual LVM volume configuration and number of inodes according to your HSM file system usage expectations. Consider the limitation on the number of inodes that can be created on a file system with specific total size and specific block size:

```
# mke2fs -b 4096 -j -N 10000000 /dev/fse_vg/fsefs_01
```

2. Create HSM file systems on all other LVM logical volumes that will be used as FSE partitions. Use values according to the purpose of the particular FSE installation.



Caution: Once the file system has been put into use, the number of inodes on file system cannot be changed without further formatting (which would lead to data loss).

Preparing HSM file systems on Windows

Creating disk volumes for HSM file systems

To create a separate volume (partition) for an HSM file system:

1. In Windows Control Panel, double-click **Administrative Tools**, and then double-click **Computer Management**.

2. Click **Disk Management**, right-click an unallocated region of a basic disk, and then click **Create Partition**, or right-click free space in an extended partition, and then click **Create Logical Drive**.
3. In the Create Partition Wizard, in each of the first three steps set the appropriate options and move to the next step by clicking **Next**.
4. In the Assign Drive Letter or Path window, select either the **Assign a drive letter** or **Mount this volume at an empty folder that supports drive paths** option and click **Next**.
5. In the next step of the Create Partition Wizard, accept the default options by clicking **Next**.
6. Click **Finish** to close the Create Partition Wizard and format the newly created partition.

For information on the access modes that are available for HSM file systems on Windows systems, see “[HSM File System Access Modes](#)” on page 146.

The next step, configuring an FSE partition, is described in “[Configuring an FSE partition for a Windows FSE client](#)” on page 56.

Step 4: Configuring FSE Partitions

Configuring an FSE partition for a Linux FSE client

After you have configured FSE media pools and prepared an HSM file system, you can configure FSE partitions as follows:

1. The FSE partition configuration contains a **major:minor** number pair for the HSM file system. To get these two parameters, enter the command below:

```
# ls -l /dev/fsesdb_vg/fs02
brw-rw---- 1 root disk 58, 12 2003-12-04 00:48
/dev/fsesdb_vg/fs02
```

Note: If the file system is located on an FSE client, the command has to be run on the client machine. It is mandatory to provide the information where the file system is located.

2. You can find the major number in the fifth column and the minor number in the sixth column; edit the configuration file accordingly.

```
#
# Where is the HSM file system?
```

```
#
Client " HSMFS HOST NAME "
{
FileSystemID = 58:12
Type = Regular
# Regular - regular read/write file system
# WORM - Write-Once-Read-Many file system

# uncomment and configure if Type = WORM :
# WORMTimeout = 450s
}
```

3. Add the new FSE partition:

```
# fsepartition --add PartitionCfgFile>
```

Once the partition is configured, a new Partition Manager is started for the newly configured partition.

4. Optionally, check the status of the respective partition and Partition Manager using the following command:

```
# fsepartition --status PartitionName
```

Configuring an FSE partition for a Windows FSE client

When configuring a volume for the HSM file system on Windows, verify that the prerequisites are met, and consider the following points about volume configuration.

About disk volumes on Windows

A Windows physical disk can be subdivided into volumes. Each volume has its unique volume ID, for example

Volume{729f2319-ba82-11d7-a2d5-00e01884ee37}. These volume IDs only change when the partitioning scheme of the physical disk changes. You get this unique volume ID by using the `mountvol` command.

Note: A Windows FSE client uses volume IDs for identifying configured partitions.

FSE partition configuration procedure

To configure an FSE partition on a Windows client, proceed as follows:

1. At the command prompt, type the `mountvol` command. Select the volume ID of the FSE partition.

This is an example of the `mountvol` command and its output:

```
C:\Program Files\Hewlett-Packard\FSE\bin>mountvol
...
Possible values for VolumeName along with current mount points are:
  \\?\Volume{97d09c72-b208-11d7-8f8b-806d6172696f}\
    C:\
  \\?\Volume{97d09c73-b208-11d7-8f8b-806d6172696f}\
    D:\
  \\?\Volume{f6f3e1b6-b2ac-11d7-9c99-00b0d02fadef}\
    D:\hsm_fs_folder
  \\?\Volume{97d09c71-b208-11d7-8f8b-806d6172696f}\
    F:\
  \\?\Volume{97d09c70-b208-11d7-8f8b-806d6172696f}\
    A:\
```

- From the output, copy the `\\?\Volume{number}\` value before the line `D:\hsm_fs_folder`.

In the example above, you would copy the value:

```
\\?\Volume{f6f3e1b6-b2ac-11d7-9c99-00b0d02fadef}\
```

- Specify this volume ID without the `\\?\` prefix and the trailing backslash in the `FileSystemID` field of the partition configuration file. Example of the required format of the volume number is shown in this excerpt of the partition configuration file:

```
#
Name = "win-ext"
#
# Where is the HSM file system?
#
Client "client.company.com"
{
  FileSystemID="Volume{f6f3e1b6-b2ac-11d7-9c99-00b0d02fadef}"
}
```

- Add the new FSE partition to the FSE server:

```
C:\>fsepartition --add PartitionCfgFile
```

Once the partition is configured, a new Partition Manager is started for the newly configured partition.

- Determine the name of the disk volume that belongs to the FSE partition using `mountvol` first, and then invoke the following command:

```
C:\>fse --dismount-ntfs <VolumeName>
```

The command will detach the automatically attached NTFS from the volume and allow it to be mounted later as an HSM file system.

6. Optionally, check the status of the FSE partition and the appropriate Partition Manager:

```
C:\>fsepartition --status PartitionName
```

Configuring a WORM FSE partition

FSE provides an option to configure an HSM file system as a WORM (Write-Once Read-Many) file system. The Type parameter in the FSE partition configuration file defines whether the type of a file system is “regular” or “WORM”.

On an FSE WORM file system a file can be written only once but can be read many times. A WORM file system prevents both files and their metadata, including access control rights and file attributes, from being modified, regardless of access permissions. Consequently, it is not possible to remove files from directories or rename them, but only to add new files.

There is also a timeout parameter (WORMTimeout) that defines the length of time after a file is closed during which it can still be changed. During this time frame, the file’s contents and its metadata can be modified; the file can be renamed, moved, or removed. Once the waiting loop has ended, the file, its metadata, and location can never be modified again. This parameter is commented out in the template file. It must be uncommented if the file system type is “WORM”. The timeout value must be set according to the specific archiving environment.

Note: A WORM partition must be related to media pools containing WORM media only, if it is used for accounting or legislation applications.

Example of a WORM FSE partition configuration file on Linux systems:

```
Client "fseclient.company.com"
{
FileSystemID = 58:2 # 58:1
Type = WORM
# Regular - regular read/write file system
# WORM - Write-Once-Read-Many file system

# uncomment and configure if Type = WORM :
WORMTimeout = 60s
}
#
# Specify which media pool(s) will be used for storing data to.
#
# NOTE: The number of pools specified defines
```

```
#the number of copies to be done.
#
Pools = ("MYWORMPOOL1", "MYWORMPOOL2")
```

Example of a WORM FSE partition configuration file on Windows systems:

```
# Client "winsrv.company.com"
{
FileSystemID = "Volume{97d09c72-b208-11d7-8f8b-806d6172696f}"
Type = WORM
# Regular - regular read/write file system
# WORM - Write-Once-Read-Many file system

# uncomment and configure if Type = WORM :
WORMTimeout = 60s
}

#
# Specify which media pool(s) will be used for storing data to.
#
# NOTE: The number of pools specified defines
#the number of copies to be done.
#
Pools = ("WINWORMPOOL1", "WINWORMPOOL2")
```

Step 5: Preparing FSE Media

Once all FSE resources except media are configured, follow the procedure below to prepare FSE media. For specifics on preparing disk media, see [“Managing Media”](#) on page 69.

1. Create an entry for the FSE medium and add it to the already configured media pool:

```
# fsemedium --add Barcode --pool PoolName
```
2. Physically place the FSE medium into a free library slot, and then trigger a physical inventory update (barcode rescan):

```
# fselibrary --update-inventory LibraryName --rescan
```
3. Format the medium using the command below. If the medium is already properly formatted (partitioned), skip this step and proceed with initialization.

```
# fsemedium --format Barcode
```



Caution: Formatting erases all data on the medium.

In case the medium has already been formatted and used by the FSE installation, you need to use an additional `--force` option:

```
# fsemedium --format Barcode --force
```

Note: `--force` will allow formatting of the FSE medium only if there is no valid FSE data stored on it (% used of all medium volumes is zero).

During formatting, the FSE medium is partitioned into FSE media volumes.

4. Once the medium is formatted, you must initialize the FSE media volumes created on it. For more information on media volumes, see “[FSE Medium Volumes](#)” on page 78.

For regular initialization use the following command:

```
# fsemedium --init Barcode
```

For forced initialization use the following command:

```
# fsemedium --init Barcode --force
```

Note: You can execute both formatting and initializing at once by using:

```
# fsemedium --format Barcode --init
```

Once FSE media are formatted and initialized, the HSM file system can be mounted.

Step 6: Mounting HSM File Systems

After you have configured FSE resources, you have to mount the HSM file systems on machines that host HSM file systems, that is on a consolidated FSE implementation or on an external FSE client, which can be either Windows or Linux. Once the file systems are mounted, the FSE installation is ready to be used.

Mounting on Linux

To mount the HSM file systems, proceed as follows:

1. Check the status of the FSE partitions using the following command:

```
# fsesystem --status
```

The status for all configured partitions must be “ready”.

2. Add file system entries for the HSM file systems to the `/etc/fstab` file. Follow the procedure for automating the mounting of the example HSM file system discussed in the *FSE Installation Guide*.

- a. Create a directory that will serve as a mount point for the FSE file system. Mount point directories are usually created in the `/fse` directory:

```
# mkdir /fse/fsefs_01
```

- b. Add the following line to the `/etc/fstab` file:

```
/dev/vg_fse/fsefs_01 /fse/fsefs_01 fsefs noauto 0 0
```

You have to use the values as specified above:

- The keyword `hsmfs` in the third column refers to the file system type, which is the HSM file system.
- The keyword `noauto` refers to the file system mounting options. An HSM file system cannot be automatically mounted at system boot time before FSE's start-up (`noauto`).
- Do not change the `0 0` values in the fifth and sixth columns.

Use the values in the first and second column according to your actual LVM volume configuration.

- c. Create mount points and add file system entries for all other existing HSM file systems that you are going to use.

3. Restart FSE and mount all existing HSM file systems with entries in the `/etc/fstab` file with the following command:

```
# fse --restart
```

Mounting on Windows

Mounting disk volumes on Windows is in fact performed by the operating system automatically on its first use. When an object on the disk volume (HSM file system) is accessed, the HSM file system is automatically mounted in Full Access Mode (FAM) (if the FSE processes are already running) or in Limited Access Mode (LAM) (if the FSE processes are not running yet). For more information on HSM file system access modes, see “[HSM File System Access Modes](#)” on page 146.

Additionally, if the FSE processes are running, use the following command to mount an HSM file system in Full Access Mode before it is actually accessed:

```
C:\>fse --mount <MountPoint> <VolumeName>
```

In either case you can later check if the HSM file system is available as follows:

1. At the Command Prompt, set the current directory to the mount point of the HSM file system and run the `dir` command.
2. Invoke the following command:

```
C:\>fsepartition --list
```

In the command output, the corresponding FSE partition should have its status set to `mounted`.

Reconfiguring FSE Interprocess Communication

FSE interprocess communication is configured according to the FSE system deployment (consolidated, distributed, or mixed FSE implementation) and the network type used for FSE interprocess communication (ordinary LAN, private network). The initial configuration must be performed after the FSE software package installation, before the FSE installation is put into operation for the first time. Afterwards, if the deployment or the network type is changed, the communication settings must be reconfigured.



Caution: An appropriate configuration of the FSE interprocess communication is of crucial importance for normal FSE operation. An improperly modified communication configuration may lead to a non-operating FSE installation.

FSE interprocess communication settings consist of two plain text files installed on each FSE host. The files and their locations are the following:

Command	Location on Linux	Location on Windows
<code>services.cfg</code>	<code>/etc/opt/fse</code>	<code>%InstallPath%\etc</code>
<code>omniORB.cfg</code>	<code>/etc/opt/fse</code>	<code>%InstallPath%\etc</code>

You can change the default path where FSE searches for the `omniORB.cfg` file using the `OMNIORB_CONFIG` environment variable.

The reconfiguration procedure

The general procedure for reconfiguring interprocess communication is as follows:

1. Stop all FSE processes with the `fse --stop` command.

Note that you must first stop the processes on external FSE clients, and then stop the processes on a consolidated FSE implementation or on the FSE server.

2. Stop the `omniNames` service with the `omniNames --stop` command.
3. Purge the `omniNames` (CORBA) service data files that are located in the `/var/opt/fse/omniORB` directory (on Linux) or the `%InstallPath%\var\omniORB` directory (on Windows).
4. Modify the `services.cfg` or `omniORB.cfg` file on all FSE hosts according to the changes in your FSE system deployment or changes of the network type employed. The required modifications are described in the following sections.

You can use FSE's `fse_net` tool to retrieve the parameters required for the configuration. For more information on `fse_net`, see “[FSE Troubleshooting Tools](#)” on page 231.

Note: SUSE LINUX specific

On a SUSE LINUX system, do not run `YaST2` after you have configured this FSE host to use a private network for the FSE interprocess communication. Running `YaST2` modifies `/etc/hosts` in such a way that the subsequent FSE system start-ups fail.

Alternatively, you can modify `/etc/sysconfig/suseconfig` by changing the line `CHECK_ETC_HOSTS="yes"` to `CHECK_ETC_HOSTS="no"`. You can then run `YaST2` without affecting FSE operation, but you cannot modify host names with it.

5. Start the `omniNames` service and all FSE processes using the `fse --start` command.

Note that you must first start the processes on a consolidated FSE implementation or on the FSE server, and then start the processes on external FSE clients.



Caution: In the following procedures for configuring a LAN connection, if you reconfigure a system with several network adapters enabled, you must configure the `omniORB.cfg` file as described in the procedures for configuring private network communications, instead of renaming the file.

In this case, the parameters you specify in `omniORB.cfg` must be verified against the actual LAN configuration for that system.

Reconfiguring Communication on a Consolidated FSE Implementation or on an FSE Server

No external FSE clients or ordinary LAN connection

This section describes the configuration procedure for the following situations:

- if your FSE installation consists of a consolidated FSE implementation and you do not plan to connect external FSE clients to it, or
 - if you want your external FSE clients to communicate with the consolidated FSE implementation or with the FSE server through an ordinary LAN.
- Ensure that the value of the `server` variable in `services.cfg` equals the fully-qualified domain name (FQDN) that identifies the consolidated FSE implementation or the FSE server inside the LAN.

The following is an example of a properly configured `services.cfg` file in FSE installations using an ordinary LAN. The file can also include the `hostname` variable, but this variable is redundant in such a set-up.

```
server = fse-server.fsecompany.com
```

- Rename the `omniORB.cfg` file (for example, to `omniORB.bak`) on the consolidated FSE implementation or the FSE server so that you will be able to retrieve it later, if required.

Private network connection

If your external (distributed) FSE clients will use a private network for communicating with the FSE server, make the following modifications on the FSE server:

- Add the `hostname` variable (if not present) to `services.cfg` and set its value to the fully-qualified domain name (FQDN) that identifies the system inside the private network.

The following is an example of a properly configured `services.cfg` file in FSE installations using a private network. The `server` variable is redundant in such a set-up.

```
hostname = fseserver.fsenet
server = fse-server.fsecompany.com
```

- Rename the `omniORB.cfg` backup file (for example, `omniORB.bak`) back to `omniORB.cfg`. Modify the following parameters in the file:

- the FQDN that identifies the system inside the private network,
- the IP address of the system,
- the subnet mask.

All these parameters must be verified against the actual private network configuration. Ensure that the FQDN you specify in `omniORB.cfg` matches the FQDN specified for the `hostname` variable in the `services.cfg` file.

The following is an example of a properly configured `omniORB.cfg` file:

```
# Which interface omniORB uses for IORs

endPoint = giop:tcp:fserver.fsenet:

# The order of network interfaces to use for accepting
connections:
# Only localhost and private network. Others are denied.

clientTransportRule = localhost tcp
clientTransportRule = 192.168.240.0/255.255.255.0 tcp
clientTransportRule = * none

# The order of network interfaces to use for opening new
connections:
# Only localhost and private network. Others are denied.

serverTransportRule = localhosttcp
serverTransportRule = 192.168.240.0/255.255.255.0 tcp
serverTransportRule = * none
```

Reconfiguring Communication on External Linux FSE Clients

Ordinary LAN connection

If the external Linux FSE clients and the consolidated FSE implementation or the FSE server will communicate through an ordinary LAN, you have to modify the `services.cfg` file on each external Linux FSE client, as follows:

- Ensure that the value of the `server` variable in `services.cfg` equals the fully-qualified domain name (FQDN) that identifies the consolidated FSE implementation or the FSE server inside the LAN.

The following is an example of a properly configured `services.cfg` file in FSE installations using an ordinary LAN. The file can also include the `hostname` variable, but this variable is redundant in such a set-up.

```
server = fse-server.fsecompany.com
```

- Rename the `omniORB.cfg` file (for example, to `omniORB.bak`) on the consolidated FSE implementation or the FSE server so that you will be able to retrieve it later if required.

Private network connection

If the external (distributed) Linux FSE clients will communicate through a private network to the FSE server, you must modify both configuration files, `services.cfg` and `omniORB.cfg`, on each external Linux client, as follows:

- Modify the value of the `server` variable in `services.cfg` to contain the fully-qualified domain name (FQDN) that identifies the consolidated FSE implementation or the FSE server inside the private network.
- Add the `hostname` variable (if not present) to `services.cfg` and set its value to the FQDN that identifies the Linux FSE client inside the private network.

The following is an example of a properly configured `services.cfg` file in FSE installations using a private network:

```
hostname = fse-lnx-client.fsenet
server = fseserver.fsenet
```

- Rename the `omniORB.cfg` backup file (for example, `omniORB.bak`) back to `omniORB.cfg`. Modify the following parameters in the file:
 - the FQDN that identifies the consolidated FSE implementation or the FSE server inside the private network,
 - the IP address of the system,
 - the subnet mask.

All these parameters must be verified against the actual private network configuration. Ensure that the FQDN you specify in `omniORB.cfg` matches the FQDN specified for the `server` variable in the `services.cfg` file.

The following is an example of a properly configured `omniORB.cfg` file:

```
# Which interface omniORB uses for IORs

endPoint = giop:tcp:fseserver.fsenet:

# The order of network interfaces to use for accepting
connections:
# Only localhost and private network. Others are denied.

clientTransportRule = localhost tcp
```

```
clientTransportRule = 192.168.240.0/255.255.255.0 tcp
clientTransportRule = * none

# The order of network interfaces to use for opening new
connections:
# Only localhost and private network. Others are denied.

serverTransportRule = localhosttcp
serverTransportRule = 192.168.240.0/255.255.255.0 tcp
serverTransportRule = * none
```

Reconfiguring Communication on External Windows FSE Clients

Ordinary LAN connection

If the external Windows FSE clients and the consolidated FSE implementation or the FSE server will communicate through an ordinary LAN, you have to modify the `services.cfg` file on each external Linux FSE client, as follows:

- Ensure that the value of the `server` variable in `services.cfg` equals the fully-qualified domain name (FQDN) that identifies the consolidated FSE implementation or the FSE server inside the LAN.

The following is an example of a properly configured `services.cfg` file in FSE installations using an ordinary LAN. The file can also include the `hostname` variable, but this variable is redundant in such a set-up.

```
server = fse-server.fsecompany.com
```

- Rename the `omniORB.cfg` file (for example, to `omniORB.bak`) on the consolidated FSE implementation or the FSE server so that you will be able to retrieve it later if required.

Private network connection

If the external Windows FSE clients and the consolidated FSE implementation or the FSE server will communicate through a private network, you must modify both configuration files, `services.cfg` and `omniORB.cfg`, on each external Windows client, as follows:

- Modify the value of the `server` variable in `services.cfg` to contain the fully-qualified domain name (FQDN) that identifies the consolidated FSE implementation or the FSE server inside the private network.

- Add the *hostname* variable (if not present) to *services.cfg* and set its value to the FQDN that identifies the Windows FSE client inside the private network.

The following is an example of a properly configured *services.cfg* file in FSE installations using a private network:

```
hostname = fse-win-client.fsenet
server = fseserver.fsenet
```

- Rename the *omniORB.cfg* backup file (for example, *omniORB.bak*) back to *omniORB.cfg*. Modify the following parameters in the file:
 - the FQDN that identifies the consolidated FSE implementation or the FSE server inside the private network,
 - the IP address of the system,
 - the subnet mask.

All these parameters must be verified against the actual private network configuration. Ensure that the FQDN you specify in *omniORB.cfg* matches the FQDN specified for the *server* variable in the *services.cfg* file.

The following is an example of a properly configured *omniORB.cfg* file:

```
# Which interface omniORB uses for IORs

endPoint = giop:tcp:fseserver.fsenet:

# The order of network interfaces to use for accepting
connections:
# Only localhost and private network. Others are denied.

clientTransportRule = localhost tcp
clientTransportRule = 192.168.240.0/255.255.255.0 tcp
clientTransportRule = * none

# The order of network interfaces to use for opening new
connections:
# Only localhost and private network. Others are denied.

serverTransportRule = localhosttcp
serverTransportRule = 192.168.240.0/255.255.255.0 tcp
serverTransportRule = * none
```

Managing Media

3

The way you manage FSE media can significantly increase the stability and performance of your FSE installation. Managing media consists of tasks that are related to the media life cycle, from formatting, initialization and optional duplication for vaulting purposes, to replacing worn-out media and media reorganization.

FSE supports several media types, which are grouped into appropriate media pools. These media pools are created by the FSE administrator according to their purpose, for example, a backup pool for media is used only for backup and holds only one type of media.

All media must be formatted and initialized before they can be used for data storage by FSE. Then you are able to track media pools and media, review their characteristics and statuses, make exact copies of your media, or decide on multiple copies made at once. For specifics regarding disk media, see [“Introducing Disk Media”](#) on page 79.

This chapter includes the following topics:

- [FSE Media Pools](#), page 69
- [FSE Media](#), page 76

FSE Media Pools

An FSE media pool is common storage for the migrated HSM file system data. It groups media of the same type into a logical unit, which is tracked as a group. All media in the same FSE media pool share such characteristics as number of medium volumes, availability of system volumes, capacity of medium volumes, and so on (see [“FSE Media”](#) on page 76 for more information on media types and characteristics). Media in the same media pool can be located in multiple FSE tape libraries or disk systems.

A media pool is related to one FSE partition only and cannot be assigned to more than one partition. However, an FSE partition can have more than one media pool assigned to it, but the assigned media pools must be either regular or WORM. The number of media pools assigned to the FSE partition determines the number of copies to be made. For more information on multiple copying, see [“Multiple Copying”](#) on page 144.

The procedures in this section provide instructions for:

- [Media Pool Types and Their Characteristics](#), page 70
- [Configuring a Media Pool](#), page 71
- [Determining the Configured Media Pools](#), page 74
- [Checking the Status of a Particular Media Pool](#), page 76

Media Pool Types and Their Characteristics

The media pool type is defined in the media pool configuration file. Configuration templates are located on:

- Linux: `/opt/fse/newconfig/`
- Windows: `%InstallPath%\newconfig`

For more information on editing templates, see [“Using FSE Commands for Configuring Resources and Policies”](#) on page 41. Examples of media pool configuration are listed in the next section, [“Configuring a Media Pool”](#) on page 71.

■ Regular media pool

This media pool type is the most common one since it is used for regular archiving of the migrated FSE installation data. It contains FSE media that can be overwritten, reorganized, reinitialized, and reformatted.

■ WORM media pool

Used for WORM media only. On these media, data can be recorded only once, therefore FSE WORM media pool type is particularly suitable for storing permanent archives. It is used to migrate files stored in an FSE WORM file system accordingly.

■ Backup media pool

Used for backup media only. A backup media pool is used differently to other media pools; since backup media do not store user data migrated from an HSM file system, but are only meant for storing backup copies of the FSE installation, there is only one backup FSE media pool configured in FSE.

A backup media pool must be configured with only one volume, and backup media in such a pool should not be initialized. For details about FSE backup, see “[Backup, Restore, and Recovery](#)” on page 191.

Configuring a Media Pool

A configured media pool is a prerequisite for the partition configuration. Each FSE partition is located on a separate HSM file system and must have at least one configured media pool.

Depending on the data characteristics of each partition, media type which you want to use for archiving such data as well as the archiving purpose (regular archiving, backup, or WORM for permanent data archiving) you need to create the appropriate media pool type.

Note: The maximum number of medium volumes per disk medium is 255. When configuring a disk media pool, you need to specify the `VolumeSize` parameter large enough that $255 * VolumeSize$ is larger than the complete disk medium. Otherwise a warning is issued and the number of volumes is set to 255, leaving some unused space.

To configure a media pool, select the appropriate template, modify it and then apply it to configure a new FSE media pool.

1. Locate the wanted template in the `newconfig` directory, for example `fse-pool-AIT.cfg`, and copy it to the selected directory. On Linux, templates are located in the `/opt/fse/newconfig` directory, while on Windows they are in `%InstallPath%\newconfig`. The example shows how you copy the template `fse-pool-AIT.cfg` to the `MyConfiguration` directory.

■ **Linux:**

```
# cd MyConfiguration
# cp /opt/fse/newconfig/fse-pool-AIT.cfg /
    MediaPoolCfgFile
```

■ **Windows:**

```
D:\FSE> cd MyConfiguration
D:\FSE\MyConfiguration> copy "C:\Program Files\
    Hewlett-Packard\FSE\newconfig\fse-pool-AIT.cfg"
    MediaPoolCfgFile
```

2. Set the proper values of the FSE media partitioning parameters in the media pool configuration file. Check that all other parameters match the characteristics of your media.

Windows specific: Note that some Windows host bus adapter (HBA) drivers (particularly Adaptec) do not support large block sizes by default. You can increase the transfer size by adding the `MaximumSGList` value to `DWORD` registry setting. See [“Media Initialization Fails with Large Block Size”](#) on page 222 for details.

Check the following examples for the specifics according to different pool types, and then continue applying the modified templates.

Example of a regular AIT media pool configuration file

```
[FSE Pool]
[Version = 1]
[Date = "2003/04/10 15:00:12"]
[User = "MaxMeier@MyCompany.com"]

Name                = "My_AIT_Pool"

MediaFamily         = AIT
PoolType            = Regular

[DefaultFormat]
BlockSize           = 32KB
NumOfVolumes        = 9
SysVolLocation      = MIDDLE
SizeOfSysVol        = 512MB
```

Example of an AIT-WORM media pool configuration file

```
[FSE Pool]
[Version = 1]
[Date = "2003/04/10 15:00:12"]
[User = "MaxMeier@MyCompany.com"]

Name                = "My_WORM_Pool"

MediaFamily         = AIT
PoolType            = WORM

[DefaultFormat]
BlockSize           = 32KB
NumOfVolumes        = 9
```

```
SysVolLocation = MIDDLE
SizeOfSysVol   = 512MB
```

Example of an SAIT media pool configuration file

```
[FSE Pool]
[Version = 1]
[Date = "2003/04/10 15:00:12"]
[User = "MaxMeier@MyCompany.com"]

Name           = "My_SAIT_Pool"
MediaFamily    = SAIT
PoolType       = Regular

[DefaultFormat]
BlockSize      = 32KB
NumOfVolumes   = 1
SysVolLocation = NONE
SizeOfSysVol   = 0MB
```

Example of an LTO media pool configuration file

```
[FSE Pool]
[Version = 1]
[Date = "2003/04/10 15:00:12"]
[User = "MaxMeier@MyCompany.com"]

Name           = "My_LTO_Pool"
MediaFamily    = LTO
PoolType       = Regular

[DefaultFormat]
BlockSize      = 32KB
NumOfVolumes   = 1
SysVolLocation = NONE
SizeOfSysVol   = 0MB
```

Example of a disk media pool configuration file

```
[FSE Pool]
[Version = 1]
[Date = "2003/04/10 15:00:12"]
[User = "MaxMeier@MyCompany.com"]
```

```

Name           = "My_DM_pool"

MediaFamily    = DISK
PoolType       = Regular

[DefaultFormat]

BlockSize      = 32KB
VolumeSize     = 100GB
    
```

3. To configure a new media pool, apply the modified template:

```
# fsepool --add MediaPoolCfgFile
```

You have configured a new FSE media pool. Its configuration is stored in the Configuration Database as the first respective configuration revision. For more information, see “[Configuration Database](#)” on page 37.

Once the media pool is configured, you need to add media that you want to use for FSE operation to this pool, as described in “[Adding Media to an FSE Media Pool](#)” on page 83.

Determining the Configured Media Pools

You can determine the attributes of all configured media pools in the FSE installation by running the following command:

```
# fsepool --list [--detail]
```

The command output lists the names and attributes of all currently configured FSE media pools in the FSE installation, as follows:

- the media pool family and type,
- the owning partition’s name,
- the medium block size,
- the number of volumes on the medium,
- the data volume size,
- the location of an optional system volume on the medium,
- the system volume size.

You can specify most of these attributes in the media pool configuration file.

If you use the `--detail` option, `fsepool` formats the output in a different way, and displays additional FSE media pool attributes, as follows:

- the UUID of the pool,
- the UUID of the owning partition,
- the media pool's RMDB entry key.

These attributes are assigned automatically by FSE during the FSE media pool (or FSE partition) configuration.

Excerpt from the `fsepool --list` output

Pool	Family	Type	Partition	BlkSize[KB]	#Vol	VSize[MB]	SysVol	SysVol[MB]
my_AIT	AIT	Regular	partAIT	128	17	n/a	First	512
my_Disk	DISK	Regular	n/a	64	1	100	None	n/a

Excerpt from the `fsepool --list --detail` output

```

Pool:          my_AIT
UUID:         c8898640-8f8a-44aa-a6fd-47687ac1f791
Family:       AIT
Type:         Regular
Partition:    partAIT
PartUUID:
BlkSize[KB]: 128
#Vol:         17
VSize[MB]:    n/a
SysVol:       First
SysVol[MB]:   512
Key:          1

Pool:          my_Disk
UUID:         f264a661-1076-41dc-baab-bae7aed9e15b
Family:       DISK
Type:         Regular
Partition:    n/a
PartUUID:
BlkSize[KB]: 64
#Vol:         1
VSize[MB]:    100
SysVol:       None

```

```
SysVol[MB]:  n/a  
Key:        2
```

Checking the Status of a Particular Media Pool

To check the current status of a particular media pool, you must determine the FSE partition to which the media pool belongs. Run the following command:

```
# fsepool --list [--detail]
```

In the `fsepool` command output, locate the name of the owning partition and supply it as an argument to the `fsepartition` command as follows

```
# fsepartition --status PartitionName
```

The output of this command lists the names of all media pools that are currently assigned to the selected partition, and their current status, as follows:

- the total and available capacity of all data volumes on media belonging to the media pool,
- the percentage of data volume usage,
- the total and available capacity of all system volumes on media in the media pool,
- the percentage of system volume usage,
- the numbers of good, unreliable, unusable, and uninitialized media of the media pool,

Excerpt from the `fsepartition --status` output

Pool	Size[MB]	Avail[MB]	Used[%]	SysSize[MB]	SysAvail[MB]
poolAIT	92173	70397	24	7150	7077
SysUsed[%]	#Good	#Unreliable	#Unusable	#Uninit	
2	2	0	0	0	

FSE Media

An FSE installation supports tape media as well as disk media. Each FSE medium has an entry and properties written in the Resource Management Database (RMDB) and is assigned to one FSE media pool according to its type. The FSE Resource Manager identifies the medium with a barcode, which is either acquired automatically from the media library or specified manually when adding media to

the media pool. A medium barcode must be unique—no other medium in any of the libraries managed by a particular FSE installation can have the same barcode, and neither can any disk medium have the same name as an already used barcode.

Once the medium is added to the FSE media pool, it can be prepared for use by the FSE installation. Preparation includes formatting and initializing of the FSE media. Formatting depends on media technology support for multiple volumes. Next step is initialization, which overwrites the volume header and adds the medium volume entry to the Resource Management Database. Both media formatting and initialization are processed by an administrative job and are executed by the FSE Back End Agent.

Disk media emulates tape media to a great extent and can be used for regular archiving of the migrated FSE installation. The advantage of using disk media is accelerated recall of offline files.

The `fsemedium` command is used to monitor and manage FSE media. It is also used to invoke administrative jobs, which perform FSE media formatting and initialization of new or already used media.

The procedures in this section provide instructions on:

- [Adding Media to an FSE Media Pool](#), page 83
- [Formatting Media](#), page 83
- [Initializing Media](#), page 84
- [Duplicating Media](#), page 85
- [Reorganizing Media](#), page 88
- [Recreating Redundant Copies of Migrated Data](#), page 97
- [Checking the Status of Media](#), page 102
- [Changing the Condition Status of Media](#), page 105
- [Closing Medium Volumes](#), page 106
- [Recreating Fast Recovery Information from Media](#), page 106
- [Removing FSE Media](#), page 107

FSE Medium Volumes

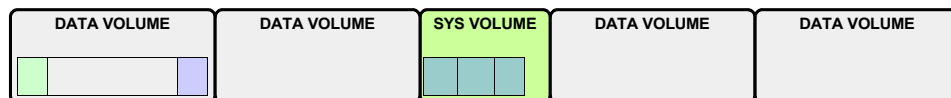
There are two types of medium volumes:

- **Data volume:**

Used to store regular migration data. Each FSE medium must have at least one data volume; usually, an FSE medium is partitioned into several data volumes. SAIT and LTO Ultrium media do not support multiple medium volumes, and they do not need to be formatted.

- **System volume:**

Used to store a copy of Fast Recovery Information (FRI) from other data volumes on the same medium. Each FSE medium can have one system volume. FSE supports system volumes on AIT media only; SAIT, LTO Ultrium, and disk media cannot have system volumes.



Partitioned AIT medium

FSE supports a maximum of 32 volumes on AIT media and 255 volumes on disk media. Other specifics of different media types are described in the next section.

FSE Media Types

The following is a list of the supported media types and their characteristics:

- LTO Ultrium 1, LTO Ultrium 2,

Used in regular and backup media pools.

LTO Ultrium 1 and LTO Ultrium 2 media do not support multiple volumes and do not need to be formatted. A single data volume is created on such media during initialization.

If a medium is used as a backup medium, it must be configured with only one volume, and backup media in such a pool should be formatted but not initialized. For details about backup media, see “[Backup Media Management](#)” on page 197.

- AIT-2 WORM, AIT-3 WORM

Used in WORM media pools only.

Once recorded, AIT-WORM media cannot be overwritten, re-formatted, or re-initialized, but new data can be appended to the existing contents. Note that the FSE system prevents AIT WORM media from being initialized in a non-WORM media pool.

- **Disk media**

Used in regular media pools.

A disk medium is a file system mounted to a subdirectory of `/var/opt/fse/dm/` on Linux or `%InstallPath%\var\dm` on Windows. For example, mounted disk media on Linux are `/var/opt/fse/dm/dm000001` and `/var/opt/fse/dm/dm000002`. The last part of a pathname (`dm000001`, `dm000002`) is used as a ‘barcode’ for identification of disk media. This naming convention prevents disk media interfering with real barcodes.

A file system for disk media can be either local or remote. For more information, see “[Introducing Disk Media](#)” on page 79.

Introducing Disk Media

An FSE disk medium is a mounted file system that emulates an FSE tape medium; therefore, managing disk media is similar to managing tape media. Using FSE disk media is advantageous in terms of recall, since it significantly decreases the time required to bring files back online (offline file access times).

When an FSE user requests an offline file, a recall job first attempts to allocate a medium from the FSE disk media pool and only if no such medium exists it allocates a tape medium. When offline files are located on a disk medium, recalls can run simultaneously and are not affected by the number of currently unallocated drives, but by the corresponding FSE partition configuration parameters. No more than $MaxNumDrivesRecall \times 2$ jobs are started at once. The *MaxNumDrivesRecall* and other configurable FSE partition parameters are described in “[Partition Allocation and Job Priority Policy](#)” on page 139.

Before starting to use disk media, you should consider the following points about the two types of disk in an FSE implementation:

- **Performance disk**

Dedicate a disk with high performance for FSE software installation and HSM file systems. Such a disk serves as a primary storage space of files with high frequency of user access, which may affect the performance of the whole FSE installation. A performance disk is not meant to be a secondary storage space (file archive storage), thus its capacity is not as important as its high performance, which enables optimal operation of FSE.

- Capacity disk

The main purpose of a capacity disk in FSE is its availability as a secondary storage space. A capacity disk should be dedicated to the FSE disk media and the FSE disk buffer. Generally, it is used as a large archive storage, meaning that the frequency of user access to archived files is relatively low compared to that of the HSM file system on the performance disk. The capacity disk is used during migrations and recalls and should have enough space for holding the migrated generations of the HSM file system files.

The available space on the disk media in an FSE disk media pool should correlate to the available space on the tape media in the FSE tape media pool assigned to the same FSE partition. The FSE tape media pool is typically used for storing the first copies. Suppose that the tape media pool contains 20 AIT-3 media (app. 20 * 100 GB of available space); the corresponding disk media pool should have 2 TB of available space.

Note: It is recommended that a separate disk partition is allocated for the FSE disk buffer to improve robustness and to avoid potential data loss.

Local and remote file systems as disk media

FSE supports local file systems to be configured as disk media. On Linux, it is also possible to use remote file systems as disk media and to access them via the NFS or CIFS protocol.

Note: If you are configuring a remote file system as a disk medium, you can mount only a complete file system created for this purpose; mounting shared directories (subtree of a file system) is not supported.

Linux specific

On Linux systems, a file system that will be configured as a disk medium must be mounted to a subdirectory of `/var/opt/fse/dm/`; for example:

```
/var/opt/fse/dm/dm000001, and
```

```
/var/opt/fse/dm/dm000002.
```

Mounting a file system on Linux includes adding the file system entries for the HSM file systems to the `/etc/fstab` file. For automating the mounting, see the man pages for the `fstab` and `mount` commands.

Remote file systems are mounted via the NFS protocol. Note that when using an NFS volume as a disk medium, you must export a file system with the `no_squash_root` parameter to enable “root” to access the file system.

You define this parameter in the `/etc/exports` file on the NFS server for this particular export. It is recommended that you make such an export available only for the FSE server. The following is the example of entry in the `/etc/exports` file:

```
/mnt/disk_medium fse.company.com(rw,no_squash_root)
```

Windows specific

On Windows systems, a file system for a disk medium must be mounted to a subdirectory of `%InstallPath%\var\dm`; for example:

```
C:\Program Files\Hewlett-Packard\FSE\var\dm\dm000001, and
```

```
C:\Program Files\Hewlett-Packard\FSE\var\dm\dm000002.
```

To configure a mount point for a file system that will be used as a disk medium, use the Disk Management interface (**Start > Settings > Control Panel > Administrative Tools > Computer Management > Disk Management**).

Disk media specifics

The main difference between tape and disk media is in preparing media for use by FSE. A disk medium must be mounted to a `dm<number>` mount point in the `/var/opt/fse/dm/` directory on Linux, or in `%InstallPath%\var\dm` on Windows, before it can be used. Afterwards it also needs to be formatted and initialized, but you must be aware of the distinctions between the different media types:

- Disk media can only be added to an FSE disk media pool.
- Disk media are formatted and initialized similarly to tape media. Formatting of disk media creates subdirectories for volumes. Each volume has its own directory named after the corresponding volume number, for example:

```
/var/opt/fse/dm/dm000001/00000001. The volume directory includes the following files:
```

- medium volume header (`00000000.hdr`)
- medium volume data (`00000003.vdf`)
- medium volume FRI data (`<FileBlockOffset>.fri`)
- medium volume size identifier (`vol.info`)

- The maximum number of medium volumes per medium is 255. You define the size of medium volumes in the media pool configuration file.

Note: When configuring a disk media pool, you need to specify a *VolumeSize* parameter large enough that $255 * VolumeSize$ is larger than the total disk medium size. Otherwise, a warning is issued and the number of volumes is set to 255 (FSE can only index 255 volumes), leaving some unused space.

- At initialization, the size of medium volumes is written to the Resource Management Database and to the volume. Disk medium volumes are closed when the sum of all files on the particular volume reaches the volume size that was specified in the disk media pool configuration.
- The barcode of a disk medium is the relative part of the mount point, for example `dm000001`, `dm000002`. Its length is limited to 64 characters.

Example layout of disk media

```

|-- dm
|  `-- dm000001      medium barcode
|     |-- 00000001   volume number
|     |  |-- 00000001.hdr volume header
|     |  `-- 00000003.vdf volume data file
|     |-- 00000002
|     |  |-- 00000001.hdr
|     |  |-- 00000003.vdf
|     |  `-- 00028545.fri fast recovery information file
|     `-- 00000003
|        `-- 00000001.hdr volume contains no data

```

- Once a disk medium is formatted, it is not possible to add additional medium volumes to the medium. When running out of space, another disk medium must be added to the disk media pool.
- Data compression is not supported for disk media.
- Location information, which you get by running the `fsemedium --list` command, is not valid for disk media. See details in “[Checking the Status of Media](#)” on page 102.

Adding Media to an FSE Media Pool

By adding media to an FSE media pool, you prepare them so they can be used by FSE. Note that the parameters specified in an FSE media pool configuration determine the characteristics for all media added to this media pool, such as the number of medium volumes and the system volume size and location. See “[FSE Medium Volumes](#)” on page 78 for a description of data and system volumes and “[Introducing Disk Media](#)” on page 79 for disk media specifics.

1. Create an entry for an FSE medium and add it to a configured FSE media pool using the `fsemedium` command, specifying the medium barcode:

```
# fsemedium --add Barcode --pool PoolName
```

Repeat this step for all media that you plan to add to the FSE installation at this time.

2. If you want to add non-disk FSE media, physically place all newly-added FSE media into free slots in the corresponding library. Trigger a physical library inventory update (medium barcode rescan) using the following command:

```
# fselibrary --update-inventory LibraryName --rescan
```

Once the media are added to the FSE media pool, you must format and initialize them, which prepares them for use as the secondary storage space in FSE.

Formatting Media

Formatting a medium prepares it for use by the FSE installation by partitioning it into FSE medium volumes. The number of medium volumes that can be created by formatting depends on the media technology and the limits of FSE. Partitioning is done according to the parameters specified in the corresponding FSE media pool configuration.

If a medium is already properly formatted (partitioned), you can skip this step and proceed with initialization. Note that LTO and SAIT media cannot be formatted.

Format the medium using the following command:

```
# fsemedium --format Barcode
```



Caution: Formatting erases all data on the medium.

In case the medium has already been formatted and used by FSE, you need to use the additional `--force` option:

```
# fsemedium --format Barcode --force
```

Note: `--force` will allow the formatting of an FSE medium only if there is no valid FSE data stored on it (that is, the percentage used of all medium volumes is zero).

Once the medium is formatted, you must initialize the FSE media volumes created on it.

Initializing Media

Initializing an FSE medium volume prepares the volume for use in the FSE installation by labeling its volume header and adding the medium volume entry to the Resource Management Database.

Note: Backup media should not be initialized. In a backup operation, FSE rejects initialized media to prevent unintentional overwriting of FSE media.

For more information on backup media specifics, see “[Backup Media Management](#)” on page 197.

The following table can help you determine what kind of initialization is allowed for different medium volume types recognised by FSE.

Medium volume type	Initialization	
	Regular (<code>--init</code>)	Force (<code>--init --force</code>)
Blank (new)	YES	YES
Unknown (contains data of unknown format)	YES	YES
Tar volume	NO	YES
ECMA/ANSI volume	NO	YES
FSE volume (but no valid FSE data stored on volume)	NO	YES
FSE volume with valid data (volume assigned to any FSE media pool in use and still contains data (% used > 0))	NO	NO

For regular initialization use the following command:

```
# fsemedium --init Barcode
```

For forced initialization use the following command:

```
# fsemedium --init Barcode --force
```

Note: You can execute both formatting and initializing at once with the following command:

```
# fsemedium --format Barcode --init
```

Duplicating Media

The FSE installation provides an option to have multiple copies of the migrated data made automatically during the migration job. You do this by configuring several FSE media pools for one FSE partition. The number of media pools assigned to the same partition corresponds to the number of copies the files will have on FSE media. For more information, see “[Multiple Copying](#)” on page 144.

Multiple copying is favorable with regards to recall, since you are able to recall your data from any available copy. All media pools that are assigned to the same partition contain the same set of data, however, it is not guaranteed that an individual medium from a certain media pool has a corresponding medium with exactly the same data in another media pool. If you want to create an exact copy of an FSE medium for eventual replacement of the original, you must duplicate the medium.

Medium duplication is usually used for vaulting purposes, where media copies are put off-site for safety reasons. It can also be useful in situations where a particular medium becomes worn out and the data on it is still valuable.

Note: Medium duplication is not available for disk media.

A duplicated medium is an exact copy of the original medium. An original medium and its copy have the following characteristics:

- same medium volume headers, including medium volume UUIDs
- same Fast Recovery Information data, on system volume and all data volumes
- same data
- same media types and capacities
- different medium barcodes

How it works

An admin job is started to perform medium duplication. The medium duplication process first requests disk buffer resources, allocates the original medium and then the target medium. If the prerequisites are met, medium duplication is started volume by volume. Once finished, the medium copy has the same attributes and data as the original except for the medium barcode.

Note: To replace an original medium with its duplicate, you should label the duplicate medium with the original barcode, remove the original medium, and use the duplicate instead.

When a medium copy is first used in place of the original, a warning is written to the `fse.log` file as the FSE installation recognizes that the medium for the particular barcode has been switched.

Typically, when you decide to duplicate a medium, for example for storing backup copies off-site, the original medium is full and all its medium volumes are closed. If the medium is not yet full and still has open volumes that are almost full, it is recommended that you close all such volumes first (as described in “[Closing Medium Volumes](#)” on page 106), and only then start duplicating. This way you will not have to repeat the duplicating procedure each time new data is migrated to the original medium.

If you are duplicating a medium that still has open volumes, the medium volumes are copied as is and are left open on the target medium. Therefore, you have to repeat the duplication procedure every time new data is migrated to the original medium in order to have an up-to-date medium copy. In such cases, you can reuse the same target medium as described in “[Procedure for duplicating media](#)” on page 86.

Note that while medium duplication is in progress, no other job (migration or recall) can use the disk buffer space that was allocated. The migration jobs are resumed when disk buffer space is freed, while recall jobs fail after the specified timeout.

Procedure for duplicating media

Before you start duplicating an FSE medium, you must ensure that the duplication prerequisites are met. Note that the most common reasons for medium duplication failures are lack of disk buffer space and mismatches in the medium characteristics.

Prerequisites

- A target medium, which will hold a copy of the original data, must be of the same type as the original medium.
- The target medium must have sufficient capacity.
- The target medium should not be initialized or should at least not contain any valid data or a valid header such as a `tar` archive label or an ECMA / ANSI label.
- The target medium should not belong to any FSE media pool that is already assigned to an FSE partition.
- There must be enough free disk space for the FSE disk buffer. The minimal required space is the native FSE medium capacity multiplied by 2, as shown in the following examples:

```
AIT-2 : 50 GB * 2 = 100 GB
AIT-3 : 100 GB * 2 = 200 GB
LTO 1 : 100 GB * 2 = 200 GB
LTO 2 : 200 GB * 2 = 400 GB
SAIT-1: 500 GB * 2 = 1 TB
```

When the prerequisites are met, and the original and target media are prepared, you can make a duplicate of the original medium using the following command:

```
# fsemedium --duplicate SourceBarcode TargetBarcode
```

Note: If medium duplication fails because the target medium was already initialized (partitioned to volumes) or holds valid data, you can use the `fsemedium --format Barcode --force` command to reformat the target medium. Forced duplication overwrites existing data to enable duplicated data to be written to the medium.

You can perform a forced medium duplication using the following command:

```
# fsemedium --duplicate SourceBarcode TargetBarcode
[--no-monitor] --force
```

If you do not want to monitor the medium duplication progress, specify the `--no-monitor` option as shown in the command syntax above. Otherwise, the duplication progress will be reported to the standard output.

Example output for medium duplication performed on a single drive, with monitoring enabled

```
/root>fsemedium --duplicate 123123 456456
```

```
[11:42:09] Started.
[11:42:09] Requesting resources...

[11:42:09] Resources allocated.
[11:42:09] Medium 123123 is already present in drive drv1.
[11:42:09] Starting Back End Agent.
[11:42:10] Medium 123123: 2 volumes found.
[11:42:10] Medium 123123, volume 1: Reading volume
identification...
[11:43:48] Medium 123123, volume 1: Reading data.
[11:43:58] Medium 123123, volume 1: Volume read and written to
disk buffer.
[11:44:10] Medium 123123, volume 2: Reading volume
identification...
[11:44:48] Medium 123123, volume 2: Reading data.
[11:45:58] Medium 123123, volume 2: Volume read and written to
disk buffer.
[11:43:58] Reading finished, releasing resources.
[11:42:09] Requesting resources...
[11:42:09] Resources allocated.
[11:48:42] Loading medium 456456 to drive drv1.
[11:43:58] Checking medium format.
[11:43:58] Blank medium detected.
[11:43:48] Medium 456456: Blank volume detected.
[11:43:58] Medium 456456, volume 1: Writing data...
[11:43:58] Medium 456456, volume 1: Volume written.
[11:43:58] Medium 456456, volume 2: Writing data...
[11:43:58] Medium 456456, volume 2: Volume written.
[11:43:58] Writing finished, releasing resources.
[11:43:58] Finished.
```

Reorganizing Media

In FSE, data stored on an HSM file system is automatically and periodically migrated to FSE media. This applies not only to every new file created on the HSM file system, but also to every existing file whose attributes or contents is modified. Each modification of the file subsequently causes it to be migrated to media as a new file generation.

Over time, the situation can arise where a great deal of space on FSE media is occupied by old, obsolete generations of files. FSE enables you to reclaim media space by reorganizing the data on the media. The media reorganization process frees space on media by preserving only important file generations and removing the others. FSE media space can then be reused for newly migrated data.

Note: AIT-2 WORM and AIT-3 WORM media cannot be rewritten or reformatted, but they can be reorganized. However, the media space reclaimed cannot be reused.

Although you cannot reuse WORM media space, the reorganization of WORM media is necessary when you need to destroy sensitive data that was migrated to these media, while preserving the rest of the data. In these circumstances, the last step of the data erasure process is the physical destruction of the WORM media that holds the sensitive data.

Media reorganization is a medium-volume-based process. This means that the smallest unit of the secondary storage space that can be reorganized is a single FSE medium volume. Similarly, the smallest unit that can be reused is an FSE medium volume. As system volumes do not contain file data, only data volumes can be reorganized.

Limitation

- After the reorganization of one or more FSE medium volumes on an FSE medium, any duplicates of the medium created before the reorganization are rendered useless, as (part of) the source medium has been reinitialized at the end of the reorganization job.

To recreate the data safety level, you must duplicate each medium containing reorganized medium volumes that was the source for a previously duplicated medium.

The media reorganization process

The media reorganization process consists of two steps: scanning and copying. Each step is controlled and performed by its own FSE job.

The scanning step is run for all FSE media in all FSE media pools that are assigned to a specified FSE partition. This step determines the locations of important and obsolete data on the already closed (full) and still open FSE medium volumes, according to parameters that you define. It then records the scan results to the Resource Management Database (RMDb). The scanning process is performed as a search through the corresponding File System Catalog (FSC), therefore the FSE media themselves are not physically scanned.

The results of the scanning step are used in the copying step. This step copies all important file generations from the medium volumes being reorganized to open and empty medium volumes in the same FSE media pool. The original volumes are then recycled and finally reinitialized. To reduce the number of drive load and unload operations during the copying step, the data is copied to (open or empty) medium volumes on the same medium, if any exist.

Note that the copying step of the reorganization process is not started until the scanning step has completed, and vice versa.

Before running the copying step, you can preview the results of the scan to decide which FSE medium volumes or FSE media you actually want to reorganize. It is only useful to reorganize volumes with a high enough proportion of obsolete data (that is, a high enough percentage of FSE slack space).

During execution of the reorganization process, FSE progressively shows status indicators (such as “scanned”, “in reorg”, and “recycled”) to the medium volumes being reorganized. These indicators help you detect and track the reorganization progress for a particular medium volume. You can display these status indicators at any time with the following command:

```
fsemedium --list [Barcode] --volume
```

The following table depicts the reorganization steps, a description of their tasks, the volume status indicators that are set, and the corresponding FSE jobs.

Reorganization step	Step tasks	Status of medium volumes	Job Type
Step 1: Scanning 1. Change the status of the FSE partition to <code>in scan</code> . 2. Scan medium volumes for obsolete data. 3. Write the scan results to the RMDB.	[open/full]	FSE reorganizational scan job
	4. Change the status of the scanned FSE partition back to [enabled/mounted].	[open/full], scanned	

Media reorganization parameters

When triggering the FSE reorganizational scan job, you can specify two parameters that define which migrated file generations are treated as important during scanning and thus preserved during copying. The parameters are in the form of arguments to the `fsepartition` command, and are the following:

Reorganization parameter	Argument to <code>fsepartition</code>
boundary migration date and time (mandatory)	"YYYY/MM/DD HH:mm:ss"
number of latest generations (optional)	<i>NumberOfGenerations</i>



Caution: With either parameter, only migrated generations containing file contents count as file generations. Migrated generations that contain only file headers (name, standard and extended attributes, ACLs, ...) or changes to a file alternate data stream do not count as migrated file generations. Generally, these generations are not preserved during media reorganization.

The boundary migration date and time parameter

If only this parameter is specified, all migrated file data generations created after the boundary migration date and time parameter are considered as important.

The number of latest generations parameter

This optional parameter limits the number of generations that would otherwise be considered as important by the boundary migration date and time parameter alone.

If the parameter value is smaller than the number of important generations determined by the boundary migration date and time parameter, then the set of important generations is reduced to the number of latest generations value. Otherwise, the parameter has no effect.

Exceptions to the reorganizational rule

There are two exceptions to the rule defined by the reorganization parameters:

- The newest migrated generation is always preserved, even if it contains only file header changes or changes to a file alternate data stream.
- If no migrated generation (with file contents) is covered by the specified parameters, the first older generation with file contents is preserved.

Handling deleted files

Files that were deleted from the HSM file system before the reorganizational scan job (either during automatic deletion or manually) are handled in a special way. In certain situations, the exceptions above are not valid for these files.



Caution: Files that had been deleted from the HSM file system and whose last “empty” generations are not covered by the *boundary migration date and time* or the *number of latest generations* parameter are not preserved during media reorganization. All migrated generations of such files are permanently lost.

Scanning media for obsolete data

You can run the scanning step of the reorganization process on all media belonging to a particular FSE partition, using the following command:

```
fsepartition --reorg-scan PartitionName "YYYY/MM/DD HH:mm:ss"
[NumberOfGenerations]
```

Note: Note that you can omit the time part (HH:mm:ss) of the ExpirationDate argument ("YYYY/MM/DD HH:mm:ss"). The time 00:00:00 is assumed.

During the execution of the job, `fsepartition` reports the job execution progress to the standard output, allowing you to monitor the scanning process in the command shell.

Example output of the `fsepartition --reorg-scan` command

```
# fsepartition --reorg-scan my_part "2003/11/26 15:25:13" 3
[11:42:09] Started (JobID: 20040121000001).
[11:42:10] Scanning FSC for reorganization info.
[11:42:10] Scan Parameters: "2003/11/26 15:25:13", generations:
3.
[11:55:22] Scanning complete. Scanned 33.005.664 file generation
copies.
[11:55:23] Use 'fsepartition --reorg-stat' to see the details.
[11:55:25] Finished.
```

Checking the resulting statistics of a reorganizational scan job

After the scan job is completed, you can check the statistics gathered during the scanning process. You can display both statistics related to individual FSE media pools and statistics related to individual FSE media assigned to the specified FSE partition.

Displaying the FSE media pool-based statistics

To check the statistics for all configured FSE media pools:

1. ensure that all scan jobs have completed.
2. Run

```
# fsepool --reorg-stat
```

Example output:

Name	NumOfMedia	ValidData[MB]	Total[MB]	SlackSpace[%]
Pool001	5	9540	25412	62%
Pool002	3	3000	10000	70%
Pool002	1	1000	1000	0%
PoolLTO	10	0	3610	0%

where:

NumOfMedia Number of FSE media assigned to the media pool.

ValidData Amount of space on the FSE media, assigned to the media pool, occupied by important file generations.

If the FSE media pool has not yet been scanned for reorganization, it has 0 (the zero value) displayed in place of the `ValidData` amount.

Total Total amount of space on the FSE media assigned to the media pool.

SlackSpace Percentage of FSE slack space.

FSE slack space is the space on FSE media occupied by migrated file generations considered obsolete by the reorganizational scan job.

Displaying the FSE media-based statistics

To check the statistics for all FSE media that belong to a particular FSE partition:

1. Ensure that the scan job running on the chosen partition has completed.
2. Run:

```
# fsepartition --reorg-stat PartitionName [SlackThreshold%]
```


You can also specify a value for the optional *SlackThreshold%* argument to filter out media for which the percentage of important data is not high enough, and thus less likely to need reorganizing.

Example output:

Barcode	VolNum	ValidData[MB]	Total[MB]	SlackSpace[%]
Med001	1	980	1200	19%
Med001	2	190	1000	81%
Med001	3	0	1000	100%
Med001	4	100	1000	90%

where:

VolNum	Number of the FSE media volume on the medium.
ValidData	Amount of space on the FSE medium occupied by important file generations. If the FSE medium has not yet been scanned for reorganization, it has 0 (zero) displayed in place of the ValidData amount.
Total	Total amount of space on the FSE medium.
SlackSpace	Percentage of FSE slack space.

FSE slack space is the space on the medium occupied by migrated file generations that were considered obsolete by the reorganizational scan job.

Running the media reorganization job

Using the `fsemedium` command, you can run the copying step of the reorganization process on all medium volumes of a specific medium, on a single FSE medium volume, or on a series of specified FSE medium volumes.

By default, the command reports the progress of the reorganization job to the command shell for monitoring purposes. You can disable monitoring by specifying the `--no-monitor` option.

Note: To prevent the copying step of the FSE reorganization job from copying the data to the same medium (that is, to the medium whose medium volumes are being reorganized), set the condition status of the medium being reorganized to “unreliable” before running the reorganization job. Use the `fsemedium --mark-unreliable` command for this purpose.

Note that FSE media with the condition status “unreliable” are not available for migrations. After the reorganization job has finished, and the reorganized medium volumes have been reinitialized, you must remove the “unreliable” status of the reorganized medium using the `fsemedium --mark-good` command.

The following is the syntax of the `fsemedium` command when used for media reorganization:

```
fsemedium --reorg Barcode [{-V | --volume} VolumeNumber...] \  
    [--no-monitor]
```

With monitoring enabled, the command generates an output similar to the following:

```
[11:42:09] Started (JobID: 20040121000002).  
[11:42:09] Requesting resources...  
[11:42:09] Resources allocated.  
[11:42:09] Medium 050003 is already present in drive drv1.  
[11:42:09] Starting Back End Agent.  
[11:42:09] Medium 050003, volume 1: Reading valid files...  
[12:20:58] Medium 050003, volume 1: Valid files written to disk  
buffer.  
[12:20:58] Requesting resources...  
[12:20:59] Medium 050003, volume 2: Reading valid files...  
[12:47:12] Resources allocated.  
[12:47:12] Loading medium 050010 to drive drv2.  
[12:47:49] Starting Back End Agent.  
[12:47:49] Medium 050010, volume 5: Writing...  
[12:49:58] Medium 050003, volume 2: Valid files written to disk  
buffer.  
[12:49:58] Reading finished, releasing resources.  
[13:00:58] Medium 050010, volume 5: Data written to medium  
volume.  
[13:01:58] Medium 050010, volume 6: Writing...  
[13:10:50] Medium 050010, volume 6: Data written to medium  
volume.  
[13:11:50] Writing finished, releasing resources.  
[13:12:38] Finished.
```

If you disabled monitoring, you can still check the status of the FSE reorganization job using the `fsejob --list` command. Alternatively, you can track the reorganizational progress in detail by inspecting the FSE event log, named `fse.log` and located in `/var/opt/fse/log/` (on Linux systems) or in `%InstallPath%\var\log` (on Windows systems).

Reusing the recycled medium volumes

Space on FSE medium volumes that participated in the reorganization job can be reused for storage of new data. The status of these medium volumes is set to “recycled” by the corresponding Partition Manager. You can retrieve a list of the recycled medium volumes using the following command:

```
# fsemedium --list Barcode --volume
```

An example output is the following:

Barcode	VolNum	Type	Size[MB]	Avail[MB]	Used[%]	Status
052002	1	Data	5949	5592	80	full, scanned, in reorg, recycled
052002	2	Data	5949	5949	60	open, scanned
052002	3	Data	5949	5949	0	empty
052002	4	Data	5949	5949	0	empty
052002	5	System	1019	1019	0	open

In the example, the first medium volume on the FSE medium with barcode *052002* was recycled during the reorganization job and can be reused.

Before recycled medium volumes can be reused, they must be reinitialized. The reinitialization is performed automatically at the end of the FSE reorganization job. All recycled medium volumes are reinitialized in sequence. If the reorganization job has been terminated for some reason before medium volume reinitialization, you have to reinitialize the recycled volumes manually. For more information on initializing FSE medium volumes, see “[fsemedium](#)” on page 302.

Recreating Redundant Copies of Migrated Data

FSE enables you to assign several FSE media pools to the same FSE partition, in order to store multiple copies of the migrated data on FSE media (via the *Pools* variable in the FSE partition configuration file). In such a configuration, each media pool contains one copy of the migrated data from the HSM file system.

Having multiple copies of the migrated data is useful for various reasons, including the possibility of storing a particular copy in a vault and increasing the overall safety of the migrated data. For information on configuring multiple copies, see “[Multiple Copying](#)” on page 144.

Since tape media wear out over time, some can eventually become unusable. Migrated file generations on such media can no longer be recalled. In effect, this means that one copy of the migrated data is lost. FSE enables you to recover the

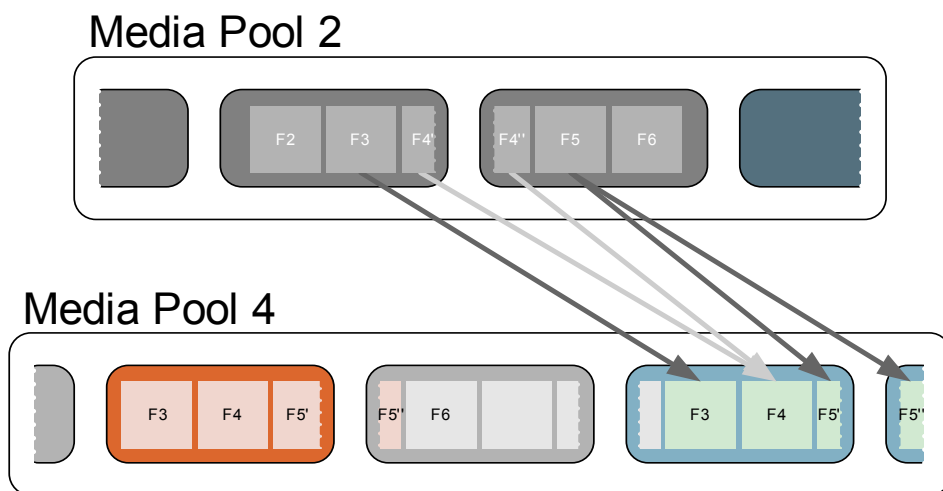
damaged copy, provided that such data was migrated from an FSE partition for which multiple copying was configured. This process is called recreating a redundant copy and is available for FSE tape media and FSE disk media.

By recreating the redundant copy of the damaged migrated data, you can restore the availability and safety of the migrated data.

The redundant copy recreation process

Redundant copy recreation is started by the `fsemedium` command and performed by an FSE maintenance job. The process first scans the File System Catalog (FSC) to find the file generations that were migrated to the specified medium, searches for alternative redundant copies in other FSE media pools, and then remigrates data to the FSE media pool of the unreadable medium. Note that the actual layout of the remigrated data on FSE media differs from the original data layout on the unreadable medium.

Recreating a copy of damaged migrated data



If alternative migrated copies of the unrecalable data exist in both tape and disk media pools, the redundant copy is recreated from the data stored in the disk media pool. By using the disk media pool, as many FSE drives as possible remain available for other FSE processes, such as migration and recall. You can define the total number of drives that can be simultaneously used for redundant copy recreation (for example, when several media are damaged) with the FSE partition configuration variable `SystemMaxNumDrivesMaint`. (This variable affects all types of maintenance jobs.)

The redundant copy recreation process operates on a “best-effort” basis. It processes all file generations that can be copied from an alternative location on FSE media and skips the files for which the copying cannot be done. The process recreates only the generations that were migrated to the unusable medium.

In cases where no alternative migrated copies of the unreadable data exist, a warning (the unreadable data belongs to a non-latest file generation) or an error (the unreadable data belongs to the latest file generation) is reported to the FSE error log for each file that cannot be recreated.

Note: You are not restricted to creating redundant copies only in the case where FSE media is already damaged (“unusable” media status). You can also perform this action with worn-out media (“unreliable” media status). In this way, you can prevent the loss of a migrated data copy before the problem becomes severe. Of course, you can create redundant media copies also while the medium is still readable.

Determining the unusable media

FSE detects read errors when recalling the migrated data from FSE disk or tape media to an HSM file system. After a read error is detected, the status of the FSE medium is automatically set to “unusable”.



Caution: Before treating an FSE medium as unreadable, make sure it is not loaded in a faulty tape drive.

In certain circumstances, an FSE medium may acquire “unusable” status because of problems with the tape drive in which it is loaded. In this situation, the drive itself gets the “error” status.

You can get a list of all currently unusable media by filtering the output of the `fsemedium --list` command, as follows:

Linux:

```
# fsemedium --list | grep unusable
```

Windows:

```
%InstallPath%\bin>fsemedium --list | find "unusable"
```

The command output will be similar to the following:

```
000006  AIT-2  pool_A      unusable  drv:AIT-1 (lib:SA:6)
9      5
000045  AIT-2  pool_C      unusable  lib:SA:4          9
5
```

```
000103 LTO-1 LTO-pool_1 unusable drv:drive_1 (lib:Sa1:1)
1      0
```

For more information on the `fsemmedium` command, see “[fsemmedium](#)” on page 302.

Starting redundant copy recreation

The recreation process is executed for a single FSE medium at a time. The medium does not have to be allocated or loaded into an FSE drive. To start the redundant copy recreation process, invoke the following command:

```
# fsemmedium --copy-contents Barcode [--no-monitor]
```

By default, `fsemmedium` reports the progress of the maintenance job to the standard output and enables you to monitor the recreation process in the command shell. You can disable monitoring (make `fsemmedium` return to the command prompt immediately) by specifying the `--no-monitor` option.

With monitoring enabled, the command generates an output similar to the following:

```
# fsemmedium --copy-contents 020776
[11:42:09] Started (JobID: 20040121000002).
[11:42:09] Preparing list of source media
[11:42:09] 2 media needed for recall (bc: 050003, 050004)
[11:42:09] Request resources (bc: 050003) ...
[11:42:09] Resources allocated.
[11:42:09] Medium 050003 is already present in drive drv1.
[11:42:09] Starting Back End Agent.
[11:43:58] Medium 050003, volume 1: Reading files...
[12:20:58] Medium 050003, volume 1: Files written to disk
buffer.
[12:46:02] Reading finished, releasing resources.
[12:47:12] Requesting resources...
[12:47:12] Resources allocated.
[12:48:42] Loading medium 050010 to drive drv2.
[12:49:09] Starting Back End Agent.
[12:49:58] Medium 050010, volume 3: Writing disk buffer.
[13:00:58] Medium 050010, volume 3: Disk buffer written to
medium volume.
[13:11:50] Writing finished, releasing resources.
[11:42:09] Request resources (bc: 050004) ...
[11:42:09] Resources allocated.
[11:42:09] Medium 050004 is already present in drive drv3.
```

```

[11:42:09] Starting Back End Agent.
[12:21:02] Medium 050004, volume 2: Reading files.
[12:45:58] Medium 050004, volume 2: Files written to disk
buffer.
[12:46:02] Reading finished, releasing resources.
[12:47:12] Requesting resources...
[12:47:12] Resources allocated.
[11:42:09] Medium 050010 is already present in drive drv2.
[12:49:09] Starting Back End Agent.
[12:49:58] Medium 050010, volume 3: Writing disk buffer.
[13:00:58] Medium 050010, volume 3: Disk buffer written to
medium volume.
[13:11:50] Writing finished, releasing resources.
[13:12:38] Finished.

```

Regardless of the monitoring setting, the `fsemedium` command logs the recreation process progress into the FSE event log, while any errors encountered during the recreation process are logged into the FSE error log. These log files are located in `/var/opt/fse/log` (on Linux) or in `%InstallPath%\var\log` (on Windows).

Removing the unusable medium from the media pool

After the redundant copy recreation process has successfully finished, remove the damaged medium from the media pool with the following command:

```
# fsemedium --remove Barcode [--force]
```

If you specify the `--force` option, the command suppresses the confirmation question.

Resolving situations with no alternative migrated copies

In terms of data safety, a configuration with only one FSE media pool assigned to a particular FSE partition is dangerous. If a damaged, unusable FSE medium is part of such an FSE media pool, the only migrated data copy is permanently lost. The only exceptions are the migrated file generations on the unusable medium that are still present on the HSM file system. These file generations would normally be lost during their release, but you can prevent their loss by manually adding them to the migration candidate list.

To put online files on the migration candidate list:

```
# fsefile -M Filename...
```

Note that `fsefile` must be run on the FSE client that hosts the FSE partition.

The command puts all specified files on the migration candidate list. In this way, the files are scheduled for regular migration according to the migration policy defined for the FSE partition. To migrate the files immediately, use the `fsefile --trigger-migration` command.

For more information on the `fsefile` command, see “[fsefile](#)” on page 283.

Checking the Status of Media

Run `fsemedium --list [--detail]` to get information on media status for all media configured in the FSE installation. As shown in the example that follows, the output is the following:

- the medium barcode,
- the name of the media pool,
- the medium status,
- the physical location of the medium,
- the number of all medium volumes (data- and system volumes),
- the system volume number according to its location on the medium.

If you use the `--detail` option, `fsemedium` formats its output in a different way and displays the following additional media attributes:

- the type of slot in which the medium is located (*SlotType*),
- the serial number of the medium (*SerialNo*),
- the medium block size (*BlockSize*),
- the size of the system volume (*SysVolSize*),
- the UUID of the currently used medium volume (*CurVol*),
- the RMDB entry key of the medium (*Key*).

Example output of the `fsemedium --list` command

```
# fsemedium --list
Barcode  Type      Pool      Status      Location      Vol  SysVolNo
000045   AIT-2    poolAIT   open        lib:libAIT:3  9    1
000046   AIT-2    poolAIT   open        lib:libAIT:5  9    1
000116   AIT-2    poolAIT   uninitialized lib:libAIT:2  0    0
656701   UNKNOWN          uninitialized lib:libAIT:4  0    0
656702   UNKNOWN          uninitialized lib:libAIT:1  0    0
dm000001 DISK     poolDisk1 open        n/a          9    0
```

```

dm000002 DISK      poolDisk1 open          n/a          9    0
dm000003 DISK      poolDisk2 full            n/a          4    0
dm000004 DISK      poolDisk2 open          n/a          9    0
dm000005 UNKNOWN          uninitialized n/a          0    0

```

Example output of the `fsemedium --list --detail` command

```

# fsemedium --list --detail
Barcode:      000045
Type:         AIT-2
Pool:         poolAIT
LibName:      libAIT
SlotAddr:     3
SlotType:     repository
DrvName:
Status:       open
Location:     lib:libAIT:3
SerialNo:     ????????
BlockSize:   32
#Vol:         9
SysVolNo:     1
SysVolSize:   0
CurVol:      8
Key:          5

Barcode:      000046
Type:         AIT-2
...

```

Note: For disk media, the “location” information is not applicable (*n/a*). The “location” indicates the current location of the tape medium in an FSE library slot (library name and slot number) or in an FSE drive (drive name).

The status of the medium can be one of the following:

<code>free</code>	The medium is available.
<code>full (or closed)</code>	Medium volumes are full and Fast Recovery Information is already written on the medium.
<code>in use</code>	The medium is currently in use.
<code>open</code>	Medium volumes are not full yet and Fast Recovery Information is stored on disk only.

`uninitialized` The medium must be initialized before it can be used.

Additionally, if FSE detects errors on a medium, the medium is marked as one of the following:

`unreliable` It can be considered for recall only.

`unusable` It is not considered for any operation.

After you have checked the medium and determined its current condition:

- If it is corrupted, remove it.
- If the error was fixed, mark the medium as fully available. For details, see [“Putting an FSE medium back into use”](#) on page 105.

Checking the status of a particular medium

Once you know the medium barcode, you can narrow your search and view the status of the selected medium volume by using the `--list` option, followed by the medium barcode and the `--volumes` option.

As shown in the example below, the output is the following:

- medium barcode
- volume
- volume type
- size
- available space in GB
- usage in percentage
- medium volume status

If the status is “closed”, medium volumes are full and Fast Recovery Information is already written on the medium. If the status is “open”, medium volumes are not full yet and Fast Recovery Information is stored on disk only.

Example output of the `fsemedium --list --volumes` command

```
# fsemedium --list 000045 --volumes
Barcode  VolNum  Type    Size[MB]  Avail[MB]  Used[%]  Status
000045   1      System   1138      1126       1        open
000045   2      Data     1138       0         95       full
000045   3      Data    48172     42391      12       open
```

Changing the Condition Status of Media

The Resource Management Database recognizes three kinds of statuses which address a medium condition: good, unreliable, and unusable. Typically, the latter two are used whenever FSE detects errors on a medium. Such a medium is then automatically marked appropriately as either unreliable or unusable.

The FSE installation provides an option to use the medium status for preventing further use of specified media without having to put them off-site, and then to change their status back to be fully available. Depending on the operation you want to prevent being performed on a medium, you can mark it as either unreliable or unusable. To change a medium condition, you need to know its barcode.

Status “good” refers to the good condition of a medium. It needs to be set manually by the administrator if the medium was previously defective but has been repaired and is ready to be put back to use. If you have manually marked the medium as unreliable or unusable, and you want to make it “good” again, you also need to mark it as such.

Prevent writing to an FSE medium

If you want to prevent further writing to a medium, but still need to access data on the medium in case no other medium copy is available, set its status to unreliable:

```
# fsemedium --mark-unreliable Barcode
```

Once the medium is marked as unreliable, the medium can only be used for reading data.

Prevent reading from and writing to an FSE medium

If you want to prevent further use of a medium, set its status to unusable:

```
# fsemedium --mark-unusable Barcode
```

Putting an FSE medium back into use

If there were errors detected on a medium, but now the medium is fixed and you want to put it back to use, you can set its condition to good:

```
# fsemedium --mark-good Barcode
```

Once the medium is marked as good, data on the medium can be read and the medium can be written to.

Closing Medium Volumes

By closing a medium volume you mark it as full and set its status in the Resource Management Database accordingly; therefore further writing to it is no longer possible. This operation also saves Fast Recovery Information to the end part of the medium volume and to the medium system volume, if it is present. If writing Fast Recovery Information to a medium fails, an appropriate warning is displayed.

For example, if you want to duplicate a medium which is almost full, you can close the original medium volumes in order to prevent further writing to it. For more details, see [“Duplicating Media”](#) on page 85.

Closing a medium volume is also useful in situations where the status of a whole FSE medium is set to unreliable due to problems with a certain medium volume. Such a corrupted volume disables further use of other error-free medium volumes. When you close the unreliable medium volume and set its status to “good” (as described in [“Putting an FSE medium back into use”](#) on page 105) you allow further use of other medium volumes.

To close a medium volume, specify the medium barcode and volume number that you want to mark as full:

```
# fsemedium --close-volume Barcode VolumeNumber
```

Recreating Fast Recovery Information from Media

Recreating Fast Recovery Information (FRI) reads the complete contents of a medium volume and creates a new FRI file. The new file contains the same information but is usually more compact because it is created in a single pass.

Recreate FRI for the following:

- Open medium volumes for which on-disk FRI has been lost. The FRI will be stored to disk.
- Closed medium volumes that contain inconsistent or incomplete FRI. The existing FRI on the medium volume and system volume will be overwritten with the newly created FRI.

Inconsistent or incomplete FRI can be detected with the FSC vs. media check. For more information, see [“Consistency Check of FSC vs. FSE Media”](#) on page 160.

To recreate FRI, use the following command:

```
# fsemedium --recreate-fri Barcode [--volume VolumeNumber]
```

Note: Recreating FRI on single-volume media such as LTO 1, LTO 2, and SAIT can take a considerable amount of time.

Removing FSE Media

You can only remove a medium from the FSE installation if a medium has not yet been used (initialized), or the respective FSE partition was removed from the FSE installation and the medium is not assigned to the partition. Before removing a medium, you need to make sure that all FSE medium volumes on it are empty:

```
# fsemedium --remove Barcode
```

Note: When you remove an FSE medium using `fsemedium --remove`, it is also logically removed from the FSE library in which it is located. This means that the medium is no longer reported in the output of the `fselibrary --list` command. If the medium is still physically present in the library, trigger a library inventory update to include the medium barcode again in the logical inventory of the library. Use the `fselibrary --update-inventory` command for this purpose.

Migration, Release, Recall, and Deletion



This chapter includes the following topics:

- [FSE Operation Basics](#), page 109
- [Migration](#), page 114
- [Release](#), page 117
- [Recall](#), page 128
- [Deletion](#), page 129
- [Resource Allocation](#), page 134
- [System Allocation and Job Priority Policy](#), page 136
- [Partition Allocation and Job Priority Policy](#), page 139
- [Priority Calculation](#), page 142
- [Multiple Copying](#), page 144

FSE Operation Basics

How does FSE function?

FSE keeps track of all migration and release tasks, thereby freeing users and administrators from constantly monitoring the state and use of disk capacity. Once the file data is released, the file directory containing information such as filename, access rights, time and date, and so on is maintained on the disk. This way you can access and recall the files at any time. The file's location—that is, whether its data resides on the primary storage space (directly on the hard disk partition) or on the secondary storage space (media)—is transparent to the user. You see network file systems or network drive letters (CIFS) in your list of system drives. If you want to access one of these files, the file is either available on the file system or automatically recalled from the fastest available secondary medium transparently to the user/application.

Policies

FSE uses a powerful sets of rules to automate migration and release jobs and the deletion of expired files: the migration, release, and deletion policies.

- The migration policy defines a set of migration rules that govern the behavior of FSE's automatic and regular migration of files to FSE media.
- The release policy includes watermarks that control file data release jobs and rules for exclusion of particular files from the release process.
- The deletion policy defines expiration periods for directories on the HSM file system, which are considered when the deletion job is triggered by an FSE user.

Recalls are handled differently: they are started automatically when an FSE user wants to read an offline file or write to it.

HSM lists

FSE manages migration and release jobs using lists of files that are candidates for migration or release, handled by the Hierarchical Storage Manager (HSM). Both jobs are configured and controlled independently.

FSE jobs

The essential jobs that are executed in FSE are migration, recall and release jobs. Migration and release jobs are usually configured to be triggered when certain criteria are fulfilled. These criteria are defined by the corresponding migration and release policies. In some situations, an explicit migration or release is required, which can be started by an FSE user with the `fsefile` command. A recall job is always triggered implicitly, when an FSE system user requests access to an offline file.

With migration and recall jobs, the Partition Manager executes the appropriate sub-process—the job that executes the defined task. When the FSE job completes its task, it is terminated. Release, however, is a simpler process that does not require an FSE job. It is managed and executed locally by the Hierarchical Storage Manager (HSM).

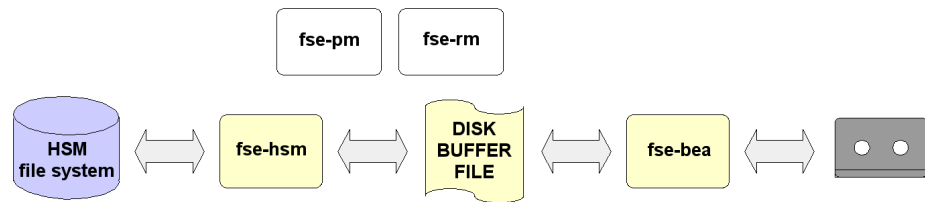
FSE disk buffer

Files that are either waiting for migration and are not yet written to the FSE media, or have been recalled and are not yet recreated on the HSM file system, are copied to the FSE disk buffer. The disk buffer holds temporary files that are waiting for resources and allows FSE to unlock files on the HSM file system

during migration, and de-allocate FSE media during recall as fast as possible. Usually, a separate disk partition is allocated for the FSE disk buffer to improve robustness and to avoid potential data loss.

Temporary FSE disk buffer files are located in the `/var/opt/fse/diskbuf/` directory on Linux and in `%InstallPath%\var\diskbuf` on Windows. They are managed by the FSE Back End Agent.

The following figure shows how the disk buffer is used.



Migration and recall path

FSE Hierarchical Storage Manager Lists

All migration and release jobs are carried out using Hierarchical Storage Manager (HSM) lists. These lists provide information about the candidates to be migrated or released and gather them accordingly. FSE executes migration or release once for a group of files rather than executing these jobs separately for each file thus improving performance. However, recall is executed for each file separately in order to fulfill the user’s request for a specific offline file without delay.

HSM maintains three types of lists: a list of changed files—a so-called “dirty files” list—a migration candidate list, and a release candidate list. Each file entry can only appear on one list at a time. Either it is a dirty file, and should not be migrated yet, or it is a migration candidate and is waiting for migration, or it is a release candidate and is waiting to be released. Once a file is released, it is cleared from the release candidate list. The file is then offline until the next recall. Offline files are not a part of any list. Their status is recorded in the HSM file system in the HSM attributes (stored in `EXT3/NTFS` extended attributes).

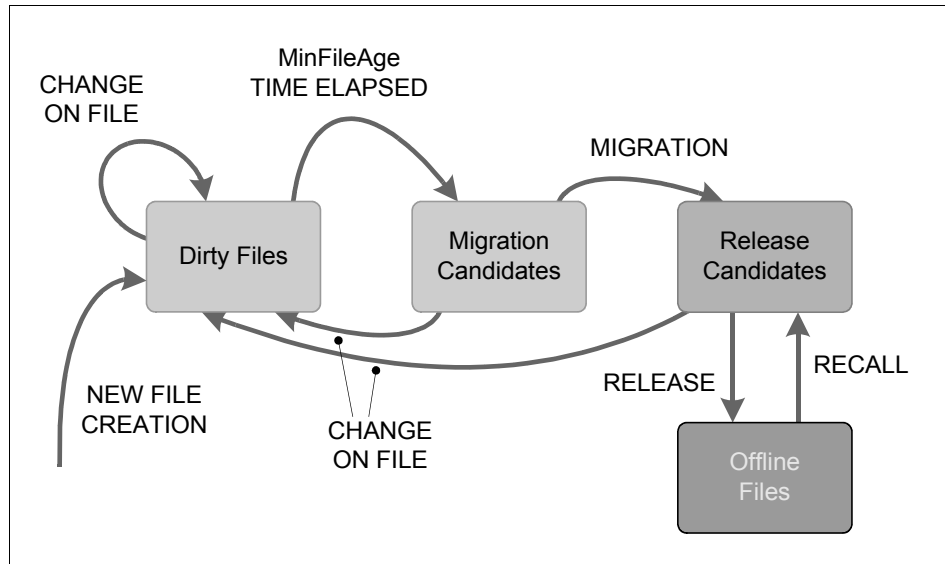
Types of HSM lists

FSE HSM list	Description
Dirty files list (list of changed files)	Gathers files that were recently accessed and changed. A dirty file can be either a new file or a file that has changed since the last migration or recall. A file is added to this list once you start changing it.
Migration candidate list	Once a changed file is not modified for a certain period of time, the file is moved onto the migration candidate list.
Release candidate list	Gathers online files that have already been migrated and have not changed since then, and files that were brought online by recall.

The way transitions between HSM lists and other aspects of migration, recall, and release are controlled is defined by parameters in the configuration file for the FSE partition. These parameters are described in “[Migration](#)” on page 114, “[Release](#)” on page 117, “[Recall](#)” on page 128, and in “[System Allocation and Job Priority Policy](#)” on page 136 and “[Partition Allocation and Job Priority Policy](#)” on page 139.

The following graphic shows file-status changes, including transitions of file entries between FSE HSM lists. The *MinFileAge* parameter is one of the configuration parameters for an FSE partition. It represents the period of time during which a file must be left intact to become a migration candidate.

Changes of state of files under FSE control



Example: FSE migration and release scenario

The example presents how migration and release work according to specified policies.

1. A user has created a file and saved it as `myfile.doc`, and he continued to work on the file. FSE HSM noticed the change and added `myfile.doc` to the dirty file list.
2. The user has stopped working on `myfile.doc` and closed it. After a predefined time to wait after file change before migration, based on the migration policy parameter `MinFileAge`, the file is passed from the changed file list to the migration candidate list.
3. The time till migration is set to 20 minutes. After 20 minutes `myfile.doc` is migrated to media.
4. The file `myfile.doc` is then added to the release candidate list. When the used capacity of the HSM file system exceeds a certain percentage, defined with a high watermark, for example 85%, FSE HSM triggers a corresponding release action. `myfile.doc` is released together with other files from the release candidate list and is now offline.

About Policies

FSE control over the files in use is based on user-defined policies. These are rules by which migration, release, and deletion jobs are executed. Each process requires its own policy, thus migration policy, release policy, and deletion policy. Each FSE partition has its own set of policies.

Different policies can be defined for individual FSE partitions, according to the partition's purpose and the way it is used. Policies are defined in FSE partition configuration files as a series of parameters. These are read and used by the FSE Hierarchical Storage Manager and the FSE Partition Manager at startup (migration and release policies) or at execution of the `fsefile --trigger-deletion` command (deletion policy). The parameters are described in comments inside the FSE configuration file templates.

You need to plan in advance to set the policies according to your environment and to optimize the jobs. The way you set up these policies will affect your installation's performance.

Migration

You define configuration parameters, relevant for your migration policy, in the FSE partition configuration file. It is very important that you create a migration policy appropriate for your environment. You should check the default migration policy and adjust settings according to the characteristics of your files.

FSE migration policies are based on file age, wait time for migration, number of files waiting for migration, and the migration volume. Applying these migration parameters in combination creates archival (migration) and retrieval (recall) processes that make the most effective use of the underlying hardware and network infrastructure.

Note: You can make redundant copies of the file data during the migration job. For more information, see [“Multiple Copying”](#) on page 144.

What is a retried migration?

In cases of a failed migration, where the files would stay on the dirty files list and would not be migrated, you can remigrate such files manually or schedule regular execution of the migration using a script, such as `cron`.

Use the following command to trigger migration:

```
# fsefile --trigger-migration PartitionName
```

Adding files to the dirty files list manually

Sometimes it can happen that a file is present on the HSM file system but is not recognized by FSE at a given time. By putting it to the dirty files list the file is recognized and its attributes are preserved. Another file could already be migrated to the FSE medium that has its current status set to unreliable, and is only available for recall if no other copy exists. Such a file can be remigrated to another medium.

In such cases you should manually add files to the dirty files list as follows:

```
# fsefile --migrate FileName
```

Once the files are added to the dirty files list, they are regularly migrated according to the migration policy. If the specified file is offline, it will be recalled and then regularly migrated with other files on the list.

Note that zero-size files are always online; they cannot be migrated nor released.

Migration Policy Parameters

The configuration parameters for migration along with their definition and impact are listed below. The combination of these parameters allows a flexible configuration of migration behavior.

Minimum file age for migration (*MinFileAge*)

<i>Definition:</i>	A period of time after file creation/change during which a file must be left intact to become a migration candidate.
<i>Impact:</i>	You should define the parameter value according to the characteristics of your file data; if your parameter value is too high, for example 5 minutes, and the file is constantly changing (e.g. it is updated every 2 minutes), the file will not be migrated to media.

Minimum wait time for migration (*MinWaitTime*)

<i>Definition:</i>	Minimum wait time for files on the migration candidate list during which no more files are added to the list.
<i>Impact:</i>	If this parameter value is high, for example 8 hours, you will have irregular and substantial migrations. A low value results in frequent, smaller migrations.

Maximum wait time for migration (*MaxWaitTime*)

<i>Definition:</i>	Maximum wait time for migration; if reached, migration is started immediately. Starts counting since the addition of the earliest file.
<i>Impact:</i>	Without this parameter, if new files keep arriving to the migration list fast enough, the <i>MinWaitTime</i> would never be reached and the files would never get migrated. <i>MaxWaitTime</i> ensures this does not happen. When the oldest file has been on the list long enough, migration is started.

Minimum number of files for migration (*MinNumMigFiles*)

<i>Definition:</i>	Minimum number of files on the migration candidate list before migration is started.
<i>Impact:</i>	You should define the parameter value according to the characteristics of your file data; create a rate according to the approximate size of your files. This parameter value should be higher if your files are small, and vice versa.

Maximum number of files for migration (MaxNumMigFiles)

Definition:	Maximum number of files on the migration candidate list; if reached, migration is started immediately.
Impact:	Without this parameter, if new files kept arriving to the migration list fast enough, the <i>MinWaitTime</i> would never be reached and the files would never get migrated. <i>MaxNumMigFiles</i> ensures this does not happen. When the maximum number of files for migration is reached, migration is started.

Minimum size for migration (MinMigSize)

Definition:	Minimum cumulative size of all files on the migration candidate list before migration is started.
Impact:	You should define the parameter value according to the characteristics of your file data; create a rate according to the approximate size of your files. The recommended value is above 2 MB in order to provide enough data to keep the tape drives streaming.

Maximum size for migration (MaxMigSize)

Definition:	Maximum cumulative size of all files on the migration candidate list; if reached, migration is started immediately.
Impact:	Without this parameter, if new files kept arriving to the migration list fast enough, the <i>MinWaitTime</i> would never be reached and the files would never get migrated. <i>MaxMigSize</i> ensures this does not happen. When the maximum size is reached, migration is started.

Note: A general rule for the migration to start is either there are all *Min* parameters reached or there is one *Max* parameter reached that will start the migration job. *Max* parameters can prevent FSE from filling up the disk in case your *Min* parameters are not set properly and the migration is not started at the right time.

Example situations that trigger migration

There are four distinct situations that trigger a new migration job.

- Situation 1

There were no files added to the list in the last *MinWaitTime* period of time and there are more than *MinNumMigFiles* files on the list and total size of files on the list is at least *MinMigSize*.

- Situation 2
The period of time since the first file was put on the list is greater than `MaxWaitTime`.
- Situation 3
There are at least `MaxNumMigFiles` on the list.
- Situation 4
The total size of files to be migrated is equal or greater than `MaxMigSize`.

Default migration policy

The following is the default migration policy for an FSE partition.

```
[Migration]
MinFileAge      = 10s

[RegularMigration]

MinWaitTime     = 20s
MinNumMigFiles  = 1
MinMigSize      = 2MB
MaxWaitTime     = 30m
MaxNumMigFiles  = 10000
MaxMigSize      = 400MB
```

Release

In contrast to migration policy, the most important release policy parameters are watermarks. The watermark parameter values define when the release process is triggered. You define configuration parameters relevant for your release policy in the FSE partition configuration file. You should check the default watermark parameters for your release policy and adjust the settings according to your environment.

You can also exclude specified files from the release process. This allows you to specify files or sets of files which should be skipped during a release operation so that they remain permanently online. Exclusion from release is described in [“Exclusion from Release”](#) on page 121.

Watermarks

FSE prevents HSM file systems from becoming full using special parameters called watermarks. They are based on a simple principle of defining the value for the normal, high, and critical occupation of disk space. Watermark parameters are the trigger for releasing the file data from the file system.

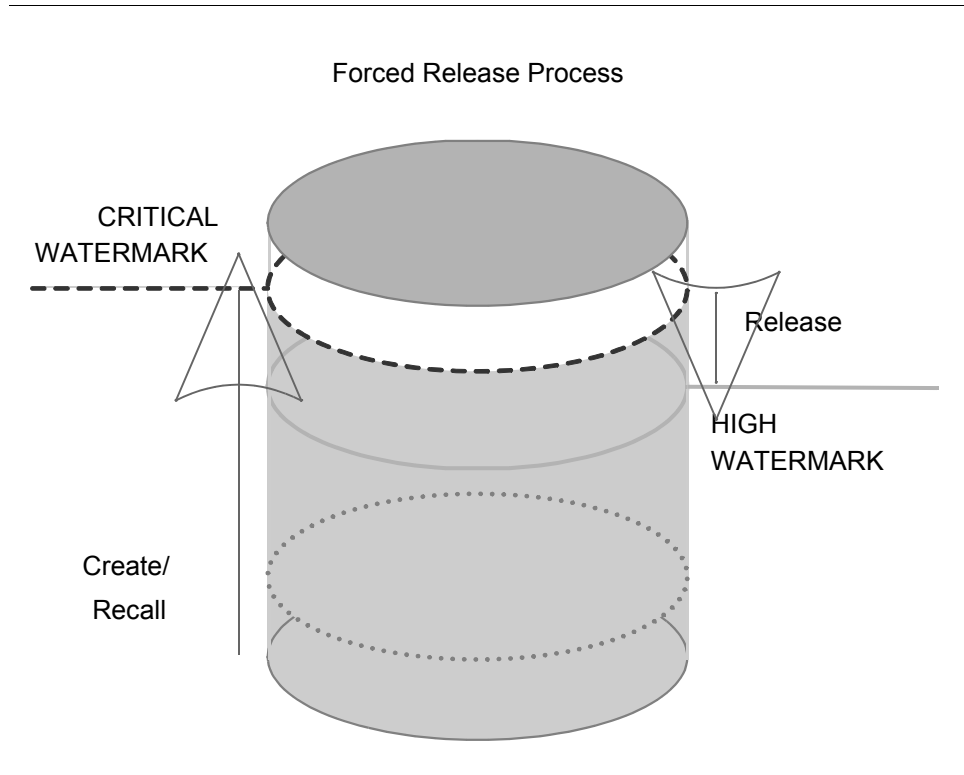
Typically, when the HSM file system volume reaches a certain percentage (defined with a watermark), all files that meet defined criteria are released from the disk. This results in freeing up capacity on disk systems thus providing space for new files and files that will be recalled.

For each configured FSE partition, three watermarks are defined in the FSE partition configuration file: low, high, and critical. Their values must follow the rule “*low watermark < high watermark < critical watermark*”.

Critical watermark

When this watermark is reached, the forced release job is started. All files that are on the release candidate list are released, regardless of other conditions—retention times and minimum file size are not checked. The release job is stopped when the high watermark on the HSM file system is reached. Files not released during the release job remain on the release list.

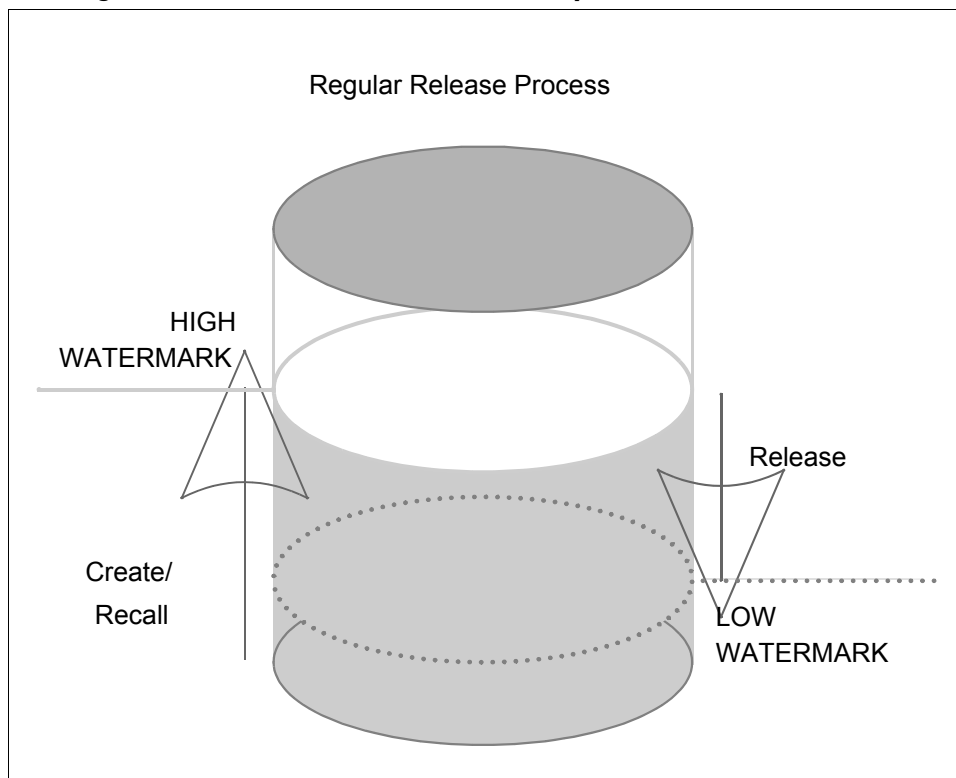
Note that the regular release job starts immediately after the forced release job has ended (because the system is still filled to the high watermark).



High watermark

High watermark is usually reached when new files are created or copied to the HSM file system. Regular release is started for all files on the release candidate list that have passed their migration or recall retention period, and are larger than *MinFileSize*. This stops when low watermark on the HSM file system is reached or when there are no further valid files on the release candidate list. Files not released remain on the release candidate list.

Reaching the critical watermark on the HSM file system



Low watermark

Low watermark is considered during a regular release triggered by reaching the high watermark on the HSM file system. The low watermark is the parameter that defines to what percentage volume the disk space should be freed, and stops the release when it has achieved this level. When it is reached, files that were not released at this point remain online and are kept on the release candidate list.

Exclusion from Release

In certain situations, you may prefer certain files in the HSM file system not to be released, but to remain permanently online. The reasons for this strategy, for example, relate to the types of files you have:

- Certain files or file types of special importance demand instant access. These files must not be released.
- Certain files or file types are used frequently enough for their release to serve no purpose and to waste FSE resources. These files can be held permanently online, as long as their total size is not prohibitively large.

FSE enables you to define sets of files that are excluded from the release process and kept resident on the HSM file systems. This feature is called exclusion from release and is part of the release policy. It is defined by the value of the *ExcludeFromRelease* variable in the FSE partition configuration file, and therefore the sets of files to be excluded are defined separately for each FSE partition.

You can also force the files that are excluded from release to be put offline unconditionally. You achieve this by releasing them manually using the `fsefile --release` command. Note that using this command on the excluded files does not modify the release policy in any way. After the manually released files are re-migrated or recalled, they are still excluded from release.

In the FSE partition configuration template, the *ExcludeFromRelease* variable is commented out. Thus, no files are excluded from release by default. You can define the files you want to keep permanently online by uncommenting *ExcludeFromRelease* and specifying appropriate filenames or filename patterns.

Note: Zero-sized files remain permanently online. They cannot be migrated nor released.

Limitations

- Only sets of files can be excluded from the release process. You cannot exclude directories.
- When an FSE partition is reconfigured by modifying the *ExcludeFromRelease* variable, the contents of the corresponding release candidate list is not affected. This means that:

- Files that have previously been excluded from release, but are not now excluded by the reconfigured `ExcludeFromRelease` parameter, *will not be released* until they are migrated or recalled again.
- Files that are on the release candidate list, and are now excluded from release by the reconfigured `ExcludeFromRelease` parameter, *will be released*. They will be excluded from release only after their first subsequent migration or recall.

Considerations before configuring exclusion from release

Keep in mind that configuring files for exclusion from release such that the total size of the excluded files is relatively large can cause undesirable consequences. In these cases, you may experience the following symptoms:

- The frequency of reaching the high and critical watermarks on the HSM file system increases. The more the total disk space is occupied by excluded files, the greater the frequency of reaching the watermarks.
- The low watermark is never reached. This happens when the total amount of disk space used for storing the excluded files exceeds the low watermark size configured on the corresponding FSE partition.

These inconveniences are a logical consequence of permanently maintaining a greater number of files on the HSM file system. You can try to minimize their effects by keeping the number and size of excluded files within reasonable bounds. You may also want to consider adjusting the watermark values.

Filename pattern conventions

An `ExcludeFromRelease` variable configuration includes a comma-separated list of filenames or filename patterns or both. Filename patterns are defined by special pattern-matching characters.

The scope of filenames and filename patterns is over an entire HSM file system. All files on a particular FSE partition whose filenames match the specified names or patterns are excluded from release. Although you can specify full paths with filenames, only the filename part of the filename pattern is considered for pattern matching—the directory part is ignored.

The following are examples of properly configured *ExcludeFromRelease* variables:

```
[ ReleaseFiles ]
MigRetentionTime = "1h"
RecallRetentionTime = "2h"
MinFileSize = "2kB"
ExcludeFromRelease = (*.exe, *.ini, fse_*, "[0-9]*")

[ ReleaseFiles ]
MigRetentionTime = "2h"
RecallRetentionTime = "10h"
MinFileSize = "2kB"
ExcludeFromRelease = ("*_[a-z].tgz", *.x??, core, "*",v")

[ ReleaseFiles ]
MigRetentionTime = "2m"
RecallRetentionTime = "4m"
MinFileSize = "2kB"
ExcludeFromRelease = ( )
```

Note: When configuring the *ExcludeFromRelease* variable, filenames and filename patterns must be specified in the UTF-8 format.

Special pattern-matching characters

The filename pattern-matching mechanism built into the FSE conforms to that used in command shells on UNIX systems, with a few differences. Special characters are permitted to simplify the definition of sets of files.

The following table lists the special characters, meanings, and usage examples:

Special pattern-matching characters used with *ExcludeFromRelease*

Special Character	Meaning	Description	Examples
?	any character	matches any single character, including the dot (.)	MyProject?? your_proj.c??
*	closure	matches zero or more characters, including dots	image_* D*.exe "* local"
[]	set	matches a single character from the set of characters enclosed within the brackets	"Image_[012]?? .png" "installation_[ABC].sh"

Special Character	Meaning	Description	Examples
[-]	range set	matches a single character from the range of characters; range sets can be combined with ordinary sets	"rep-[2-6]?0" "CurrentDoc-[ABCD][a-z].txt" "weekly_rep-[1-589]00.doc"
\	quote character (Linux)	suppresses the special meaning of the character that follows	"file1_*[A-\]].*" "file2_["-\-]"
` (grave accent)	quote character (Windows)	suppresses the special meaning of the character that follows	"file1_*[A-`]].*" "file2_["-`-]"

Notes:

- On case-insensitive file systems, like NTFS on Windows, if a character is covered by a filename pattern, then this character's upper- or lower-case counterpart is also covered.
- All the filenames and filename patterns that contain space characters or commas, and all filename patterns that use sets ([]) or range sets ([-]) must be enclosed within double quotes (" ").
- A right bracket (]) can only be used inside sets and range sets if it is quoted (\]).
- A hyphen (-) can only be used inside sets and range sets if it is quoted (\-), or if it is the first character specified inside the set or range set ([- . . .), or if it is specified as a single-character range (---).
- Quote characters can only be matched if they are quoted. On Linux systems, to match a backslash (\), use two backslashes (\\). On Windows systems, to match a grave accent (`), use two grave accents (``).

Configuring exclusion from release

After you have modified the FSE partition configuration template or an already used FSE configuration file with a configured *ExcludeFromRelease* variable, you have to apply it by running the `fsepartition --add` or the `fsepartition --modify` command, respectively. Potential syntax errors in the *ExcludeFromRelease* parameter specification are immediately reported to the standard command output.

For information on the `fsepartition` command, see “[fsepartition](#)” on page 313.

Releasing excluded files

The files that have been excluded from release can be unconditionally released using the following command:

```
# fsefile --release FileName...
```

You must run the command on the FSE client that hosts the chosen FSE partition.

For more information on the `fsefile` command, see “[fsefile](#)” on page 283.

Release Policy Parameters

The following are the configuration parameters for release along with their definition and impact. The combination of these parameters allows a flexible configuration of release behavior.

Critical watermark (CriticalWaterMark)

Definition:	The percentage of used disk space that triggers a forced release job.
Impact:	This parameter value should not be too low as it will result in frequent forced releases (it should typically be above 95%).

High watermark (HighWaterMark)

Definition:	The percentage of used disk space that triggers a regular release job.
Impact:	Consider the value for acceptable occupation of disk space. If you set a low value, it will result in numerous regular releases.

Low watermark (LowWaterMark)

Definition:	The percentage of used disk space that stops the regular release job—the files below the defined low watermark are not released and remain on the disk.
Impact:	Consider the value for disk space needed for the number of small files that should not be released if you want to optimize the use of the FSE installation.

Release interval (ReleaseInterval)

Definition:	The frequency of the disk space checks for watermark conditions.
Impact:	Define this value according to the frequency of recalls and the size of the recalled files. If this value is too high, for example 5 minutes, and you have constant recalls of large files, the file system will be overloaded. Such occupation of disk space will make all recalls of offline files impossible.

Migration retention time (MigRetentionTime)

Definition:	A period of time during which a certain file is going to be kept online after migration before it will be released.
Impact:	Consider for how long a migrated file should stay online. Adjust the value according to your data characteristics. If you use the partition for archiving purposes, the value should be low. Partitions with frequently used files should have a higher value set, so that the files do not get released and recalled all the time.

Recall retention time (RecallRetentionTime)

Definition:	A period of time during which a certain file is going to be kept online after recall before it will be released again.
Impact:	Consider for how long a recalled file should stay online. If the specified time is too small, such a file will be recalled every time it is accessed; if it is too long, it will take up disk space without any reason.

Minimum file size for release (MinFileSize)

Definition:	Minimum size for a file on the release candidate list to be regularly released. In case of a forced release, this parameter is disregarded and files of all sizes will be released.
Impact:	It is not recommended to put very small files offline since the load of the library and media in this case is rather significant whereas the disk space usage is hardly affected. The recommended value is above 2 KB (default).

Note: On Windows, the recommended *MinFileSize* for release should not be smaller than the NTFS file system cluster size (allocation unit). This is typically 4 KB for volumes larger than 2 GB. A lower specified value would not free any additional disk space. (The NTFS file system cluster size is selected when formatting the NTFS volume.)

Files that are never released (`ExcludeFromRelease`)

Definition:	A comma-separated list of filenames and filename patterns that defines a set of files that will be skipped by the automatic release process. If the list is empty or this variable is not configured, all files are considered for automatic release.
Impact:	The files covered by this parameter will not be released. Instead, they will be kept permanently online, with their data residing on the HSM file system.

Example situations that trigger release

There are two distinct situations that trigger a new release job.

■ Situation 1

A high watermark is reached on the HSM file system. At this point, all files on the release candidate list larger than the `MinFileSize` variable value and with their retention time reached are scheduled for regular release. The release job is stopped when the low watermark is reached or if there are no more files on the release candidate list that match the specified criteria.

■ Situation 2

A critical watermark is reached on the HSM file system. In this case, all files on the release candidate list (regardless of their size and retention time) are scheduled for forced release. The release job is stopped when the high watermark is reached or if there are no more files to release.

In both cases, the files that are on the release candidate list and were not released during the most recent release job remain on the top of the release candidate list.

Default release policy

Note that the default release policy does not include the `ExcludeFromRelease` variable. If you do not configure this variable, all files are considered for release according to the other release policy parameters specified while configuring the FSE partition.

The following is the default release policy for an FSE partition:

```
[Release]
CriticalWatermark = 95%
HighWatermark    = 85%
LowWatermark     = 35%
ReleaseInterval  = 30s
```

```
[ReleaseFiles]
MigRetentionTime = 2h
RecallRetentionTime = 24h
MinFileSize      = 2kB
```

Recall

A recall job is started on a request for a read or write operation on an offline file. The contents of the file are then retrieved from the FSE medium and copied back to the HSM file system. The only parameter for the recall configuration is the timeout value.

Recall behavior is automated to the extent that recall is retried from alternative copies in cases that a resource allocation is affected. This usually happens when a medium is corrupted and marked in the Resource Management Database accordingly as unusable, or when a hardware error occurred and the drive is marked as bad.

In such cases, the recall job requests new resources to perform the operation until reading from the tape is successful, or all resources (media that contain respective data copies or drives) are exhausted. In this latter event, the recall job fails.

Note: An error message is issued regardless of the recall retry if the recall lasts longer than specified with the *RecallTimeOut* parameter. However, when resource allocation is disturbed as described, this parameter is ignored and recalling a file succeeds if any copy is available.

Recall Timeout Parameter

There is only one configurable parameter for the recall policy:

Recall timeout (RecallTimeout)

Definition:	When a user requests an offline file, the Recall Timeout is the time given to FSE to bring an offline file back online. If FSE cannot bring the file online in this time, the user is notified that the file is unavailable and the request is discarded.
Impact:	Set the value according to the load of the FSE installation. The value can be lower if your system performance is optimized. If the load of the FSE installation is significant and the value is low, the following can happen: <ul style="list-style-type: none"> ■ At the defined time all drives are in use. ■ There is a significant number of jobs with a higher priority already running. ■ The needed medium cannot be found (it is offline). ■ In such cases, the HSM notifies the user that the file is not available.

Note: Having multiple media copies can speed up the recall of files, since any of the copies can be used for recall. For further information on multiple copying, see [“Multiple Copying”](#) on page 144.

Default recall policy

The following is the default recall policy for an FSE partition:

```
[Recall]
RecallTimeout          = 15m
```

Deletion

FSE provides the ability to delete files automatically on HSM file systems. You can use this feature for various purposes, such as in an information lifecycle management (ILM) environment made necessary by legal requirements, or for freeing disk space of redundant, out-of-date files.

Automatic Deletion Process

Automatic deletion is a process executed on an FSE partition. The process is controlled and performed by a special type of FSE job: the deletion job. You configure FSE deletion jobs by specifying a deletion policy in each FSE partition configuration file. The deletion policy determines the files considered for deletion and the criteria that determine when they are deleted.

The automatic file deletion process consists of two phases. When an FSE deletion job is initiated, it first scans the File System Catalog (FSC) for files on the FSE partition and makes a list of the files that satisfy the deletion policy criteria. This list is then passed to Hierarchical Storage Manager (HSM), which deletes the files on the HSM file system.

Note: Automatic deletion is also supported on WORM HSM file systems. During deletion, the files are deleted from the HSM file system. Afterwards, the deleted files are “migrated” to the WORM FSE media.

To trigger an FSE deletion job, you use the `fsefile` command, specifying the name of the FSE partition on which you want to run the deletion job. You can also configure a regular cron job (on Linux) or schedule a task (on Windows) and include the `fsefile` invocation in it. This way, you can completely automate the file deletion tasks.

The progress of the deletion process is logged to the FSE event log, and potential errors are logged to the FSE error log. The list of files that were deleted is written to the HSM file system log file on the FSE client that hosts the FSE partition. The HSM file system log file is called `hsmfs_<PartitionName>.log` and located in `/var/opt/fse/log/` (on Linux systems) or in the `%InstallPath%\var\log` directory (on Windows systems).

Deletion Policy Parameters

You can set up a deletion policy for an FSE partition by specifying appropriate parameters in the FSE partition configuration file. The deletion policy can be set up while configuring a new FSE partition or when modifying an existing FSE partition configuration.

You define a deletion policy with pairs of *FileExpiration* and *Path* variables. Each specified *FileExpiration* variable requires a corresponding *Path* variable.

File expiration time (*FileExpiration*)

Definition:	A period of time that determines which files in the specified <i>Path</i> directories are to be considered for deletion. The following units are available for specifying the time period: <i>s</i> (seconds), <i>m</i> (minutes), <i>h</i> (hours), <i>D</i> (days), <i>M</i> (months), and <i>Y</i> (years).
Impact:	This variable determines the files to consider for deletion by checking the period of time since each file on the HSM file system was last changed; files changed more than <i>FileExpiration</i> time ago are added to the file list. If the UNLIMITED value is specified for the <i>FileExpiration</i> variable, the files will never be considered for deletion. However, you can still delete such files manually using operating system utilities.

Directory paths (*Path*)

Definition:	A set of directory paths on the HSM file system on whose files the corresponding file expiration period <i>FileExpiration</i> is considered.
Impact:	During a deletion job, all files located in the directories specified by the <i>Path</i> variable (including all subdirectories), and for which the corresponding expiration period has passed, are deleted from the HSM file system.

When specifying the *FileExpiration* and *Path* variables for the deletion policy of an FSE partition, keep in mind the following rules:

- Directory paths are relative to the mount point of the HSM file system (the directory where the HSM file system is mounted).
- Directory paths that contain spaces must be enclosed within double quotes (" ").
- A particular directory path cannot be specified more than once.

Examples of a Configured Deletion Policy

The following are two excerpts from FSE partition configuration files. They show examples of the configured deletion policies and their explanations: one for an FSE partition located on a Windows FSE client, and the other for an FSE partition located on a Linux FSE client.

Linux:

```
[Delete]

FileExpiration "UNLIMITED"
{
Path = (/archive/permanent_files)
}

FileExpiration "2D"
{
Path = ("/tmp/daily reports", /tmp/temporary)
}
```

This deletion policy will have the following effect:

- All files in the directory `/archive/permanent_files` on the owning HSM file system will never be considered for deletion.
- All files in the directories `"/tmp/daily reports"` and `/tmp/temporary` are to be considered for deletion as soon as two days have passed since they were last changed.

Windows:

```
[Delete]
FileExpiration "1M"
{
Path = (\Months\Current, "\Arch\June 2004")
}
FileExpiration "7D"
{
Path = (\Weeks\Current, "\Arch\Week 24")
}
```

This deletion policy will have the following effect:

- All files in the directories `\Months\Current` and `"\Arch\June 2004"` on the owning HSM file system are to be considered for deletion as soon as one month has passed since they were last changed.
- All files in the directories `\Weeks\Current` and `"\Arch\Week 24"` are to be considered for deletion as soon as seven days have passed since they were last changed.

Part of the FSE partition configuration template that includes the deletion policy is depicted in [“Template for FSE Partitions—Deletion Part”](#) on page 349.

Starting the Deletion Job

You start a deletion job by running the following command on the FSE client:

```
# fsefile --trigger-deletion PartitionName [--force]
  [--no-monitor]
```

Before running the command, ensure that the HSM file system of the selected FSE partition is mounted.

By default, `fsefile` requires confirmation before starting the deletion process. You can suppress the confirmation request by specifying the `--force` option. This is particularly useful when using the command within scripts.

During execution of the deletion job, `fsefile` reports the job execution progress to the standard output, allowing you to monitor the deletion process in the command shell. You can disable monitoring (force `fsefile` to return to the command prompt immediately) by specifying the `--no-monitor` option.

Note: If two consecutive deletion jobs using the same deletion policy are started on the same HSM file system in a short period of time, the files that were deleted by the first job may show up as still present on the HSM file system during the execution of the second job. This happens if the deletions have not yet “migrated” to FSE media by the time the second job starts. In this case, the second deletion job does not perform any operation on these files, but returns an error for the files that have already been deleted.

If the HSM file system on which the deletion job is running is unmounted during the job, the job is aborted. To complete the deletion process, you need to restart it from the beginning.

For more information on `fsefile`, see “[fsefile](#)” on page 283.

Recalling Deleted Files

Files deleted from an HSM file system can still be recreated using FSE’s “recall by file ID” functionality. File IDs are stored in the File System Catalog (FSC) and can be retrieved with the `fsefile --history` command. To recall deleted files using their file IDs, run `fsefile --recall` on the FSE client. You must run the command separately for each deleted file.

Note that files that were deleted from an HSM file system and whose file data was stored on FSE medium volumes that were subsequently reorganized can no longer be recalled. This happens when all migrated generations of the deleted files were considered obsolete (outdated) by the FSE media reorganization process and all medium volumes that stored these file generations were reorganized.



Caution: When sensitive data is to be completely erased from the FSE installation, you must ensure that they are first deleted from the HSM file system, and then reorganize its FSE media so that all migrated generations (and all additional copies if multiple copying is configured for the appropriate FSE partition) of the data are considered as obsolete.

Additionally, if the migrated data reside on WORM-type media, these media must be removed from FSE and destroyed.

Resource Allocation

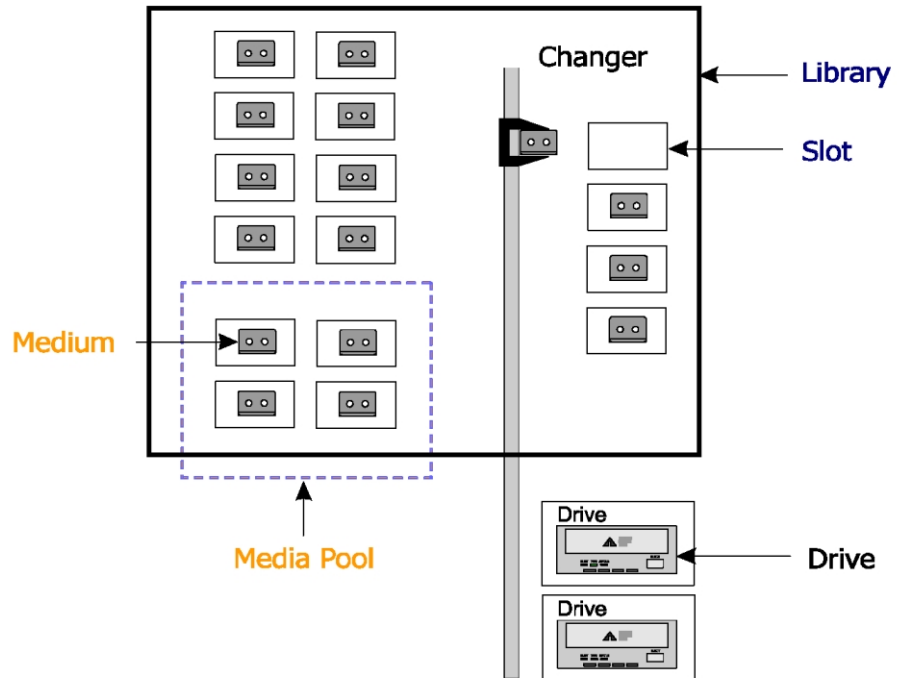
FSE resources are physical and logical objects, controlled by the FSE installation. For detailed information on configuring resources, see “[Configuring FSE](#)” on page 33.

Resource Allocation Concepts

FSE resources can be grouped according to their characteristics, as follows:

- Physical resources:
 - FSE library, including library slots
 - FSE drives
- Logical resources:
 - FSE medium cartridge, including medium volumes
 - FSE media pool

The following figure schematically shows the resources as they are represented in the FSE installation.



Example of FSE resources (tape only)

Central FSE resource manager

All resources of the FSE installation are controlled by a central FSE Resource Manager. It manages the Resource Management Database and allocates resources. Resource allocation is used each time an FSE job requires new resources. Typically, the FSE Resource Manager locks the use of resources in the Resource Management Database for jobs that have requested allocation.

Media selection by media location

Allocation selects a medium according to its location. Possible locations of a medium are:

- **Online:** Medium is loaded into a drive.
- **Nearline:** Medium is in a library slot.
- **Offline:** Medium location is unknown.

The FSE Resource Manager always allocates media and drives that are in the same library first to avoid time-intensive load/unload operations. If there are still jobs in the queue and no online media are available, then nearline media are selected.

Whenever possible, the current volume from an allocated medium is used. Otherwise, the volume with the highest use is selected to minimize the number of open volumes.

Diverse allocation parameters

There is only one Resource Manager for the whole FSE installation, and therefore it is responsible for all FSE partitions. The parameters needed for resource allocation are different for the system and each partition; they are defined in the systems policies and in the partitions policies.

Note that resource allocation requests from different partitions can have different drive and priority policies, and that one media pool cannot be assigned to multiple partitions. For more information on parameters for resource allocation, see [“System Allocation and Job Priority Policy”](#) on page 136 and [“Partition Allocation and Job Priority Policy”](#) on page 139.

Job priorities

Each resource request is put into a resource allocation queue with a certain initial priority. When a new job is put into the queue, the priorities of all existing jobs are recalculated; the priority of each job is modified according to the elapsed time and the phase of the job. The resource request is then processed according to its job priority. Each time a resource is freed, the queue is checked again to allocate the free resource. For more details, see [“Priority Calculation”](#) on page 142.

System Allocation and Job Priority Policy

The system allocation and job priority parameters are read by the Resource Manager upon its startup.

System Allocation and Job Priority Policy Parameters

The configuration parameters for the system along with their definition and impact are listed below.

Maximum number of drives for migration (**SystemMaxNumDrivesMigration**)

Definition:	Maximum number of drives that can be used for migration by all FSE partitions.
Impact:	The purpose of this parameter is to prevent migration jobs from using all the drives in the system. By limiting drives for migration, you ensure that drives remain available for other jobs (recall, admin, recovery...). You should carefully balance resources according to your environment.

Maximum number of drives for recall (**SystemMaxNumDrivesRecall**)

Definition:	Maximum number of drives that can be used for recall by all FSE partitions.
Impact:	The purpose of this parameter is to prevent recall jobs from using all drives in the system. By limiting drives for recall, you ensure that drives remain available for other jobs (migration, admin, recovery...). You should carefully balance resources according to your environment.

Maximum number of drives for administration jobs (**SystemMaxNumDrivesAdmin**)

Definition:	Maximum number of drives that can be used for administration jobs.
Impact:	The purpose of this parameter is to prevent administration jobs from using all drives in the system. By limiting drives for administration, you ensure that drives remain available for other jobs (migration, recall, recovery...). You should carefully balance resources according to your environment.

Maximum number of drives for recovery (**SystemMaxDrivesRecovery**)

Definition:	Maximum number of drives that can be used for recovery.
Impact:	The purpose of this parameter is to prevent recovery jobs from using all drives in the system. By limiting drives for recovery, you ensure that drives remain available for other jobs (migration, recall, admin...).

Resource allocation threshold (ResourceAllocationThreshold)

Definition:	When the Resource Manager tries to assign free resources, it will only consider jobs whose priority is not above the specified threshold of the first job in the queue (a job with top priority). When allocating resources, the Resource Manager takes into account only those jobs whose priority is no less than <i>ResourceAllocationThreshold</i> % lower than the top priority in the queue.
Impact:	<p>The purpose of this parameter is to consider only those queued allocation requests that have a higher priority than a percentage of the priority of the job with highest priority. This ensures some fairness to jobs with higher priorities that need many resources and cannot get all of them at once.</p> <p>For example, if the resource allocation threshold value is 90% and the highest priority is 5000, all jobs with priority higher than 4500 (90% of 5000) will be checked. If you specify 0%, all jobs in queue will be considered for allocation. If you specify 100%, only the first job in the queue will be considered for allocation.</p>

Time step (TimeStep)

Definition:	System-wide factor by which the priority of the queued jobs is increased due to the age of the job.
Impact:	<p>The purpose of this parameter is to leverage the elapsed time (the time since the job was started) and increase the priority of jobs that are waiting in the queue. The time factor is multiplied by the number of seconds since the job started, and by the <i>JobTypePriority</i> divided by 100. Note that if you set a value to 0 the job priorities will not get their priorities increased due to the age factor.</p> <p>For example, if the time-factor value is set to 10 and the job (Recall, Recall Priority = 80) has been waiting in the queue for 60 minutes, its priority will be increased by 28800 (10 * 60 min * 60 * 80/100). This will ensure that this job will have a higher priority than any new jobs added to the queue.</p>

Phase factor (PhaseFactor)

Definition:	System-wide factor by which the priority of each multi-phase job is increased.
Impact:	<p>The purpose of this parameter is to increase the priority of multiple phase jobs (migration to multiple copies or recall of files that span multiple media volumes) after some phases are already finished, such as migration phases with multiple copying, where each copy represents one phase. In case the multiple copying process was overtaken by another job with a higher priority, the files are still waiting to be migrated to the FSE media on the disk buffer thus occupying disk space. By setting a higher value, the priority is increased after each copy.</p>

Backup job priority (BackupJobPriority)

Definition:	System-wide priority level for backup jobs on a scale of 1 to 10000.
Impact:	The purpose of this parameter is to define the initial priority of backup jobs. You should adjust it according to the priorities defined in the FSE partition configuration for other jobs (migration, recall, recovery, maintenance).

System administration job priority (SystemAdminPriority)

Definition:	System-wide priority level for administration jobs (formatting and initializing) on a scale of 1 to 10000.
Impact:	The purpose of this parameter is to define the initial priority of administration jobs. You should adjust it according to the priorities defined in the FSE partition configuration for other jobs (migration, recall, recovery, maintenance).

Default system allocation and job priority policy

The following is the default system allocation and job priority policy.

```
[Allocation]
SystemMaxNumDrivesMigration = 20
SystemMaxNumDrivesRecall    = 20
SystemMaxNumDrivesAdmin     = 20
SystemMaxNumDrivesRecovery  = 20
ResourceAllocationThreshold = 50%
TimeStep                     = 2
PhaseFactor                  = 500

[Jobs]
SystemBackupPriority = 1000
SystemAdminPriority = 1000
```

Partition Allocation and Job Priority Policy

When a new Partition Manager starts (for a new partition), it reads and sets the parameters for the new partition. When a Partition Manager shuts down, the Resource Manager is notified in order to free up all reserved drives for the respective partition; consequently, its parameters are not considered during resource allocation.

Partition Allocation and Job Priority Policy Parameters

The configuration parameters for the partition along with their definition and impact are listed below.

Maximum number of drives (MaxNumDrives)

Definition:	Maximum number of all drives that can be used for all jobs by the partition.
Impact:	With this parameter you can prevent a single partition from using all the drives and not leaving any drives for other partitions and administrative jobs. You should carefully balance resources. Note that all other jobs are queued if the Partition Manager uses the maximum number of drives.

Maximum number of drives for migration (MaxNumDrivesMigration)

Definition:	Maximum number of drives that can be used for migration by the partition.
Impact:	The purpose of this parameter is to prevent the respective Partition Manager from using all the drives at once for migration jobs. You should carefully balance resources according to your environment.

Maximum number of drives for recall (MaxNumDrivesRecall)

Definition:	Maximum number of drives that can be used for recall by the partition.
Impact:	The purpose of this parameter is to prevent the respective Partition Manager from using all the drives at once for recall jobs. You should carefully balance resources according to your environment.

Maximum number of drives for recovery (MaxNumDrivesRecovery)

Definition:	Maximum number of drives that can be used for recovery by the partition.
Impact:	The purpose of this parameter is to prevent the respective Partition Manager from using all the drives at once for recovery jobs. You should carefully balance resources according to your environment.

Partition priority (PartitionPriority)

Definition:	The overall priority level for all jobs of one partition on a scale of 1 to 100.
Impact:	Partition priority is used together with job type priority (migration, recall, recovery, maintenance) to calculate the initial priority of the job. With this parameter you can differentiate overall priority of different FSE partitions.

Migration priority (MigrationPriority)

Definition:	The priority for migration jobs on a scale of 1 to 100.
Impact:	This priority is multiplied by the partition priority in order to calculate the initial priority of a migration job. Typically, the value of the migration priority is lower than the value of the recall priority (because recalls are more critical than migrations); however, you need to balance your job priorities carefully in order not to override other jobs. See "Priority Calculation" on page 142.

Recall priority (RecallPriority)

Definition:	The priority for recall jobs on a scale of 1 to 100.
Impact:	This priority is multiplied by the partition priority in order to calculate the initial priority of a recall job. Typically, the value for the recall priority is higher than the value for the migration priority; however, you need to balance your job priorities carefully in order not to override other jobs. See "Priority Calculation" on page 142.

Recovery priority (RecoveryPriority)

Definition:	The priority for recovery jobs on a scale of 1 to 100.
Impact:	This priority is multiplied by the partition priority in order to calculate the initial priority of a recovery job. See "Priority Calculation" on page 142.

Maintenance priority (MaintenancePriority)

Definition:	The priority for maintenance jobs on a scale of 1 to 100.
Impact:	This priority is multiplied by the partition priority in order to calculate the initial priority of a maintenance job. See "Priority Calculation" on page 142.

Default partition allocation and job priority policy

The following is the default allocation and job priority policy for an FSE partition that is provided with the FSE partition configuration template:

```
[Allocation]
MaxNumDrives           = 10
MaxNumDrivesMigration = 8
MaxNumDrivesRecall     = 8
MaxNumDrivesRecovery   = 8
PartitionPriority      = 50
MigrationPriority       = 20
RecallPriority          = 80
RecoveryPriority        = 100
MaintPriority           = 80
```

Priority Calculation

The resource allocation queue holds resource requests sorted and processed by their job priority. The priorities of all jobs in the queue are recalculated each time some changes to a job happen or resources are released or added to the system. These job priorities are dynamically increasing according to their age and phase of the job. A new job enters the queue with its initial priority; in case this priority is not high enough to be processed immediately, it is recalculated and increased over time thus enabling each job to be processed in good time.

Depending on job type, the priority formula differs for administration and backup jobs that have system-wide priority, and jobs that are associated with an FSE partition (migration, recall, recovery and maintenance). The priority calculation is based on parameters, set in the related partition policy or system policy, and a number of additional factors:

■ Phase

The phase of a job increases if it requires additional resources. This is relevant for migration jobs when the following is true:

- The phase is increased by 1 if a copy was made and other copies are still waiting for resources.
- The medium volume is full, therefore a new split for the file is needed.

It is also relevant for recall jobs when the following is true:

- The phase is increased by 1 when a split of a file is recalled which spans over multiple media volumes.

Other jobs do not have a phase.

- **TimeStep**
The value by which the priority is increased for each elapsed second of *ElapsedTime*.
- **JobType**
The specified priority for the relevant job type, for example MigrationPriority.
- **ElapsedTime**
The time in seconds since the job was started.
- **PhaseFactor**
The value by which the priority of a job is increased after each phase.

Administration and backup job priority calculation

For administration and backup jobs the priority is calculated by the following formula:

```
Priority = Admin or Backup Job Priority  
          + (ElapsedTime * TimeStep * JobTypePriority/10000)  
          + (Phase * PhaseFactor)
```

Migration, recall, recovery and maintenance job priority calculation

For migration, recall, recovery and maintenance jobs the priority is calculated by the following formula:

```
Priority = PartitionPriority * Job Type Priority  
          + (ElapsedTime * TimeStep * JobTypePriority/100)  
          + (Phase * PhaseFactor)
```

Note: You can manually increase or decrease a job priority using the `fsejob --priority` command.

Example of resource allocation

The following is an example of the resource allocation queue, as written in the log file.

```
cumulus:/root # /opt/fse/sbin/fsejob -l
JobId      Part   JobType  Prio Started          Status
20021119000043 Part01 Migration n/a 2002/11/19 08:02:05 Writing to Medium
20021119000044 Part01 Migration 1023 2002/11/19 08:02:40 Waiting for resources
20021119000045 Part01 Migration 1009 2002/11/19 08:03:15 Waiting for resources
20021119000046 Part02 Recall    4000 2002/11/19 08:03:38 Waiting for resources
20021119000047 Part01 Migration 1030 2002/11/19 08:02:40 Waiting for resources
20021119000048 Part02 Migration 1016 2002/11/19 08:03:15 Waiting for resources
20021119000049 Part01 Recall    n/a 2002/11/19 08:03:38 Reading from Medium
20021119000050 Part01 Migration n/a 2002/11/19 08:04:30 Writing Disk Buffer
```

Multiple Copying

FSE migrates data to FSE media to maintain an offline copy of the data on HSM file systems. Each file on an HSM file system has a corresponding copy on an FSE medium. The medium that stores such a copy belongs to the FSE media pool assigned to the respective FSE partition. Therefore all files on a particular HSM file system are migrated to the same media pool. In case of a disaster, this has the following advantages:

- A complete HSM file system can be restored from the FSE media, including the file system structure for the migrated file data.
- The HSM file system restore period is reduced since only the active (online, frequently used) file data need to be restored.

In order to enhance security and eliminate the risk of physical damage to your media, which can result in data loss, you can configure FSE to make redundant copies of the file data during migration jobs. To do this, you need to configure additional media pools, allocate additional media, and assign them to the particular HSM file system. All media pools must be configured with the same block size (*BlockSize* variable in the FSE media pool configuration file).

The number of FSE media pools assigned to the same FSE partition equals the number of copies the data from that FSE partition will have on FSE media. If you have more copies of file data on FSE media, you are able to recall your data from any available copy, thus benefiting on recall speed and data safety.

Depending on the configuration of your FSE installation and availability of required resources, you can have copies made in parallel or sequentially.

Parallel Copying

The number of FSE media pools assigned to an FSE partition determines the number of copies of the migrated data; you set up parallel copying by configuring at least two drives for the FSE partition (the *MaxNumDrivesMigration* variable in the FSE partition configuration file). Note that a media pool cannot be assigned to multiple partitions.

For best performance, there must be at least n drives free when the migration job with parallel copying starts, where n is the number of media pools assigned to the partition. Because of that, the number of parallel copies must not exceed the total number of drives that can be used for migration with the partition (*MaxNumDrivesMigration*), neither can it exceed the system-wide number of drives that can be used for migrations (*SystemMaxNumDrivesMigration* in the FSE system configuration file). Furthermore, the number of parallel copies must be balanced with the need for drives for other jobs running in FSE.

If there are not enough free drives available during a migration job with multiple copies, the copies that cannot be made in parallel will be made sequentially.

Sequential Copying

The number of configured FSE media pools determines the number of data copies for the corresponding FSE partition; if you have only one drive configured in FSE or if your FSE partition can use a maximum of one drive for migration at a time (*MaxNumDrivesMigration* variable in the FSE partition configuration file), your copies will be done sequentially. Also, if there are not enough free drives available during the migration job, copies that could not be made in parallel will be made sequentially. Note that a media pool cannot be assigned to multiple partitions.

Once a job is started it must be finished as soon as possible to free the disk buffer (the same disk buffer is used for creating each copy). The job priority increases over time to ensure that the job will allocate enough resources—drives and media—to complete. If some copies have already been made while others still need to be made and another job with a higher job priority was triggered in between, the priority of the migration job is recalculated (increased) using a special factor (*PhaseFactor* variable in the FSE system configuration file). Such handling gives the job a high enough priority to allocate the required resources as soon as possible.

HSM File System Access Modes

Normally, data on HSM file systems is only available when FSE processes are running, that is, when the FSE system is in a fully operational state. In certain situations, however, you may need access to the data even though the FSE system cannot be started up for some reason. The Limited Access Mode (LAM) feature of the FSE system allows this by enabling read-only access to directories and files on the mounted HSM file system. While in the read-only mode, the HSM file system is not controlled by Hierarchical Storage Manager.

Note: Limited Access Mode is available only on supported Windows operating systems.

Limited Access Mode (LAM) and Full Access Mode (FAM)

A properly configured HSM file system on a Windows FSE client, from which NTFS has been detached using the `fse --dismount-ntfs` can be mounted implicitly or explicitly. The implicit mount is triggered automatically after an object on the HSM file system is accessed for the first time. The explicit mount is executed manually invoking the `fse --mount` command for the corresponding disk volume, and can be performed only while FSE processes are running.

Once mounted, each HSM file system remains mounted all the time, even after the FSE client host is restarted. A mounted HSM file system, however, can operate in Limited Access Mode (LAM) or Full Access Mode (FAM). The operational mode depends on whether the FSE processes are running and whether an FSE administrator decided to switch between the two modes manually.

An HSM file system operates in LAM:

- before the FSE system has been started up
- after the FSE system has been shut down
- while the FSE system is fully operational, after the FSE administrator has manually switched from FAM to LAM using the FSE command-line interface.

An HSM file system operates in FAM:

- after the FSE system has been started up
- while the FSE system is fully operational, after the FSE administrator has manually switched from LAM to FAM using FSE command-line interface.

In Limited Access Mode, FSE users and third-party applications can only browse the HSM file system directory tree and access files that are still online. They cannot modify existing data, add new directories and files, or recall offline files. If an application tried to access a file while the HSM file system is operating in LAM, it receives an error. If LAM is manually set for an HSM file system while all FSE processes are running, the HSM process continues to trigger migrations, releases and deletions on this HSM file system, according to the migration, release, and deletion policies that are configured for the appropriate FSE partition.

In contrast, Full Access Mode does not constrain FSE users in any way. All data movement processes are enabled and you can use all the features that are generally available for HSM file systems in the FSE system.

Note: LAM and FAM are compatible with both regular and WORM FSE partitions. When an HSM file system is operating in FAM, WORM is fully supported, while when operating in LAM, the LAM restrictions prevail over the configured WORM policies.

For information on how to switch manually between the HSM file system access modes, see [“Switching the Access Mode for an HSM File System”](#) on page 154.

Monitoring and Maintaining FSE

5

The procedures in this chapter provide instructions for:

- [About Monitoring and Maintaining in FSE](#), page 149
- [Monitoring the FSE Installation](#), page 149
- [Managing FSE Partitions](#), page 150
- [Monitoring FSE Jobs](#), page 155
- [Checking the Consistency of the File System Catalog](#), page 157
- [Recalling Older Generations of a File](#), page 163
- [Viewing Logs](#), page 166
- [Retrieving Detailed Information](#), page 178

About Monitoring and Maintaining in FSE

You can use the FSE command-line interface to monitor and manage every aspect of the FSE installation, for example, jobs, partitions, and resources such as libraries, drives, and media.

For general information on FSE commands, see “[About FSE Commands](#)” on page 38. You can find specific options for each command by running it with the `--help` option. Note that the FSE command-line reference is also available on Linux as UNIX man pages and on Windows as HTML help.

Monitoring the FSE Installation

The `fssystem` command is used to manage the FSE system configuration and also to monitor the current FSE system status. You can view the status of all configured FSE partitions by running the `fssystem --status` command.

As shown in the example below, the output is the following:

- number of partitions in the FSE installation

- number of currently running administration jobs
- partition name, current status, the number of running jobs, host name, and mount point of the owning HSM file system for each partition. The partition status can be one of: disabled, stopped, running, mounted, unknown.

Output of the `fssystem --status` command on Linux:

```
# fssystem --status
# of Partitions: 2
# of Admin Jobs: 0

Partition      Status      #Jobs Host Mountpoint
video_captures mounted     2 abakus.company.com /fsefs/video_captures
mail_archive   mounted     2 abakus.company.com /fsefs /mail_archive
```

Output of the `fssystem --status` command on Windows:

```
C:\Program Files\Hewlett-Packard\FSE\bin>fssystem --status
# of Partitions: 2
# of Admin Jobs: 1

Partition Status #Jobs Host Mountpoint
part1 ready 0 testa.company.com <not mounted>
part2 mounted 0 testa.company.com I:\fse2\part2

C:\Program Files\Hewlett-Packard\FSE\bin>
```

Managing FSE Partitions

An FSE system can be configured to include several FSE partitions. FSE partitions can be located on consolidated FSE systems and on external FSE clients. The `fsepartition` command helps manage FSE partitions in such environments. Apart from partition configuration tasks, the FSE administrator can use this command to monitor the status of an FSE partition, temporarily disable or permanently remove an FSE partition, and retrieve detailed FSE partition information. You can also use the `fse` command to switch between the two HSM file system access modes available on Windows systems.

Monitoring FSE Partition Status

You can check the partition status by running the `fsepartition --status` command. This will give you an overview of the status of the partition and assigned media pools.

As shown in the following example, the output is the following:

- the current partition's status, which can be one of: disabled, stopped, ready, mounted, unknown
- the host and mount point of the containing HSM file system
- the number of migration, recall, maintenance, and recovery jobs running on the partition
- capacity information, usage information, and medium volum statistics for each media pool assigned to the partition.

Example output of the `fsepartition --status` command on Linux:

```
# fsepartition --status video_captures
Partition:          video_captures
Status:            mounted
Host:              host.domain.com
Mountpoint:        /fse/video_captures
# of Migration Jobs:          4
# of Recall Jobs:            1
# of Maintenance Jobs:       0
# of Recovery Jobs:          0

Pool  Size[MB] Avail[MB] Used[%] SysSize[MB] SysAvail[MB]
SysUsed[%] #Good
pool1 194598  177945   8     1018     997         2
2
pool2 194598  177824   8     1018     916        10
2
#Unreliable #Unusable #Uninit
0           0         0
0           0         0
```

Example output of the `fsepartition --status` command on Windows:

```
C:\Program Files\Hewlett-Packard\FSE\bin>fsepartition --status
part2
Partition:          part2
Status:            mounted
Host:              testa.company.com
Mountpoint:        I:\fse2\part2
# of Migration Jobs:          0
# of Recall Jobs:            0
# of Maintenance Jobs:       0
```

```
# of Recovery Jobs:                                0

Pool      Size[MB] Avail[MB] Used[%] SysSize[MB] SysAvail[MB]
SysUsed[%] #Good  pooldbl 49152   46632    6      0
0          0      1
#Unreliable #Unusable #Uninit
0          0      0
C:\Program Files\Hewlett-Packard\FSE\bin>fsepartition --status
part2
```

Disabling an FSE Partition

To disable an FSE partition, proceed as follows:

1. Ensure that the running processes do not have open handles on the HSM file system.
2. On the FSE client system, unmount the belonging HSM file system.

Linux specific

From the `/etc/fstab` file, determine the mount point of the HSM file system, and invoke the following command:

```
# umount Device
```

Windows specific

Using the `fsepartition --show` and `mountvol` commands determine the corresponding volume name and invoke the following command:

```
C:\>fse --umount MountPoint VolumeName
```

The command terminates the corresponding HSM process and switches the HSM FS filter operation for managing accesses to the FSE partition to Limited Access Mode (LAM).

3. On the FSE server system, run the following command:

```
C:\>fsepartition --disable PartitionName
```

This stops the Partition Manager (PM) that has been running for the FSE partition.

Enabling an FSE Partition

To enable an FSE partition, perform the following:

1. On the FSE server system, start the appropriate Partition Manager (PM):

```
fsepartition --enable PartitionName
```

2. On the FSE client system, mount the belonging HSM file system.

Linux specific

From the `/etc/fstab` file, determine the mount point of the HSM file system, and invoke the following command:

```
# mount MountPoint
```

Windows specific

Invoke the following command:

```
C:\>fse --mount MountPoint VolumeName
```

The command starts the appropriate HSM process and switches the HSM FS filter operation for managing accesses to the FSE partition to Full Access Mode (FAM).

Removing an FSE Partition



Caution: When an FSE partition is removed from the FSE system, all files that have been released from the corresponding HSM file system (offline files) will be lost.

To remove an FSE partition, proceed as follows:

1. On the FSE client system, unmount the appropriate HSM file system.

Linux specific

From the local `/etc/fstab` file, determine the mount point of the HSM file system, and use the following command:

```
# umount MountPoint
```

Windows specific

Using the `fsepartition --show` and `mountvol` commands determine the appropriate volume name and use the following command:

```
C:\>fse --umount MountPoint VolumeName
```

This stops the corresponding HSM process and switches the HSM FS filter operation for managing accesses to the FSE partition to Limited Access Mode (LAM). The partition will remain accessible in LAM until the disk volume is reformatted or the client system is restarted.

2. On the FSE server system, use the following command:

```
fsepartition --remove PartitionName
```

Retrieving Detailed Information about Configured FSE Partitions

To retrieve detailed information about the configured FSE partition, use the following command:

```
# fsepartition --list --detail
```

An example of this command's output is shown in [“Excerpt from an example output of the fsepartition --list --detail command”](#) on page 186.

Switching the Access Mode for an HSM File System

The `fse` command provides special options that allow you to manually switch between Limited Access Mode (LAM) and Full Access Mode (FAM) for a particular HSM file system on a Windows FSE client.

Switching to Limited Access Mode

Prerequisites

- FSE processes on the FSE server and the FSE client must be running.
- The HSM file system must be operating in FAM.

To switch the HSM file system access to LAM, use the following command. Use the name of the corresponding disk volume:

```
C:\>fse --limit-access <VolumeName>
```

Switching to Full Access Mode

Prerequisites

- FSE processes on the FSE server and the FSE client must be running.
- The HSM file system must be operating in LAM.

To switch the HSM file system access to FAM, use the following command. Use the name of the corresponding disk volume:

```
C:\>fse --full-access <VolumeName>
```

For more information on HSM file system access modes, see [“HSM File System Access Modes”](#) on page 146.

Monitoring FSE Jobs

The `fsejob` command is used to view the details of running FSE jobs, change their priorities and abort them.

You can list FSE jobs by running the `fsejob --list --detail` command to check running jobs or to get a particular job ID.

As shown in the example, the output is the following:

- the job ID
- the corresponding partition name
- the job type
- the job priority
- the time when the job was started
- the current status of the job

You can use the `--list` option followed by the job type command, for example `--migration`, to narrow your search and view only jobs of a particular type.

Example output of the `fsejob --list` command

```
# fsejob --list
JobId          Partition      Type           Prio  Started
Status
20021202000082 video_captures Recall         8478  2002/12/02
10:22:57  Waiting for resources
20021202000083 video_captures Migration      7436  2002/12/02
10:23:07  Writing DiskBuffer
20021202000084 scanned_img    Migration      4439  2002/12/02
10:24:34  Writing to Medium
```

Example output of the `fsejob --list --detail` command

```
# fsejob --list --detail
JobID:         20021202000082
Partition:     video_captures
Type:          Recall
Priority:       8478
Started:       2002/12/02 10:22:57
Status:        Waiting for resources

JobID:         20021202000083
Partition:     video_captures
```

```
Type:      Migration
Priority:   7436
Started:   2002/12/02 10:23:07
Status:    Writing DiskBuffer
```

```
JobID:     20021202000084
Partition: scanned_img
Type:      Migration
Priority:   4439
Started:   2002/12/02 10:24:34
Status:    Writing to Medium
```

The retrieved data is the same for both `fsejob --list` and the same with the `--detail` option, but in the second case, the data is displayed using a different layout so that the output can be parsed by other commands and scripts.

Another example of the `fsejob --list --detail` command output is shown in “[Excerpt from an example output of the `fsejob --list --detail` command](#)” on page 188.

Once you know the job ID, you can narrow your search and view the status of the selected FSE job by using the `--status` option followed by the job ID.

As shown in the example, the output is the following:

- the job ID
- the partition name
- the job type
- the job priority
- the time when the job was started

Example output of the `fsejob --status` command

```
# fsejob --status 20021202000173
JobId:      20021202000173
Partition:  part2
Type:       Migration
Priority:
Started:    2002/12/02 14:32:54
```

Copy	Phase	Drive	Medium	Volume	Status
1	1	n/a	n/a	n/a	Writing Disk Buffer

Checking the Consistency of the File System Catalog

The File System Catalog (FSC) consists of a Data Location Catalog and a Name Space Catalog. It is related to a single partition, and located in the `/var/opt/fse/part/<PartitionName>/fsc` directory on Linux and in `%InstallPath%\var\part\<PartitionName>\fsc` on Windows.

Since the File System Catalog functions as a link between files on the HSM file system and their storage positions on media, and contains important data for logical verification of media and for HSM file system recovery, it has to be consistent to enable proper FSE installation operation. It is recommended that you check the consistency of the File System Catalog regularly.

Consistency check of the File System Catalog is a maintenance job, running on the specified FSE partition. It compares file data recorded in the File System Catalog either with actual files on the corresponding HSM file system, or with Fast Recovery Information (FRI) on the corresponding FSE media. Any inconsistencies are recorded in the FSE check log.

Consistency Check of FSC vs. HSM File System

The consistency check of the File System Catalog (including the Name Space Catalog) against the HSM file system matches FSC and NSC information on files, such as their size, file types, attributes, and modification time with the status on the HSM file system. In case of mismatch, the results are written to the FSE check log as follows:

- Linux: `/var/opt/fse/log/check_hsmfs_fsc_<PartitionName>.log`
- Windows:
`%InstallPath%\var\log\check_hsmfs_fsc_<PartitionName>.log`

To perform the FSC consistency check, use the following command:

```
fsecheck --fsc-hsmfs PartitionName
```

Inconsistency types

Reported inconsistencies in the FSE check log can be any of the following:

- missing in FSC

A file is recorded as missing in FSC when the file ID on the HSM file system exists, but is not recorded in FSC. Probably the file has not yet been successfully migrated, therefore you need to migrate it manually by running the command: `fsefile --migrate FileName`.

■ suspicious

A file is recorded as suspicious when it is created on the HSM file system but does not have a file ID. It is possible that the FSE installation is unaware of its existence. This can happen in situations, such as:

- The disk was full when the file was created and the FSE attributes could not be written.
- The file was created before the Ext3/NTFS file system was mounted as an HSM file system (for example, the `lost+found` directory).
- A power failure occurred and the file ID was not written or the Hierarchical Storage Manager did not get the event request.

■ offline file is prolonged or not shortened

Files that have not been released properly or which have not been fully recalled due to an error have the Prolonged status. In such cases, the file is in an offline state but still occupies storage space. To resolve the problem, first recall the file by accessing it, and then release it with the `fsefile --release` command.

■ mismatch report, such as:

- MigrationID mismatch
- Owner mismatch
- Name mismatch
- Not all attributes in NSC
- Size mismatch

Example FSC vs. HSM FS consistency check results on Linux:

```
[2003/12/03 14:49:45] Started FSC vs. HSMFS check. Partition
cosmos
[2003/12/03 15:06:02] 578684 0
/mnt/fsefs_01/A9/linux-2.4.19-ea/NewFile
Missing in FSC.
[2003/12/03 15:06:49] 154342 273932147200
/mnt/fsefs_01/A9/linux-2.4.19-ea/COPYING
DUMP:                FSC                HSMFS Notes
MigID                273932147200        273932147200
FileSize             18691                32930 MISMATCH
FileName             COPYING              COPYING
OwnerID              154336               154336
```



```
[2003/12/03 15:09:52] 77174 273931007488
/mnt/fsefs_01/A4/linux-2.4.19-ea/REPORTING-BUGS
DUMP:                FSC                HSMFS Notes
MigID                273931007488          273931007488
FileSize             2815                0 MISMATCH
FileName             REPORTING-BUGS    REPORTING-BUGS
OwnerID              57896                57896

[2003/12/03 15:11:51] 578672 273989682176
/mnt/fsefs_01/A1/linux-2.4.19-ea/nekaj
DUMP: FSC HSMFS Notes
MigID 273989682176 273989682176
FileSize 6 0 MISMATCH
FileName nekaj nekaj
OwnerID 31 31

[2003/12/03 15:12:29] Finished (checked: 578684, missing: 4,
suspicious: 0, prolonged:0).

[2003/12/03 15:12:29] Finished (JobID: 20031203000011).
```

Example FSC vs. HSM FS consistency check results on Windows:

```
C:\Program Files\Hewlett-Packard\FSE\bin>fsecheck --fsc-hsmfs
FS04-CLIENT01
[2004/03/23 08:15:09] Started. (JobID: 20040323000011).
[2004/03/23 08:13:28] Started FSC vs. HSMFS check. Partition
FS04-CLIENT01
[2004/03/23 08:13:55] 64732 276482757120
i:\fse\part1\soft02\MCC
Installation\Docs\MenD°punte ERM.xls
DUMP:                FSC                HSMFS Notes
MigID                276482757120          276482757120
FileSize             16896                16896
FileName             MenD°punte ERM.xlsMenD°punte ERM.xls MISMATCH
OwnerID              64728                64728

[2004/03/23 08:15:27] 39339 276478525440
i:\fse\part1\soft01\MCC
Installation\Docs\MenD°punte ERM.xls
DUMP:                FSC                HSMFS Notes
MigID                276478525440          276478525440
FileSize             16896                16896
FileName             MenD°punte ERM.xlsMenD°punte ERM.xls MISMATCH
OwnerID              39335                39335

[2004/03/23 08:16:15] Finished (checked: 57036, missing: 2,
suspicious: 0, prolonged:0).
```

```
[2004/03/23 08:17:55] Finished (JobID: 20040323000011).
```

For inconsistent files, the full path is recorded in the FSE check log.

Note: The FSE installation is able to trace a “suspicious” file and put it under FSE control in time, but to solve its status immediately, you can use the `fsefile --migrate` command to put it into the dirty files list. This way the file is recognized and its attributes are preserved.

Consistency Check of FSC vs. FSE Media

The consistency check of the File System Catalog (including the Name Space Catalog) against FSE media matches FSC and NSC information on files with Fast Recovery Information (FRI) on FSE media. In case of a mismatch, the results are written to the FSE check log as follows:

- Linux: `/var/opt/fse/log/check_fsc_media_<PartitionName>.log`
- Windows:
`%InstallPath%\var\log\check_fsc_media_<PartitionName>.log`

To perform the FSC consistency check, use the following command:

```
fsecheck --fsc-media [--autocorrect] PartitionName
```

Note: Depending on the inconsistency type, for example, if a reference to a file on the FSE medium is missing in the File System Catalog, or if there are inconsistencies in NSC, you can use the automatic correction option `--autocorrect` to update FSC (including NSC) with missing information.

In case of other inconsistency types, you need to perform a FSC recovery. For details, see “[File System Catalog Recovery](#)” on page 213.

Inconsistency types

Reported inconsistencies in the FSE check log can be any of the following:

- missing in FSC
A file is recorded as missing in FSC when the file ID on the HSM file system exists but is not recorded in FSC. Probably the file has not yet been successfully migrated, therefore you need to migrate it manually by running the command: `fsefile --migrate FileName.`
- missing on media

- no NSC
- mismatch report, such as:
 - Wrong file size
 - Wrong file name
 - Wrong owner-ID
 - ADS name
 - Not last NSC

Example FSC vs. FSE media consistency check results:

```
[2003/12/03 15:28:23] Started FSC vs. media check. Partition
cosmos
[2003/12/03 15:28:23] Preparing FSC for check.
[2003/12/03 15:28:36] Preparing of FSC finished.
[2003/12/03 15:28:37] Reading FRI from medium: dm001.
[2003/12/03 15:28:38] Reading FRI from medium: dm002.
[2003/12/03 15:28:49] Started FSC chk vol num: 1, medium: dm002
[2003/12/03 15:29:01] Finished FSC chk vol num: 1 (checked:
72893, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:29:01] Started FSC chk vol num: 2, medium: dm002
[2003/12/03 15:29:10] Finished FSC chk vol num: 2 (checked:
73907, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:29:10] Started FSC chk vol num: 3, medium: dm002
[2003/12/03 15:29:18] Finished FSC chk vol num: 3 (checked:
65380, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:29:18] Started FSC chk vol num: 4, medium: dm002
[2003/12/03 15:29:18] 154340 274037495040 MAINTAINERS
Missing in FSC.
DUMP: FRI
MigID 274037495040
MediaKey 4
MedVolNum 4
MedVolOffset 472
FileType 1
FileSize 41643
MigSize 41643
MigFlags HD
SplitFlags L
```

```
SplitOffset 0
SplitSize 41643
FileName MAINTAINERS
OldFileName
OwnerID 154336
OldOwnerID 0
[2003/12/03 15:29:18] 578684 274037495040 NewFile
Missing in FSC.
DUMP: FRI
MigID 274037495040
MediaKey 4
MedVolNum 4
MedVolOffset 473
FileType 1
FileSize 8
MigSize 8
MigFlags HD
SplitFlags L
SplitOffset 0
SplitSize 8
FileName NewFile
OldFileName
OwnerID 154336
OldOwnerID 0
[2003/12/03 15:29:19] Finished FSC chk vol num: 4 (checked: 15,
missing in FSC: 2, missing on media: 0, mismatch: 0, no NSC: 0)
[2003/12/03 15:29:19] Finished FSC chk MED: dm002 (checked:
212195, missing in FSC: 2, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:29:19] Started FSC chk vol num: 1, medium: dm001
[2003/12/03 15:29:29] Finished FSC chk vol num: 1 (checked:
77182, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:29:29] Started FSC chk vol num: 2, medium: dm001
[2003/12/03 15:29:38] Finished FSC chk vol num: 2 (checked:
72400, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:29:38] Started FSC chk vol num: 3, medium: dm001
[2003/12/03 15:29:47] Finished FSC chk vol num: 3 (checked:
73599, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:29:47] Started FSC chk vol num: 4, medium: dm001
```

```
[2003/12/03 15:29:55] Finished FSC chk vol num: 4 (checked:
67326, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:29:55] Started FSC chk vol num: 5, medium: dm001
[2003/12/03 15:30:03] Finished FSC chk vol num: 5 (checked:
75999, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:30:03] Finished FSC chk MED: dm001 (checked:
366506, missing in FSC: 0, missing on media: 0, mismatch: 0, no
NSC: 0)

[2003/12/03 15:30:03] FINISHED FSC chk grand TOTAL : (checked:
578701, missing in FSC: 2, missing on media: 0, mismatch: 0, no
NSC: 0)
[2003/12/03 15:30:03] Finished (JobID: 20031203000012).
```

Recalling Older Generations of a File

An administrator can recall any older generation of a migrated file using the `fsefile` command executed on the FSE client. The `fsefile` command is also used to show file history with information about older migrated generations of a file on an HSM file system. For further information about using the `fsefile` command, see “[fsefile](#)” on page 283.

Note: The `fsefile` command can be used only on files which are in a local HSM file system.

The procedure for recalling an older generation of a migrated file is the same on Linux and Windows systems. The main difference in the examples that follow is in the pathnames used.

Recalling an older generation of a migrated file can take some time. The command, however, starts a recall job and returns quickly without a message if the job has been started successfully.

Note: Permissions, ownership, and dates change on recall. Creation time, modification time, and access time are set to the current time when the file has been recalled. The ownership is set to root.

The following is an example of the `fsefile --history` command and its output.

Example output of the `fsefile --history` command (on Linux):

```
fsefile --history /fse/part01/20030328/root/*
FileID MigID Time Size[KB] FileName
1023065 268507808768 2003/03/28 14:37:08 7276
/fse/part01/20030328/root/20021201-1.log
1023126 268507808768 2003/03/28 14:37:08 390
/fse/part01/20030328/root/formatjob.sh
1023127 268507808768 2003/03/28 14:37:08 9442
/fse/part01/20030328/root/gmon.out
1023249 268530735104 2003/03/29 15:29:44 5512
/fse/part01/20030328/root/hsmdb.out
1023249 268509426688 2003/03/28 16:22:28 4684
/fse/part01/20030328/root/hsmdb.out
1023249 268509288448 2003/03/28 16:13:28 2014
/fse/part01/20030328/root/hsmdb.out
1023128 268507808768 2003/03/28 14:37:08 4096
/fse/part01/20030328/root/img
1023193 268507808768 2003/03/28 14:37:08 120994
/fse/part01/20030328/root/fse-la-m
1023194 268507808768 2003/03/28 14:37:08 192142
/fse/part01/20030328/root/fse-la-s
1023195 268507808768 2003/03/28 14:37:08 0
/fse/part01/20030328/root/fse-lib.cfg
1023196 268507808768 2003/03/28 14:37:08 1425396
/fse/part01/20030328/root/fse-mif
1023197 268507808768 2003/03/28 14:37:08 718971
/fse/part01/20030328/root/fse-rm
1023198 268507808768 2003/03/28 14:37:08 262
/fse/part01/20030328/root/fse_drivestatus.sh
1023199 268507808768 2003/03/28 14:37:08 1343
/fse/part01/20030328/root/lib-ait2-output.log
1023200 268507808768 2003/03/28 14:37:08 602
/fse/part01/20030328/root/lib-ait3-output.log
1023201 268507808768 2003/03/28 14:37:08 834
/fse/part01/20030328/root/lib-ait3.cfg
1023202 268507808768 2003/03/28 14:37:08 1200325
/fse/part01/20030328/root/libfseipc.so
1023203 268507808768 2003/03/28 14:37:08 4096
/fse/part01/20030328/root/logs
1023212 268507808768 2003/03/28 14:37:08 4096
/fse/part01/20030328/root/manpages
```

```
1023231 268507808768 2003/03/28 14:37:08 169316
/fse/part01/20030328/root/mbox
```

In the command output, the lines with the file ID 1023249 show three migrated generations of the `hsmdb.out` file. These entries have different migration IDs and different time attributes.

You can recall a file by specifying either a date (the `--date` option) or a file migration ID (the `--migid` option). The examples below show how to recall a file by date and by file migration ID.

Example of a recall by date (on Linux):

```
fsefile --recall /fse/part01/20030328/root/hsmdb.out
--date "2003/03/28 16:22:28"
--into /tmp
```

Example of a recall by file migration ID (on Windows):

```
C:\Program Files\Hewlett-Packard\FSE\bin>fsefile --recall
i:\fse\part01\20030328\mylittledoc.doc
--migid 268509426688
--into D:\tmp
```

Each file must be recalled separately. You define a recall by specifying a filename or by using the argument combination `--id PartitionName FileID`, as shown in the examples below.

A recall on a deleted file (undelete operation) can be invoked only by using the file ID. In this case, the undeleted file will get the same name as it had when it was migrated for the last time.

Example of a recall by file ID instead of file name (on Linux):

```
# fsefile --recall --id FSEPartition010 30600
--date "2002/12/10 15:36:50"
--into /tmp/fse_deleted_files
```

Example of a recall by file ID instead of file name (on Windows):

```
C:\Program Files\Hewlett-Packard\FSE\bin>fsefile --recall --id
FSEPartition021 17800
--date "2002/12/18 11:30:20"
--into D:\tmp\fse_deleted_files
```

Viewing Logs

FSE log files provide a variety of information on FSE events, such as error and status messages, details on FSE job execution, and resource allocation. You can use them to monitor FSE operations, for statistical purposes as well as for backup tracing.

The FSE log files are the following:

- **FSE event log** (`fse.log`)

This log records relevant information on FSE-process events that occur during FSE operation.

- **FSE error log** (`error.log`)

This log records error messages generated by FSE processes.

- **FSE check log**

(`check_hsmfs_fsc_<PartitionName>`,
`check_medium_fsc_<PartitionName>`)

This log records inconsistencies found by the maintenance job that performs a File System Catalog consistency check with the HSM file system or with FSE media, for the specified FSE partition.

- **FSE backup log** (`backup.log`)

This log records information relevant for tracing the events that occur during a backup of the FSE installation.

- **HSM file system log** (`hsmfs_<PartitionName>.log`)

This log records all events that occur on an HSM file system: file creation, rename, move, deletion, and so on. Each configured FSE partition has its own HSM file system log. The log file is located on the FSE client that hosts the FSE partition.

For information on the locations of FSE log files, see “[Directory Layout](#)” on page 249.

The FSE Event Log

The FSE event log records specific information on events occurring in FSE processes during the operation of FSE. It provides summary reporting on routine and non-routine events. These events are, for example, start-up of the FSE agents and daemons/services, start-up and completion of FSE jobs, reconfiguration of FSE resources, operations with FSE media and so on. This helps you to determine

the load on the FSE installation, the frequency and the duration of FSE jobs, backup creation interval, disk usage (watermarks), and so on. All these statistics can be used for fine-tuning the FSE installation.

Logging FSE media operations

Some operations on FSE tape media may be rather long-lasting. In order to facilitate tracking, monitoring, and debugging of operations like migrations, recalls, and media formatting and initialization, these operations are also logged.

The following operations on media are logged in the FSE event log:

- Searching for EOD
- Searching to the beginning of an FSE medium volume
- Changing the current medium volume
- Writing a filemark
- Searching to a filemark
- Searching to a particular block on tape

Example FSE event log on Linux:

```
=====
FSE Event log file

  Hostname (IP): gtl-lnx.company.com (10.65.1.58), official:
gtl-lnx.company.com (10.65.1.58)
  OS           : Linux (2.4.19-4GB,#1 Mon Aug 4 23:23:04 UTC 2003)
on i686
  Create time  : 2004/06/17 13:55:07
=====

[2004/06/17 13:55:07, cumulus, 009779, SVC, , 0, ] Started
(3.0.145 "Beta_02").
[2004/06/17 13:55:07, cumulus, 009787, RMG, , 0, ] Started
(3.0.145 "Beta_02").
[2004/06/17 13:55:07, cumulus, 009799, MIF, , 0, ] Started
(3.0.145 "Beta_02").
[2004/06/17 13:55:08, cumulus, 009817, LAS, lib01, 0, ] Started
(3.0.145 "Beta_02").
[2004/06/17 13:55:08, cumulus, 010092, FSE, , 0, ] Started
(3.0.145 "Beta_02").
[2004/06/17 13:55:09, cumulus, 009817, LAS, lib01, 0, ]
Inventory
```

```
[2004/06/17 13:55:09, cumulus, 009817, LAS, lib01, 0, ] Library
reconfigured.
[2004/06/17 13:55:26, cumulus, 009817, LAS, lib01, 0, ]
Inventory
[2004/06/17 13:55:26, cumulus, 009799, MIF:DRIVE, , 0, drv01]
Added new.
[2004/06/17 13:57:25, cumulus, 009799, MIF:DRIVE, , 0, drv01]
Updated (rev. 2).
[2004/06/17 13:57:37, cumulus, 009817, LAS, lib01, 0, ]
Inventory
[2004/06/17 13:57:37, cumulus, 009799, MIF:DRIVE, , 0, drv02]
Added new.
[2004/06/17 13:58:34, cumulus, 009799, MIF:POOL, , 0, pool01]
Added new.
[2004/06/17 14:03:47, cumulus, 009799, MIF:POOL, , 0, pool02]
Added new.
[2004/06/17 14:11:31, cumulus, 010321, PMG, part01, 0, ] Started
(3.0.145 "Beta_02").
[2004/06/17 14:11:32, cumulus, 009799, MIF:PARTITION, , 0,
part01] Added new.
[2004/06/17 14:16:43, cumulus, 009799, MIF:MEDIUM, , 0, 000006]
Medium 000006 added to pool pool01.
[2004/06/17 14:16:47, cumulus, 009799, MIF:MEDIUM, , 0, 000007]
Medium 000007 added to pool pool01.
[2004/06/17 14:16:50, cumulus, 009799, MIF:MEDIUM, , 0, 000008]
Medium 000008 added to pool pool01.
[2004/06/17 14:16:53, cumulus, 009799, MIF:MEDIUM, , 0, 000009]
Medium 000009 added to pool pool01.
[2004/06/17 14:16:59, cumulus, 009799, MIF:MEDIUM, , 0, 000010]
Medium 000010 added to pool pool01.
[2004/06/17 14:17:56, cumulus, 009799, MIF:ADMIN, ,
20040617000001, 000006] Started.
[2004/06/17 14:17:56, cumulus, 009799, MIF:ADMIN, ,
20040617000001, 000006] ERROR: Problems with Back End Agent
(6004: Caught Corba exception.
'CORBA::COMM_FAILURE#IDL:omg.org/CORBA/COMM_FAILURE:1.0').
[2004/06/17 14:17:56, cumulus, 009799, MIF:ADMIN, ,
20040617000001, 000006] Finished.
[2004/06/17 14:17:57, cumulus, 009799, MIF:ADMIN, ,
20040617000002, 000007] Started.
[2004/06/17 14:18:52, cumulus, 009817, LAS, lib01,
20040617000002, 000046] Move: D=2 -> S=9
[2004/06/17 14:18:57, cumulus, 009817, LAS, lib01,
20040617000002, 000007] Move: D=2 <- S=7
[2004/06/17 14:19:10, cumulus, 010445, BEA, /dev/sg2,
20040617000002, 000007] Vol 4: Changing to volume 3...
```

```
[2004/06/17 14:20:06, cumulus, 010445, BEA, /dev/sg2,
20040617000002, 000007] Vol 3: Rewind...
[2004/06/17 14:20:10, cumulus, 009799, MIF:ADMIN, ,
20040617000002, ] WARNING: Medium 000007, volume 3:
Initialization of HSM/ECMA/TAR/FSE NT volume not allowed. Use
force option.
[2004/06/17 14:20:10, cumulus, 010445, BEA, /dev/sg2,
20040617000002, 000007] Vol 3: Changing to volume 1...
[2004/06/17 14:20:57, cumulus, 010445, BEA, /dev/sg2,
20040617000002, 000007] Vol 1: Rewind...
[2004/06/17 14:21:01, cumulus, 009799, MIF:ADMIN, ,
20040617000002, ] WARNING: Medium 000007, volume 1:
Initialization of HSM/ECMA/TAR/FSE NT volume not allowed. Use
force option.
[2004/06/17 14:21:01, cumulus, 010445, BEA, /dev/sg2,
20040617000002, 000007] Vol 1: Changing to volume 2...
[2004/06/17 14:21:20, cumulus, 010445, BEA, /dev/sg2,
20040617000002, 000007] Vol 2: Rewind...
[2004/06/17 14:21:25, cumulus, 009799, MIF:ADMIN, ,
20040617000002, ] WARNING: Medium 000007, volume 2:
Initialization of HSM/ECMA/TAR/FSE NT volume not allowed. Use
force option.
[2004/06/17 14:21:25, cumulus, 010445, BEA, /dev/sg2,
20040617000002, 000007] Vol 2: Changing to volume 4...
[2004/06/17 14:21:48, cumulus, 009799, MIF:ADMIN, ,
20040617000003, 000008] Started.
[2004/06/17 14:21:48, cumulus, 009799, MIF:ADMIN, ,
20040617000004, 000009] Started.
[2004/06/17 14:21:48, cumulus, 009799, MIF:ADMIN, ,
20040617000005, 000010] Started.
[2004/06/17 14:22:24, cumulus, 010445, BEA, /dev/sg2,
20040617000002, 000007] Vol 4: Rewind...
[2004/06/17 14:22:29, cumulus, 009799, MIF:ADMIN, ,
20040617000002, ] WARNING: Medium 000007, volume 4:
Initialization of HSM/ECMA/TAR/FSE NT volume not allowed. Use
force option.
[2004/06/17 14:22:29, cumulus, 009799, MIF:ADMIN, ,
20040617000002, 000007] WARNING: Aborted.
[2004/06/17 14:22:29, cumulus, 009799, MIF:JOB, , ,
20040617000002, ] JOB 20040617000002 ABORTED.
[2004/06/17 14:22:41, cumulus, 009817, LAS, lib01,
20040617000003, 000006] Move: D=1 -> S=6
[2004/06/17 14:22:41, cumulus, 009799, MIF:ADMIN, ,
20040617000003, 000008] WARNING: Aborted.
[2004/06/17 14:22:41, cumulus, 009799, MIF:JOB, , ,
20040617000003, ] JOB 20040617000003 ABORTED.
```

```
[2004/06/17 14:23:08, cumulus, 009817, LAS, lib01,
20040617000004, 000007] Move: D=2 -> S=7
[2004/06/17 14:23:08, cumulus, 009799, MIF:ADMIN, ,
20040617000004, 000009] WARNING: Aborted.
[2004/06/17 14:23:08, cumulus, 009799, MIF:JOB , ,
20040617000004, ] JOB 20040617000004 ABORTED.
[2004/06/17 14:23:13, cumulus, 009817, LAS, lib01,
20040617000005, 000010] Move: D=1 <- S=10
[2004/06/17 14:23:13, cumulus, 009799, MIF:ADMIN, ,
20040617000005, 000010] WARNING: Aborted.
```

Example FSE event log on Windows:

```
=====
FSE Event log file

  Hostname (IP): gtl-host.company.com (10.81.1.45), official:
gtl-host.company.com (10.81.1.45)
  OS           : Microsoft Windows 2000 (5.0,Service Pack 4) on
  Create time  : 2004/07/06 15:15:10
=====

[2004/07/06 15:15:10, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040706000004, ] Started.
[2004/07/06 15:15:10, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040706000004, ] Finished.
[2004/07/06 15:20:08, gtl-host, 002260, HSM, c:\fsefs2\, 0, ]
Migration triggered by user (#files: 39).
[2004/07/06 15:20:08, gtl-host, 001980, PMG:MIGRATION, part2,
20040706000005, ] Started (#files: 39, size: 309 KB).
[2004/07/06 15:20:09, gtl-host, 002260, HSM, c:\fsefs2\,
20040706000005, ] Migration (#files: 39, size: 278 KB).
[2004/07/06 15:20:09, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY228L2) has no usable volumes for migration
[2004/07/06 15:20:09, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY228L2) has no usable volumes for migration
[2004/07/06 15:20:09, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY229L2) has no usable volumes for migration
[2004/07/06 15:27:54, gtl-host, 001704, MIF:ADMIN, ,
20040706000006, PFY228L2] Started.
[2004/07/06 15:27:54, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY228L2) has no usable volumes for migration
[2004/07/06 15:27:54, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY229L2) has no usable volumes for migration
```

```
[2004/07/06 15:27:55, gtl-host, 000560, BEA, id:3:0:1:0,
20040706000006, PFY228L2] Vol 1: Changing to volume 1...
[2004/07/06 15:27:55, gtl-host, 000560, BEA, id:3:0:1:0,
20040706000006, PFY228L2] Vol 1: Rewind...
[2004/07/06 15:27:56, gtl-host, 000560, BEA, id:3:0:1:0,
20040706000006, PFY228L2] Vol 1: Changing to volume 1...
[2004/07/06 15:27:56, gtl-host, 000560, BEA, id:3:0:1:0,
20040706000006, PFY228L2] Vol 1: Rewind...
[2004/07/06 15:28:02, gtl-host, 000560, BEA, id:3:0:1:0,
20040706000006, PFY228L2] Vol 1: Writing filemark(s) 1...
[2004/07/06 15:28:03, gtl-host, 000560, BEA, id:3:0:1:0,
20040706000006, PFY228L2] Vol 1: volume initialized.
[2004/07/06 15:28:04, gtl-host, 001704, MIF:ADMIN, ,
20040706000006, PFY228L2] Finished.
[2004/07/06 15:28:04, gtl-host, 002688, BEA, id:3:0:1:0,
20040706000005, PFY228L2] Vol 1: Changing to volume 1...
[2004/07/06 15:28:09, gtl-host, 002688, BEA, id:3:0:1:0,
20040706000005, PFY228L2] Vol 1: Rewind...
[2004/07/06 15:28:09, gtl-host, 002688, BEA, id:3:0:1:0,
20040706000005, PFY228L2] Vol 1: Seeking to EOD...
[2004/07/06 15:28:09, gtl-host, 002688, BEA, id:3:0:1:0,
20040706000005, PFY228L2] Vol 1: Writing filemark(s) 0...
[2004/07/06 15:28:15, gtl-host, 002688, BEA, id:3:0:1:0,
20040706000005, PFY228L2] Vol 1: Data written: 352 KB (57 KB/s).
[2004/07/06 15:28:15, gtl-host, 002688, BEA, id:3:0:1:0,
20040706000005, PFY228L2] Vol 1: Seeking to EOD...
[2004/07/06 15:28:15, gtl-host, 001980, PMG:MIGRATION, part2,
20040706000005, ] Finished.
[2004/07/06 15:28:28, gtl-host, 002260, HSM, c:\fsefs2\, 0, ]
Forced release by user (#files: 39, size: 270 KB).
[2004/07/06 15:44:26, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040706000007, ] Started.
[2004/07/06 15:44:26, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040706000007, ] Finished.
[2004/07/08 15:49:42, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040708000001, ] Started.
[2004/07/08 15:49:42, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040708000001, ] Finished.
[2004/07/08 16:04:09, gtl-host, 001704, MIF:PARTITION, , 0,
part2] Updated (rev. 3).
[2004/07/08 16:04:20, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040708000002, ] Started.
[2004/07/08 16:04:20, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040708000002, ] Finished.
[2004/07/08 16:06:06, gtl-host, 001704, MIF:PARTITION, , 0,
part2] Updated (rev. 4).
```

```
[2004/07/08 16:10:25, gtl-host, 001704, MIF:PARTITION, , 0,
part2] Updated (rev. 5).
[2004/07/08 16:10:42, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040708000003, ] Started.
[2004/07/08 16:10:43, gtl-host, 001980, PMG:MAINTENANCE, part2,
20040708000003, ] Finished.
[2004/07/08 16:18:47, gtl-host, 001156, CLI:PARTITION, , 0,
part1] Remove operation confirmed.
[2004/07/08 16:18:47, gtl-host, 001704, MIF:PARTITION, , 0,
part1] ERROR: HSM File System mounted. Cannot remove partition.
[2004/07/08 16:19:26, gtl-host, 002120, HSM, c:\fsefs1\, 0, ]
Stopped (3.0.165 "Beta_02").
[2004/07/08 16:20:56, gtl-host, 002672, CLI:PARTITION, , 0,
part1] Remove operation confirmed.
[2004/07/08 16:20:56, gtl-host, 001712, FSE, , 0, ] Configured
volume Volume{6fb11c3a-bed1-11d8-b4f4-000e7ff13328} not found.
Will be removed from registry.
[2004/07/08 16:20:56, gtl-host, 001704, MIF:PARTITION, , 0,
part1] Partition removed.
[2004/07/08 16:20:57, gtl-host, 002068, PMG, part1, 0, ] Stopped
(3.0.165 "Beta_02").
[2004/07/08 16:21:07, gtl-host, 002672, PMG, part1, 0, ] Started
(3.0.165 "Beta_02").
[2004/07/08 16:21:08, gtl-host, 001704, MIF:PARTITION, , 0,
part1] Added new.
[2004/07/08 16:21:24, gtl-host, 001704, MIF:ADMIN, ,
20040708000004, PFY226L2] Started.
[2004/07/08 16:21:25, gtl-host, 002256, BEA, id:2:0:2:0,
20040708000004, PFY226L2] Vol 1: Changing to volume 1...
[2004/07/08 16:21:25, gtl-host, 002256, BEA, id:2:0:2:0,
20040708000004, PFY226L2] Vol 1: Rewind...
[2004/07/08 16:21:25, gtl-host, 002256, BEA, id:2:0:2:0,
20040708000004, PFY226L2] Vol 1: Changing to volume 1...
[2004/07/08 16:21:25, gtl-host, 002256, BEA, id:2:0:2:0,
20040708000004, PFY226L2] Vol 1: Rewind...
```

The FSE Error Log

The FSE error log stores information on all major errors occurring during FSE operation, such as failed initialization of an FSE medium, erroneous FSE drive operation, FSE migration job failure, and so on. These errors often require the intervention of the FSE administrator.

The FSE error log (`error.log`) is located in the `/var/opt/fse/log` directory on Linux, and in the `%InstallPath%\var\log` directory on Windows.

Example FSE error log on Linux:

```

=====
FSE Error log file
  Create time : 2004/06/17 14:11:05
=====

[W MIF() ,009799.010245 2004/06/17 14:11:05.045;
root@gtl-lnx.company.com; mif_CfgRep::ReplaceClients,
mif_cfggrep.cpp,v 5.3 :449]
Host: "gtl-lnx" resolved to FQDN: "gtl-lnx.company.com"

  Execution stack dump:
00000002. mif_CfgRep::ReplaceClients mif_cfggrep.cpp,v 5.3:449
00000001. i_ManagementInterface_i::AddPartition
i_mif_partition.cpp,v 5.13:803
  End of stack dump.
[W MIF() ,009799.011269 2004/06/17 14:11:30.983;
root@gtl-lnx.company.com; mif_CfgRep::ReplaceClients,
mif_cfggrep.cpp,v 5.3 :449]
Host: "gtl-lnx" resolved to FQDN: "gtl-lnx.company.com"

  Execution stack dump:
00000002. mif_CfgRep::ReplaceClients mif_cfggrep.cpp,v 5.3:449
00000001. i_ManagementInterface_i::AddPartition
i_mif_partition.cpp,v 5.13:803
  End of stack dump.

[E MIF() ,009799.016390 2004/06/17 14:17:56.860;
root@gtl-lnx.company.com; fse_BaseException::_Exception_,
fse_exception.cpp,v 5.2 :187]
fse_Error: (6004: Caught Corba exception.)
CORBA::COMM_FAILURE#IDL:omg.org/CORBA/COMM_FAILURE:1.0

  Execution stack dump:
00000005. fse_BaseException::_Exception_ fse_exception.cpp,v
5.2:187
00000004. i_AdminJob_i::StartBEA i_adminjob_impl.cpp,v 5.10:325
00000003. i_AdminJob_i::Execute i_adminjob_impl.cpp,v 5.10:1351
00000002. job_Executor::Run job_executor.cpp,v 5.5:62
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
  End of stack dump.

```

```
[E MIF() ,009799.016390 2004/06/17 14:17:56.861;
root@gtl-lnx.company.com; i_AdminJob_i::StartBEA,
i_adminjob_impl.cpp,v 5.10 :328]
[20040617000001] Caught error when starting BEA.fse_Error:
(6004: Caught Corba exception.)
CORBA::COMM_FAILURE#IDL:omg.org/CORBA/COMM_FAILURE:1.0
```

Execution stack dump:

```
00000004. i_AdminJob_i::StartBEA i_adminjob_impl.cpp,v 5.10:328
00000003. i_AdminJob_i::Execute i_adminjob_impl.cpp,v 5.10:1351
00000002. job_Executor::Run job_executor.cpp,v 5.5:62
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
End of stack dump.
```

```
[E MIF() ,009799.016390 2004/06/17 14:17:56.861;
root@gtl-lnx.company.com; log_EventLogger::Write,
log_eventlogger.cpp,v 5.7 :223]
[2004/06/17 14:17:56, gtl-lnx, 009799, MIF:ADMIN, ,
20040617000001, 000006] ERROR: Problems with Back End Agent
(6004: Caught Corba exception.
'CORBA::COMM_FAILURE#IDL:omg.org/CORBA/COMM_FAILURE:1.0').
```

Execution stack dump:

```
00000005. log_EventLogger::Write log_eventlogger.cpp,v 5.7:223
00000004. i_AdminJob_i::StartBEA i_adminjob_impl.cpp,v 5.10:328
00000003. i_AdminJob_i::Execute i_adminjob_impl.cpp,v 5.10:1351
00000002. job_Executor::Run job_executor.cpp,v 5.5:62
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
End of stack dump.
```

```
[E MIF() ,009799.016390 2004/06/17 14:17:56.872;
root@cumulus.hermes.si; i_AdminJob_i::Execute,
i_adminjob_impl.cpp,v 5.10 :1354]
Caught exception in Execute(): fse_Error: (6004: Caught Corba
exception.)
CORBA::COMM_FAILURE#IDL:omg.org/CORBA/COMM_FAILURE:1.0
```

Execution stack dump:

```
00000003. i_AdminJob_i::Execute i_adminjob_impl.cpp,v 5.10:1354
00000002. job_Executor::Run job_executor.cpp,v 5.5:62
```



```

00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
  End of stack dump.
[E MIF() ,009799.016390 2004/06/17 14:17:56.880;
root@cumulus.hermes.si; job_Executor::Run, job_executor.cpp,v
5.5 :71]
[20040617000001] Execute failed. fse_Error: (6004: Caught Corba
exception.)
CORBA::COMM_FAILURE#IDL:omg.org/CORBA/COMM_FAILURE:1.0

  Execution stack dump:
00000002. job_Executor::Run job_executor.cpp,v 5.5:71
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
  End of stack dump.

[W LAS(lib01) ,009817.006149 2004/06/17 14:17:57.203;
root@cumulus.hermes.si; la_SCSILibrary::EjectMedium,
la_scsi.cpp,v 5.4 :507]
Medium eject attempted on non-configured drive index: 2

```

Example FSE error log on Windows:

```

=====
FSE Error log file
  Create time : 2004/07/06 15:20:09
=====

[W RMG() ,001696.001912 2004/07/06 15:20:09.156;
SYSTEM@gtl-host.company.com; log_EventLogger::Write,
log_eventlogger.cpp,v 5.7 :219]
[2004/07/06 15:20:09, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY228L2) has no usable volumes for migration

  Execution stack dump:
00000006. log_EventLogger::Write log_eventlogger.cpp,v 5.7:219
00000005. rm_Operator::SelectMediumVolumeMig librmdb.exx,v
5.11:10301
00000004. rm_Operator::SelectMediumForMig librmdb.exx,v
5.11:10556
00000003. rm_Operator::AllocateMig librmdb.exx,v 5.11:10778
00000002. rm_DBThread::Run rm_dbthread.cpp,v 5.0:99

```

```
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
End of stack dump.
```

```
[W RMG() ,001696.001912 2004/07/06 15:20:09.171;
SYSTEM@gtl-host.company.com; log_EventLogger::Write,
log_eventlogger.cpp,v 5.7 :219]
[2004/07/06 15:20:09, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY228L2) has no usable volumes for migration
```

```
Execution stack dump:
00000006. log_EventLogger::Write log_eventlogger.cpp,v 5.7:219
00000005. rm_Operator::SelectMediumVolumeMig librmdb.exx,v
5.11:10301
00000004. rm_Operator::SelectMediumForMig librmdb.exx,v
5.11:10556
00000003. rm_Operator::AllocateMig librmdb.exx,v 5.11:10860
00000002. rm_DBThread::Run rm_dbthread.cpp,v 5.0:99
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
End of stack dump.
```

```
[W RMG() ,001696.001912 2004/07/06 15:20:09.171;
SYSTEM@gtl-host.company.com; log_EventLogger::Write,
log_eventlogger.cpp,v 5.7 :219]
[2004/07/06 15:20:09, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY229L2) has no usable volumes for migration
```

```
Execution stack dump:
00000006. log_EventLogger::Write log_eventlogger.cpp,v 5.7:219
00000005. rm_Operator::SelectMediumVolumeMig librmdb.exx,v
5.11:10301
00000004. rm_Operator::SelectMediumForMig librmdb.exx,v
5.11:10556
00000003. rm_Operator::AllocateMig librmdb.exx,v 5.11:10860
00000002. rm_DBThread::Run rm_dbthread.cpp,v 5.0:99
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
End of stack dump.
```

```
[W RMG() ,001696.001912 2004/07/06 15:27:54.640;
SYSTEM@gtl-host.company.com; log_EventLogger::Write,
log_eventlogger.cpp,v 5.7 :219]
```

```
[2004/07/06 15:27:54, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY228L2) has no usable volumes for migration
```

```
Execution stack dump:
```

```
00000006. log_EventLogger::Write log_eventlogger.cpp,v 5.7:219
00000005. rm_Operator::SelectMediumVolumeMig librddb.exx,v
5.11:10301
00000004. rm_Operator::SelectMediumForMig librddb.exx,v
5.11:10556
00000003. rm_Operator::AllocateMig librddb.exx,v 5.11:10860
00000002. rm_DBThread::Run rm_dbthread.cpp,v 5.0:99
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
End of stack dump.
```

```
[W RMG() ,001696.001912 2004/07/06 15:27:54.875;
SYSTEM@gtl-host.company.com; log_EventLogger::Write,
log_eventlogger.cpp,v 5.7 :219]
[2004/07/06 15:27:54, gtl-host, 001696, RMG, , 0, ] WARNING:
Medium (BC:PFY229L2) has no usable volumes for migration
```

```
Execution stack dump:
```

```
00000006. log_EventLogger::Write log_eventlogger.cpp,v 5.7:219
00000005. rm_Operator::SelectMediumVolumeMig librddb.exx,v
5.11:10301
00000004. rm_Operator::SelectMediumForMig librddb.exx,v
5.11:10556
00000003. rm_Operator::AllocateMig librddb.exx,v 5.11:10860
00000002. rm_DBThread::Run rm_dbthread.cpp,v 5.0:99
00000001. cmn_Thread::run cmn_thread.cpp,v 5.3:355
End of stack dump.
```

```
[E MIF() ,001704.002232 2004/07/08 16:05:10.171;
SYSTEM@gtl-host.company.com; ::yylex, cfg_l.cpp,v 5.3 :120]
invalid string")
```

```
Execution stack dump:
```

```
00000005. ::yylex cfg_l.cpp,v 5.3:120
00000004. pf_File::DoParse pf.cpp,v 5.4:450
00000003. pf_File::Parse pf.cpp,v 5.4:417
00000002. mif_CfgRep::ParseToFile mif_cfgrep.cpp,v 5.3:137
```

```
00000001. i_ManagementInterface_i::UpdatePartition
i_mif_partition.cpp,v 5.13:927
End of stack dump.

[E MIF() ,001704.002232 2004/07/08 16:05:10.171;
SYSTEM@gtl-host.company.com; ::yylex, cfg_l.cpp,v 5.3 :120]
invalid string")
    }

Execution stack dump:
00000005. ::yylex cfg_l.cpp,v 5.3:120
00000004. pf_File::DoParse pf.cpp,v 5.4:450
00000003. pf_File::Parse pf.cpp,v 5.4:417
00000002. mif_CfgRep::ParseToFile mif_cfgrep.cpp,v 5.3:137
00000001. i_ManagementInterface_i::UpdatePartition
i_mif_partition.cpp,v 5.13:927
```

Retrieving Detailed Information

The FSE command-line interface (FSE CLI) includes several commands and options that you can use to retrieve information related to configured FSE resources, files on HSM file systems, and running FSE jobs. This information indicates, for example, the parameters of configurable FSE entities, their current states, their usage information, and so on.

The scope of the retrieved information depends on the type of entity under consideration. The entities and their monitoring commands are the following:

FSE monitoring commands

FSE entity	Monitoring command
FSE libraries	fselibrary --list
FSE drives	fsedrive --list
FSE media pools	fsepool --list
FSE media	fsemedium --list
FSE medium volumes	fsemedium --list --volume
FSE partitions	fsepartition --list
Files on HSM file systems	fsefile --list
FSE jobs	fsejob --list

Additionally, FSE extends each of these monitoring commands with the `--detail` option, which has two benefits:

- It causes the monitoring commands to format their output in a different way, where each retrieved parameter of the queried entity is displayed in a separate line in the form of `ParameterName: ParameterValue`. Multiple entities of the same type are displayed one after another.

This kind of command output is suitable for parsing within scripts and helps automate FSE administrator tasks.

- With most monitoring commands, it reports additional entity details, such as low-level information, ID strings, serial numbers, usage statistics, and so on.

Note: You can only use the `--detail` option in combination with the `--list` option.

For explanation of the parameters retrieved using the `--detail` option, see the corresponding command section in Appendix B, “[Command Reference](#)” on page 251.

The next section provides examples of command outputs generated with the different FSE commands using the `--detail` option.

Examples of Command Outputs Generated using the `--list --detail` Options

Excerpt from an example output of the `fselibrary --list --detail` command

```
Name:          lib01
Family:       AIT
Type:        SCSI
Host:        gtl-host.MyCompany.com
CtrlDev:     /dev/sg3
Status:      online
ProductID:   ITL-340i
SerialNo:    0000000300
VendorID:    GRAU
FirmwareRev: 1.5
ScsiID:      id:0:0:5:0
LoadNr:      26
UnloadNr:    22
Key:         1
```

```
Name:          lib02
Family:       AIT
Type:        SCSI
Host:        gtl-host.MyCompany.com
CtrlDev:     /dev/sg4
Status:      disabled
ProductID:   ITL-340i
SerialNo:    0000000360
VendorID:    GRAU
FirmwareRev: 1.5
ScsiID:      id:0:0:4:0
LoadNr:      141
UnloadNr:    138
Key:         2

Name:          libLTO01
Family:       LTO
...
```

Excerpt from an example output of the `fsedrive --list --detail` command

```
Name:          drv01
Family:       AIT
Library:     lib01
Host:        gtl-host.MyCompany.com
CtrlDev:     /dev/sg1
Status:      online
Medium:      000010
VendorID:    SONY
ProductID:   SDX-500C
SerialNo:    0000907618
FirmwareRev: 010c
ScsiID:      id:0:0:2:0
Index:       1
UsageCount:  137
Capabilities: 0
Key:         1

Name:          drv02
Family:       AIT
Library:     lib01
```

```

Host:          c2-gtl-host.MyCompany.com
CtrlDev:      /dev/sg2
Status:       online
Medium:       000008
VendorID:     SONY
ProductID:    SDX-500C
SerialNo:     0000906652
FirmwareRev:  010c
ScsiID:       id:0:0:3:0
Index:        2
UsageCount:   20
Capabilities: 0
Key:          2

Name:         drv_lto01
Family:       LTO
...

```

Excerpt from an example output of the `fsepool --list --detail` command

```

Pool:         pool01
UUID:         ac3050b1-a838-458e-9a21-d3d3429cade9
Family:       AIT
Type:         Regular
Partition:    part01
PartUUID:     891af3c2-0aa9-4cc2-bd57-b16140691f31
BlkSize[KB]: 64
#Vol:         9
VSize[MB]:   n/a
SysVol:       Middle
SysVol[MB]:   512
Key:          1

Pool:         pool02
UUID:         33d4ef52-7ec8-4926-97f9-909edfe66c41
Family:       DISK
Type:         Regular
Partition:    part01
PartUUID:     891af3c2-0aa9-4cc2-bd57-b16140691f31
BlkSize[KB]: 64
#Vol:         1
VSize[MB]:   500
SysVol:       None

```


CurVol: 1
Key: 6

Barcode: 656701
Type: UNKNOWN
Pool:
LibName: lib01
SlotAddr: 5
SlotType: repository
DrvName:
Status: uninitialized
Location: lib:lib01:5
SerialNo:
BlockSize: 0
#Vol: 0
SysVolNo: 0
SysVolSize: 0
CurVol: 1
Key: 5

Barcode: dm01
Type: DISK
Pool: pool02
LibName:
SlotAddr:
SlotType:
DrvName:
Status: open
Location: n/a
SerialNo: 58:4
BlockSize: 65536
#Vol: 50
SysVolNo: 0
SysVolSize: 0
CurVol: 33
Key: 11

Barcode: dm02
Type: DISK
Pool: pool02
LibName:
SlotAddr:

```
SlotType:
DrvName:
Status:    open
Location:  n/a
SerialNo:  58:5
BlockSize: 65536
#Vol:      50
SysVolNo:  0
SysVolSize: 0
CurVol:   12
Key:       12
```

...

Excerpt from an example output of the `fsemedium --list --volume --detail` command

```
Barcode:    000009
VolNum:     1
UUID:       1acf6a41-a23f-4836-9e4d-539d31ad572f
Type:       Data
Size[MB]:   1166
Avail[MB]:  1166
Used[%]:    0
Status:     empty
AccessNum:  0
WriteNum:   0
OwriteNum:  0
InitTime:   2004/06/18 12:00:09
AccessTime: 1970/01/01 01:00:00
WriteTime:  1970/01/01 01:00:00
OwriteTime: 2004/03/01 17:41:05
LastVer:    1970/01/01 01:00:00
ValidData:  0
TotalData:  0

Barcode:    000009
VolNum:     2
UUID:       eca50ba5-f068-4650-a587-945fcffda287
Type:       Data
Size[MB]:   15815
Avail[MB]:  15815
```

Used[%]: 0
Status: empty
AccessNum: 0
WriteNum: 0
OwriteNum: 0
InitTime: 2004/06/18 12:01:12
AccessTime: 1970/01/01 01:00:00
WriteTime: 1970/01/01 01:00:00
OwriteTime: 2004/03/01 17:41:05
LastVer: 1970/01/01 01:00:00
ValidData: 0
TotalData: 0

Barcode: 000009
VolNum: 3
UUID: cfe954e0-21bb-428b-84fa-076754539f76
Type: System
Size[MB]: 15815
Avail[MB]: 15815
Used[%]: 0
Status: empty
AccessNum: 0
WriteNum: 0
OwriteNum: 0
InitTime: 2004/06/18 11:59:04
AccessTime: 1970/01/01 01:00:00
WriteTime: 1970/01/01 01:00:00
OwriteTime: 2004/03/01 17:41:05
LastVer: 1970/01/01 01:00:00
ValidData: 0
TotalData: 0

Barcode: 000009
VolNum: 4
UUID: b8e4d73b-890c-41d0-bb20-eefe19d3f057
Type: Data
Size[MB]: 17391
Avail[MB]: 17391
Used[%]: 0
Status: empty
AccessNum: 0
WriteNum: 0

```
OwriteNum: 0
InitTime: 2004/06/18 12:01:47
AccessTime: 1970/01/01 01:00:00
WriteTime: 1970/01/01 01:00:00
OwriteTime: 2004/03/01 17:41:05
LastVer: 1970/01/01 01:00:00
ValidData: 0
TotalData: 0

Barcode: 000007
VolNum: 1
UUID: 8920cdd9-57ee-4f2e-a09a-d701249c6434
Type: Data
Size[MB]: 1166
Avail[MB]: 1166
Used[%]: 0
Status: empty
AccessNum: 0
WriteNum: 0
OwriteNum: 0
InitTime: 2004/06/18 11:44:37
AccessTime: 1970/01/01 01:00:00
WriteTime: 1970/01/01 01:00:00
OwriteTime: 2004/03/01 17:41:05
LastVer: 1970/01/01 01:00:00
ValidData: 0
TotalData: 0

...
```

Excerpt from an example output of the `fsepartition --list --detail` command

```
Partition: part01
UUID: 891af3c2-0aa9-4cc2-bd57-b16140691f31
Status: mounted
Host: hsmfs-host01.MyCompany.com
Mountpoint: /mnt/hsmfs01
FSID: 1088756689
MigIdTime: 0
MigIdCount: 58:3
```

```
Partition: part2
UUID:      bfca4697-a1e0-4881-ba12-953acbb72a02
Status:    ready
Host:      hsmfs-host02.MyCompany.com
Mountpoint: <not mounted>
FSID:      1089124603
MigIdTime: 0
MigIdCount: 58:7

Partition: fix21
UUID:      7eb74690-69f9-433a-9c61-2394010a9ebe
Status:    disabled
Host:      hsmfs-host02.MyCompany.com
Mountpoint: <not mounted>
FSID:      1086338495
MigIdTime: 0
MigIdCount: 58:1

...
```

Excerpt from an example output of the `fsefile --list --detail` command

```
FileID:    752194
Status:    offline
FileName:  bigfile

FileID:    752195
Status:    online
FileName:  bigfile1

FileID:    42110
Status:    online
FileName:  dir18

FileID:    42111
Status:    online
FileName:  dir19

FileID:    12
Status:    offline
FileName:  linux-2.4.19-64GB-SMP.tar.bz2

FileID:    1
```

Status: online
FileName: lost+found

FileID: 2
Status: online
FileName: new_dir01

FileID: 11
Status: online
FileName: new_dir010

FileID: 3
Status: online
FileName: new_dir02

...

Excerpt from an example output of the `fsejob --list --detail` command

JobID: 20031225000001
Partition: partAIT
Type: Recall
Priority: n/a
Started: 2003/12/25 18:21:00
Status: Reading from medium

JobID: 20031225000002
Partition: partAIT
Type: Recall
Priority: n/a
Started: 2003/12/25 18:21:00
Status: Reading from medium

JobID: 20031225000003
Partition: partLTO
Type: Migration
Priority: 4001
Started: 2003/12/25 18:21:00
Status: Waiting for resources

JobID: 20031225000004
Partition: partAIT

Type: Recall
Priority: 4001
Started: 2003/12/25 18:21:00
Status: Waiting for resources

...

Backup, Restore, and Recovery



FSE enables you to have several migrated copies of the same file data, thus ensuring availability of files in case one of the copies is damaged. For details on configuring FSE to create multiple migrated copies, see “[Multiple Copying](#)” on page 144.

To make your FSE environment as safe and as stable as possible, you need to plan ahead and regularly copy FSE databases and system files (FSE system data) onto different storage media. This copying process is called FSE backup. Backup copies can be used to recreate original FSE data to a state equivalent to the one at the time of backup. This process is called FSE restore.

Depending on the data that you have lost and the applicability of your backup copies, another process can be used to recreate your data: FSE recovery. While FSE restore is the only way to recreate your data in cases of an irreparably damaged server, FSE recovery is recommended if you need to recreate a particular HSM file system or File System Catalog (FSC). The advantage of FSE recovery is that the process does not depend on backup, and ensures that your FSE installation is up-to-date with the actual state of FSE media and/or FSC. You should consider it as an alternative to the backup and restore process in cases where there are no existing up-to-date backup copies or where it is more efficient considering time and complexity of the process.

The next sections provide recommendations on recreating your data, helpful hints on how to define your backup strategy, and information on how to recover from actual data loss.

This chapter includes the following topics:

- [Defining Your Backup Strategy](#), page 192
- [Backup](#), page 195
- [Restore](#), page 207
- [Recovery](#), page 212

Defining Your Backup Strategy

Typically, if you have backup copies made regularly and according to recommendations (see “[Backup Media Management](#)” on page 197), you are able to restore any part of the FSE installation (FSE databases, FSE system files and user data). However, you should also consider using FSE recovery as a method of recreating your data, and define your backup strategy accordingly.

FSE System Components

The following table lists all FSE components that can be backed up. It states whether a component is FSE client specific, and provides notes on possible methods of recreating your data.

FSE component	Description	Recovery method
FSE file system (<i>client data</i>)	Needed to recover your files, file system structure, file attributes, and metadata. Offline files on the HSM file system are backed up as sparse files so they occupy negligible space on the backup medium.	<ul style="list-style-type: none"> ■ Recover FSE FS (see “HSM File System Recovery” on page 213) ■ Backup and restore (using full backup or single client backup)
Hierarchical Storage Management Database (HSMDB) (<i>client data</i>)	Contains migration- and release candidate lists for files on the HSM file system. The restore of this database recreates the state of restored files; without it, the state of files cannot be recreated during restore, preventing files from being migrated and released.	<ul style="list-style-type: none"> ■ If HSM FS is recovered, special recovery of HSMDB is not needed ■ Backup and restore (using full backup or single client backup)
File System Catalog (FSC)	Contains information about the location of migrated data on FSE media as well as the Name Space Catalog with metadata information on files, and is therefore needed to access files on FSE media.	<ul style="list-style-type: none"> ■ FSC recovery (see “File System Catalog Recovery” on page 213) ■ Backup and restore (using any backup type)
Resource Management Database (RMDB)	Contains records of all configured FSE resources. These are library, drive, pool, and media. In addition, it stores all relevant resource information, such as configuration parameters, resource characteristics, current status, usage statistics, and relations to other resources. In case it is not backed up, the whole resource configuration is lost.	Backup and restore (using any backup type)

FSE component	Description	Recovery method
Configuration Database (CfgDB)	Contains currently used and old revisions of the FSE configuration files for configured FSE resources; without it, all FSE configuration files are lost.	Backup and restore (using any backup type)
Fast Recovery Information (FRI)	FRI for all open media volumes is stored on disk only. It must be included in the backup since it is not yet recorded on media.	<ul style="list-style-type: none"> ■ Recreate FRI (see “Recreating Fast Recovery Information from Media” on page 106) ■ Backup and restore (using any backup type)
Custom configuration files, located in /etc/opt/fse on Linux, and in %InstallPath%\etc on Windows	Tracing, omniORB and backup configurations are customization files that hold configurations deviating from defaults.	<ul style="list-style-type: none"> ■ Backup and restore (using any backup type) ■ Configure from scratch



Caution: The only way to recreate the Resource Management Database is to restore it from the FSE system backup. This database is critical for other recovery methods, such as recovering the FSC and recreating FRI. You must always have a backup of this database; otherwise you may not be able to recover your system.

FSE Backup Types

`fsebackup` backs up all vital FSE components and data according to the backup type. FSE supports three types of backup:

- **server backup**

Backup of all required databases stored on the FSE server. Data on FSE clients (HSM file systems and HSMDBs) are excluded from a server backup.

Server backup is the preferred backup type because it is the fastest one and includes all data needed to restore your FSE installation in combination with recommended HSM file system recovery which recovers data on clients.

- **full backup**

Backup of all required databases that are stored on the FSE server as well as on FSE clients. You are able to restore the whole FSE installation (FSE server & clients) from a full backup. Note that such backup and restore can be very time consuming since full backup is usually stored on more media while backup is done on single drive. Consider using the above server backup in combination with recovery of the HSM file system instead.

- **single client backup**

Backup of all required databases that are stored on the FSE server and a specific FSE client. You are able to restore the FSE server and a specific client from a single client backup, while for recreating other clients you need to perform HSM file system recovery.

Note: An external FSE client has its file data and extended attributes stored locally, while the FSE databases (except HSMDB, which is also located on a client) and configuration are kept on the FSE server. Therefore a single client backup also includes backing up FSE databases on the FSE server.

Methods of Retrieving Data

The table below lists cases, where specified FSE system components are corrupted or lost, and suggests methods you can use to retrieve your data along with recommendations and further actions, if needed.

FSE system component	Backup type	Method of recreating data
FSE server	server backup/full backup	<ol style="list-style-type: none"> 1. FSE server restore 2. Consistency check of FSC vs. HSM file system (optional)
FSE client	no backup	Recommended: HSM file system recovery
	full backup/client backup	<ol style="list-style-type: none"> 1. FSE client restore 2. Consistency check of FSC vs. HSM file system
HSM file system	no backup	Recommended: HSM file system recovery
	full backup	<ol style="list-style-type: none"> 1. FSE client restore from full backup 2. Consistency check of FSC vs. HSM file system
File System Catalog (FSC)	no backup	Recommended: <ol style="list-style-type: none"> 1. FSC recovery 2. Consistency check of FSC vs. HSM file system or HSM file system recovery
	server backup/full backup	FSE server restore



Caution: Although there are data recreation methods available other than restoring from the backup that you can use in case of a disaster, it is strongly advised to make backups of the FSE installation on a regular basis. Restoring FSE from its backup is also the most reliable restore method.

Backup

Before you configure an automatic backup, you must know how it works, select the backup type that is appropriate for your environment, and ensure that the prerequisites are met.

What is FSE Backup?

FSE backup creates copies of data on backup media for emergency purposes. It saves a consistent state of information stored in FSE databases, FSE system files, and user data on HSM file systems.

FSE backup is performed using the `fsebackup` command. It includes all needed steps of the backup procedure; if required, it also issues commands for media loading and changing. Both backup and restore are performed on a single drive, therefore they can be very time consuming in case you perform full backups or your databases are big. However, FSE backup is an optimized process that does not obstruct normal FSE system operation if it is well considered. Therefore, you need to plan your backup strategy carefully, defining your backup type as described in “[FSE Backup Types](#)” on page 194, and be aware of backup requirements and facts, described in the following section.

How Does it Work?

You need to consider some special points when performing an FSE backup and ensure that the prerequisites are met. You should also check the recommendations for recreating data that can include an alternative to FSE backup.

Prerequisites

- On Linux installations, all HSM file systems, FSE configuration and FSE databases and system files must reside on LVM logical volumes.
- On Windows 2000 Server systems, StorageCraft Volume Snapshot must be installed. Note that the snapshot is not visible as a drive letter and is available only to the FSE backup command, which can access it to back up data.
- FSE backup media pool and FSE backup media have to be configured; for details see “[Backup Media Management](#)” on page 197.

The following information describes the FSE backup process in general. Except for the snapshot tool, which is platform specific, backing up Windows and Linux FSE clients is performed in the same way.

Presuming that the prerequisites are met, running the `fsebackup` command on the FSE server (regardless of the backup type) starts backing up the FSE installation as follows:

1. FSE is suspended to achieve a consistent state of all File System Catalogs (FSCs) and Hierarchical Storage Management Data Bases (HSMDBs).
2. Snapshots of file systems in use are created.

3. FSE system operation is resumed.
4. Data from snapshots are backed up.

While the first three steps only take a few minutes, the process of backing up can take a considerable amount of time. Check “[FSE Backup Types](#)” on page 194 for the recommended backup type.

Backup data is stored on specified backup media using `tar` format on Linux, and FSE backup data format (`fse-arch`) on a Windows systems.

Typically, FSE backup is started automatically by `cron` or similar utilities, and runs in the background. It is not possible to run multiple backup processes in parallel.

The backup progress is recorded in the `backup.log` file. Once backup is finished, it reports barcodes of the FSE backup media where backed up data is stored.

Backup start and stop times are written to the `fse.log` file and can help estimate the time needed for backup.

Backup Media Management

FSE backup requires special media management to ensure safety of the backed up data and to prevent data loss.

Note: The FSE administrator is responsible for preparing and allocating backup media as well as for keeping records on backup media, their usage and contents. These tasks are crucial for a successful backup and restore of the FSE installation.

Configuring a backup media pool

A separate media pool, called a backup media pool, must be configured before backing up the FSE installation. The backup media pool should be configured with only one volume in order to use all the space on the medium. Backup media should not be initialized. The backup process rejects initialized media in order to prevent overwriting FSE media by accident.

Note: Backup is done sequentially by one drive, therefore all backup media must belong to the inventory of the same library.

Estimating resources

You must allocate enough media for a complete backup of the FSE installation. If there is not enough space on the requested media, backup fails. You can estimate the required capacity by calculating the sizes of FSE system components involved in backup. Additionally, you can check the FSE backup log for results on used media. Note that the FSE backup process always uses a complete medium; therefore partially filled media from previous backups are not taken into account.

When the `fsebackup` command is run, it specifies a list of backup media it is going to use. The media are used in the same sequence as specified in the command line.

Media recycling

New AIT, LTO Ultrium, and SAIT media have only one preformatted partition, and can be used for FSE backup immediately, without the otherwise required formatting. However, if the media have already been used for some purpose (FSE migrations or any other application), they may contain some data that will cause FSE to refuse to overwrite them.

Preparing already used media for FSE backup purposes depends on media type.



Caution: Do not use the commands below while FSE is running. Ensure that the appropriate medium is in the drive so it will not be accidentally reloaded by a migration or recall job. You can temporarily disable a drive to ensure that the FSE jobs will not exchange the medium in the meantime. Do not use these commands on media in active FSE pools since this can corrupt the whole system and cause unrecoverable data loss.

■ Reusing AIT media for backup

To prepare used AIT media for backup, you need to format them to one partition only in order to use the whole tape for backup. Proceed as follows:

Load the medium manually into the drive that is controlled by the `DeviceFile` driver, and format it to one partition using the following command:

```
fse_scsi DeviceFile --fmt 1
```

`DeviceFile` on Linux installations is `/dev/sgX`, while on Windows installations it is represented as the string `id:P:B:T:L`. For details on how to identify SCSI device addresses, see “[About SCSI identification](#)” on page 44 for details.

The AIT medium is now ready to be used in the FSE backup pool.

■ Reusing LTO Ultrium or SAIT media for backup

To prepare used LTO Ultrium or SAIT media for backup, you only need to clear the FSE header or any other tape header; these headers prevent FSE from overwriting the tape. LTO Ultrium and SAIT media are already formatted to one partition. Proceed as follows:

Load the medium manually into the drive and clear the medium header using the `fse_scsi` tool. In place of `<DeviceFile>`, specify the SCSI device ID of the drive:

```
fse_scsi <DeviceFile> --rew
fse_scsi <DeviceFile> --erase
```

SCSI device IDs have the form `/dev/sgX` (on Linux systems) or `id:P:B:T:L` (on Windows systems). For details on how to identify SCSI device addresses (IDs), see “[About SCSI identification](#)” on page 44 for details.

The LTO Ultrium or SAIT medium is now ready to be used in the FSE backup pool.

Estimating backup frequency

You need to estimate how often you should run an FSE backup. Analyze your needs and your environment with regard to the backup duration time and the FSE usage policy.

Note that you cannot run more than one `fsebackup` at a time and that backup frequency depends on the FSE usage policy. If there is a lot of traffic on your file system and big changes are frequent, this should be reflected in your plans for backup. The more frequent the changes on the file system, the more frequent your backups should be. Once you determine these factors, schedule your backups accordingly:

- **Linux:** Using the `crontab` command, configure a regular cron job that will start `fsebackup` with the desired frequency and backup set.
- **Windows:** Schedule a task to run the `fsebackup` command at a specified time using either Scheduled Task Wizard (**Start > Settings > Control Panel > Scheduled Task > Add Scheduled Tasks**) or the `at` command.

Note: It is recommended that you perform your backups on a daily basis. You should have at least two separate sets of backup media and alternate their use, thus ensuring that there is always at least one successful backup media set available for restore.

Modifying backup configuration on Linux

On Linux, FSE provides a `backup.cfg` file in the `/etc/opt/fse` directory. It can be used for backup configurations if the defaults do not suit your needs. You can specify different logical groups/volumes for `/var/opt/fse` and `/var/opt/fse/part` mount points according to your specific configuration. You can also modify the default LVM snapshot size (as a percentage of the original logical volume size, default is 10). Increase the default value if there are many changes expected on the HSM file system while the backup is in progress.

If the `/etc/opt/fse/backup.cfg` file is not found, default values are used for the FSE backup process.

The `fsebackup` Command

FSE backup is started with the `fsebackup` command, followed by the mandatory `--media Barcode` option. When performing a full backup, you need to use the general `--client` option, while backing up a specific client requires using the `--client ClientName` command option.

- **Linux:** `/opt/fse/sbin/fsebackup --media Barcode [--client ClientName]`
- **Windows:** `%InstallPath%\bin\fsebackup --media Barcode [--client ClientName]`

If you do not want to back up the HSM file system, you can use option `--skip-hsmfs` together with the `--client` option. This way only metadata of the FSE partition (HSMDB) is backed up, while the HSM file system is skipped. Exceptions to this are symbolic links, file extended attributes (EAs) and access control lists (ACLs), which are ignored and need to be backed up separately if the file system is not backed up.

Example of backing up an FSE server

The FSE server is going to be backed up to the FSE medium with barcode 100190:

```
& fsebackup --media 100190
```

Example of a full backup

The whole FSE installation (the FSE server and all FSE clients) is going to be backed up to the FSE media with barcodes 100190, 100191, 100192:

```
& fsebackup --media 100190 100191 100192 --client
```

Example of an FSE client backup

The FSE server and a specified FSE client are going to be backed up to the FSE medium with barcode 001243:

```
& fsebackup --media 001243 --client fseclient.company.com
```

Note that the FSE client must be specified by its fully-qualified domain name (FQDN).

Example FSE backup of a Linux installation:

```
2003/12/16 10:54:09 alfa.company.com Backup started.
2003/12/16 10:54:09 alfa.company.com Configured partitions and
their hosts:
2003/12/16 10:54:09 alfa.company.com   stratol on host
alfa.company.com
2003/12/16 10:54:09 alfa.company.com   strato2 on host
alfa.company.com
2003/12/16 10:54:09 alfa.company.com   strato3 on host
alfa.company.com
2003/12/16 10:54:10 alfa.company.com Suspending all
partitions...
2003/12/16 10:54:11 alfa.company.com ...partitions suspended:
Suspend performed for
  3 FSC of 3
  3 HSM of 3

2003/12/16 10:54:11 alfa.company.com Make snapshots for
alfa.company.com...
2003/12/16 10:54:13 alfa.company.com Creating snapshot of
vgraid1/fsevar...
2003/12/16 10:54:13 alfa.company.com lv_snap_size = 1500 MB
2003/12/16 10:54:14 alfa.company.com Creating snapshot of
vgraid5/lvpart...
2003/12/16 10:54:14 alfa.company.com lv_snap_size = 2000 MB
2003/12/16 10:54:17 alfa.company.com Mounting
vgraid1/fsevar_snap...
2003/12/16 10:54:17 alfa.company.com Mounted.
2003/12/16 10:54:17 alfa.company.com Mounting
vgraid5/lvpart_snap...
2003/12/16 10:54:17 alfa.company.com Mounted.
2003/12/16 10:54:17 alfa.company.com Must make snapshots for
HSMFS:
2003/12/16 10:54:17 alfa.company.com stratol vgraid5/lvstrato1
ext3
2003/12/16 10:54:17 alfa.company.com strato2 vgraid5/lvstrato2
ext3
```

```
2003/12/16 10:54:17 alfa.company.com strato3 vgraid5/lvstrato3
ext3
2003/12/16 10:54:17 alfa.company.com Creating snapshot of
vgraid5/lvstrato1...
2003/12/16 10:54:17 alfa.company.com lv_snap_size = 1500 MB
2003/12/16 10:54:19 alfa.company.com Creating snapshot of
vgraid5/lvstrato2...
2003/12/16 10:54:19 alfa.company.com lv_snap_size = 100 MB
2003/12/16 10:54:20 alfa.company.com Creating snapshot of
vgraid5/lvstrato3...
2003/12/16 10:54:20 alfa.company.com lv_snap_size = 200 MB
2003/12/16 10:54:21 alfa.company.com Mounting
vgraid5/lvstrato1_snap...
2003/12/16 10:54:21 alfa.company.com Mounted.
2003/12/16 10:54:21 alfa.company.com Dump EA to file...
2003/12/16 10:54:23 alfa.company.com EA written to file.
2003/12/16 10:54:23 alfa.company.com Mounting
vgraid5/lvstrato2_snap...
2003/12/16 10:54:23 alfa.company.com Mounted.
2003/12/16 10:54:23 alfa.company.com Dump EA to file...
2003/12/16 10:54:26 alfa.company.com EA written to file.
2003/12/16 10:54:26 alfa.company.com Mounting
vgraid5/lvstrato3_snap...
2003/12/16 10:54:26 alfa.company.com Mounted.
2003/12/16 10:54:26 alfa.company.com Dump EA to file...
2003/12/16 10:54:29 alfa.company.com EA written to file.
2003/12/16 10:54:29 alfa.company.com Pickling host info (for
later)...
2003/12/16 10:54:29 alfa.company.com Pickling partition info (to
send to server)...
2003/12/16 10:54:29 alfa.company.com Sending pickle file...
2003/12/16 10:54:29 alfa.company.com Snapshots done for
alfa.company.com.
2003/12/16 10:54:29 alfa.company.com Adding pickled partition
info to context...
2003/12/16 10:54:29 alfa.company.com Resuming partitions...
2003/12/16 10:54:29 alfa.company.com Partitions are back online:
Continue performed for
  3 FSC of 3
  3 HSM of 3
2003/12/16 10:54:29 alfa.company.com Preparing RMDB backup
dump...
2003/12/16 10:54:30 alfa.company.com ...RMDB dump done.
2003/12/16 10:54:30 alfa.company.com Reserving medium...
2003/12/16 10:54:36 alfa.company.com ...reserved medium and will
use drive /dev/sg5.
2003/12/16 10:54:36 alfa.company.com Creating archive...
```

```
2003/12/16 10:54:46 alfa.company.com Rewind tape & check medium
header...
2003/12/16 10:55:06 alfa.company.com Writing client data of
partition stratol on host alfa.company.com...
2003/12/16 10:55:27 alfa.company.com Changing medium...
2003/12/16 10:55:27 alfa.company.com Changing tape 050004 with
050005...
2003/12/16 10:56:02 alfa.company.com Waiting for tape to settle
down...
2003/12/16 10:56:12 alfa.company.com Medium changed.
2003/12/16 10:55:27 alfa.company.com Changing medium...
2003/12/16 10:55:27 alfa.company.com Changing tape 050004 with
050005...
2003/12/16 10:56:02 alfa.company.com Waiting for tape to settle
down...
2003/12/16 10:56:12 alfa.company.com Medium changed.
2003/12/16 10:58:02 alfa.company.com Client alfa.company.com
done for component hsmfs.
2003/12/16 10:58:02 alfa.company.com Medium has been changed 1
times.
2003/12/16 10:58:02 alfa.company.com Writing client data of
partition stratol on host alfa.company.com...
2003/12/16 10:58:06 alfa.company.com Client alfa.company.com
done for component hsmdb.
2003/12/16 10:58:06 alfa.company.com Writing FSC of partition
stratol...
2003/12/16 10:58:07 alfa.company.com Writing client data of
partition strato2 on host alfa.company.com...
2003/12/16 10:58:50 alfa.company.com Client alfa.company.com
done for component hsmfs.
2003/12/16 10:58:50 alfa.company.com Writing client data of
partition strato2 on host alfa.company.com...
2003/12/16 10:58:53 alfa.company.com Client alfa.company.com
done for component hsmdb.
2003/12/16 10:58:54 alfa.company.com Writing FSC of partition
strato2...
2003/12/16 10:58:55 alfa.company.com Writing client data of
partition strato3 on host alfa.company.com...
2003/12/16 10:59:58 alfa.company.com Client alfa.company.com
done for component hsmfs.
2003/12/16 10:59:58 alfa.company.com Writing client data of
partition strato3 on host alfa.company.com...
2003/12/16 11:00:01 alfa.company.com Client alfa.company.com
done for component hsmdb.
2003/12/16 11:00:01 alfa.company.com Writing FSC of partition
strato3...
2003/12/16 11:00:02 alfa.company.com Writing RMDB...
2003/12/16 11:00:04 alfa.company.com Writing CfgDB, FRI...
```

```
2003/12/16 11:00:07 alfa.company.com Writing etc...
2003/12/16 11:00:12 alfa.company.com Writing backup spec to
tape...
2003/12/16 11:00:13 alfa.company.com Archive created.
2003/12/16 11:00:13 alfa.company.com Used media: 050004 050005
2003/12/16 11:00:13 alfa.company.com Host alfa.company.com:
medium 0, filemark 0 (incl: 4).
2003/12/16 11:00:13 alfa.company.com Cleanup of temporary backup
files.
2003/12/16 11:00:13 alfa.company.com Release snapshots for
alfa.company.com...
2003/12/16 11:00:16 alfa.company.com Unmount and remove snapshot
for vgraid1/fsevar...
2003/12/16 11:00:16 alfa.company.com Unmount and remove snapshot
for vgraid5/lvpart...
2003/12/16 11:00:17 alfa.company.com Unmount and remove snapshot
for FS_strato1...
2003/12/16 11:00:18 alfa.company.com Unmount and remove snapshot
for FS_strato2...
2003/12/16 11:00:18 alfa.company.com Unmount and remove snapshot
for FS_strato3...
2003/12/16 11:00:19 alfa.company.com Snapshots released for
alfa.company.com.
2003/12/16 11:00:19 alfa.company.com Backup successful.
```

Example FSE backup of a Windows installation:

```
Backup (#files: 0, #failed files: 1, size: 0 KB).
2004/04/14 10:20:29 testa.company.com Writing FSC of partition
part2...
2004/04/14 10:20:29 testa.company.com Executing: fse-arch
--backup "" 32
"\\?\Volume{cfe0dd5f-3155-11d8-a16f-806d6172696f}\ part2|
fse-dd --write id:2:0:2:0 "" 32 --eom "CHANGE_MEDIUM_CMD"
2004/04/14 10:20:32 testa.company.com Warning: Skipping
\\?\Volume{cfe0dd5f-3155-11d8-a16f-806d6172696f}\part2

Backup (#files: 0, #failed files: 1, size: 0 KB).
2004/04/14 10:20:32 testa.company.com Writing RMDB...
2004/04/14 10:20:32 testa.company.com Executing: fse-arch
--backup "" 32 "C:\WINNT\Temp\FSE\fse" RMDB.bckp| fse-dd
--write id:2:0:2:0 "" 32 --eom "CHANGE_MEDIUM_CMD"
2004/04/14 10:20:34 testa.company.com
2004/04/14 10:20:29 testa.company.com Warning: Skipping
\\?\Volume{cfe0dd5f-3155-11d8-a16f-806d6172696f}\part1
Backup (#files: 1, #failed files: 0, size: 30 KB).
2004/04/14 10:20:34 testa.company.com Writing CfgDB, FRI...
```

```
2004/04/14 10:20:34 testa.company.com Executing: fse-arch
--backup "" 32 "C:\Program Files\Hewlett-Packard\FSE\var" \ cfg
fri | fse-dd --write id:2:0:2:0 "" 32 --eom "CHANGE_MEDIUM_CMD"
2004/04/14 10:20:38 testa.company.com
Backup (#files: 39, #failed files: 0, size: 11628 KB).
2004/04/14 10:20:38 testa.company.com Writing etc...
2004/04/14 10:20:38 testa.company.com Executing: fse-arch
--backup "" 32 "C:" \ "Program Files\Hewlett-Packard\FSE\etc" |
fse-dd --write id:2:0:2:0 "" 32 --eom "CHANGE_MEDIUM_CMD"
2004/04/14 10:20:41 testa.company.com
Backup (#files: 10, #failed files: 0, size: 8 KB).
2004/04/14 10:20:41 testa.company.com Writing backup spec to
tape...
2004/04/14 10:20:41 testa.company.com Executing: fse-arch
--backup "" 32 "C:\WINNT\Temp\FSE\.fse" \ SPEC.bckp SPEC.pickle|
fse-dd --write id:2:0:2:0 "" 32 --eom "CHANGE_MEDIUM_CMD"
2004/04/14 10:20:43 testa.company.com
Backup (#files: 2, #failed files: 0, size: 1 KB).
2004/04/14 10:20:43 testa.company.com Archive created.
2004/04/14 10:20:43 testa.company.com Used media: 052010
2004/04/14 10:20:43 testa.company.com Host testa.company.com:
medium 0, filemark 0 (incl: 2).
2004/04/14 10:20:44 testa.company.com Cleanup of temporary
backup files.
2004/04/14 10:20:44 testa.company.com Release snapshots for
testa.company.com...
2004/04/14 10:20:48 testa.company.com Removing snapshot
\\.\Snapshot0, volume ID
\\?\Volume{cfe0dd5f-3155-11d8-a16f-806d6172696f}
2004/04/14 10:20:49 testa.company.com Snapshots released for
testa.company.com.
2004/04/14 10:20:49 testa.company.com Backup successful.
2004/04/14 10:19:40 testa.company.com Backup started.
2004/04/14 10:19:42 testa.company.com Configured partitions and
their hosts:
2004/04/14 10:19:42 testa.company.com part1 on host
testa.company.com
2004/04/14 10:19:42 testa.company.com part2 on host
testa.company.com
2004/04/14 10:19:42 testa.company.com Suspending all
partitions...
2004/04/14 10:19:42 testa.company.com ...partitions suspended:
Suspend performed for
2 FSC of 2
0 HSM of 2

2004/04/14 10:19:42 testa.company.com Make snapshots for
testa.company.com...
```

```
2004/04/14 10:19:45 testa.company.com Client-side script
initialized, making snapshots.
2004/04/14 10:19:45 testa.company.com Creating snapshot of
\\?\Volume{cfe0dd5f-3155-11d8-a16f-806d6172696f}...
2004/04/14 10:19:47 testa.company.com Mounting
\\?\Volume{cfe0dd5f-3155-11d8-a16f-806d6172696f}...
2004/04/14 10:19:47 testa.company.com Snapshot mounted as
\\.\SnapShot0\
2004/04/14 10:19:47 testa.company.com Skip HSMFS snapshots.
2004/04/14 10:19:47 testa.company.com Pickling host info (for later)...
2004/04/14 10:19:47 testa.company.com Pickling partition info
(to send to server)...
2004/04/14 10:19:47 testa.company.com Sending pickle file...
2004/04/14 10:19:47 testa.company.com Snapshots done for
testa.company.com.
2004/04/14 10:19:48 testa.company.com No pickled partition info
to add.
2004/04/14 10:19:48 testa.company.com Resuming partitions...
2004/04/14 10:19:48 testa.company.com Partitions are back
online:
  Continue performed for
  2 FSC of 2
  0 HSM of 0

2004/04/14 10:19:48 testa.company.com Preparing RMDB backup
dump...
2004/04/14 10:19:48 testa.company.com Invoking gbak.cmd
2004/04/14 10:19:49 testa.company.com

2004/04/14 10:19:49 testa.company.com C:\Program
Files\Hewlett-Packard\FSE\bin>"C:\Program
Files\Firebird\bin\gbak" -B -USER sysdba -PASS masterkey
localhost:"C:\Program
Files\Hewlett-Packard\FSE\var\rmdb\rmdb.gdb"
"C:\WINNT\Temp\FSE\.fse\RMDB.bckp" 2>&1
2004/04/14 10:19:49 testa.company.com ...RMDB dump done.
2004/04/14 10:19:49 testa.company.com Reserving medium...
2004/04/14 10:19:52 testa.company.com ...reserved medium and
will use drive id:2:0:2:0.
2004/04/14 10:19:52 testa.company.com Creating archive...
2004/04/14 10:20:02 testa.company.com Rewind tape & check medium
header...
2004/04/14 10:20:24 testa.company.com Writing FSC of partition
part1...
2004/04/14 10:20:24 testa.company.com Executing: fse-arch
--backup "" 32
"\\?\Volume{cfe0dd5f-3155-11d8-a16f-806d6172696f}"\ part1|
fse-dd --write id:2:0:2:0 "" 32 --eom "CHANGE_MEDIUM_CMD"
```



```
2004/04/14 10:20:29 testa.company.com Warning: Skipping
\\?\Volume{cfe0dd5f-3155-11d8-a16f-806d6172696f}\part1
```

Restore

Contrary to FSE backup, restoring FSE data is an interactive process. Before performing a restore, you must know how to restore your data and be fully aware of the FSE concepts and the meaning of restore. You need to be familiar with the FSE directory layout. For more information, see “[Directory Layout](#)” on page 249.

Note: FSE also provides recovery of the File System Catalog and the HSM file system as an alternative way of recreating data to backup and restore. See the recommended ways of using backup and restore and FSE recovery in “[Backup](#)” on page 195.

What is FSE Restore?

FSE restore recreates the specified parts or the whole FSE installation to a state equal to the one at the time of backup. Providing that the FSE backup was performed by the `fsebackup` command according to the described procedure, restore is run by the `fserestore` command. An FSE administrator must ensure that the prerequisites are met and follow the restore procedure that guides him/her through the whole process of restoring.

Depending on what you have backed up, the following FSE components can be restored:

- **FSE server**

Restore of FSE configuration and databases, located in the `/var/opt/fse` directory on Linux, and in the `%InstallPath%\var\` directory on Windows.

To restore the Resource Management Database (RMDB), Firebird SuperServer must be running on the consolidated FSE implementation or on the FSE server.

- **FSE server and all FSE clients**

Includes the FSE server and FSE client data.

- **FSE client**

Restore of the HSM file system data and file extended attributes for every configured partition. File system data is restored using the same format that was used for backup, that is `tar` format for Linux systems and FSE backup

data format (`fse-arch`) for Windows systems. On Linux systems, the `setfattr` tool is used for restoring the extended attributes.

Restore Procedure

You perform a restore of FSE by running the `fserestore` command. The restore process requires interactive input from the FSE administrator.

Note: You need to run the `fserestore` command on the FSE server system, regardless of the type of data you want to restore (FSE server or FSE client).

Restore requirements

- Restoring data to its original location
In order to successfully restore the FSE installation, all data that was backed up must be restored to its original location. You may want to back up current data before performing a restore (in case the restore fails).
- Restoring to different devices
Since restore is often needed when the hard disk has failed, it is very common to make a restore to devices different than the ones used for backup. The only restriction is that the replacement disk must have the same characteristics as the original disk had at the backup time (LVM volume names, volume sizes, mount points).
- Backup media must be ready for restore
All backup media that is needed for restore must be available. Since the FSE installation is stopped, the loading and unloading of backup media must be done manually.

Linux prerequisites

Before restoring a Linux system:

- You have to be logged on to the system as root in order to perform the restore steps.
- Logical volumes on all the clients that you are going to restore must have the same names, file system types, and sizes as they did at the time of backup. These logical volumes are `fsevar`, `fsepart`, and `fse-diskbuf` (the latter only on the standalone FSE system or FSE server).

- Logical volumes `fsevar`, `fsepart`, and `fseiskbuf` should be mounted on the specified location in the FSE directory structure. See “[Directory Layout](#)” on page 249 for more information.
- HSM file systems must not be mounted.
- HSM file systems must have the same logical volume names as before.

Windows prerequisites

Before restoring a Windows system:

- You must have administrative privileges in order to perform the restore steps. These privileges are granted only if you are locally logged to the system as Administrator. Remote Desktop cannot be used for this task.
- Disk volumes that will store the restored HSM file systems must have at least the same sizes as they had at the time of backup creation.
- Mount points for the restored HSM file systems must be the same as they were when the backup was made.

Executing FSE restore

To restore the FSE system, perform the following steps on each FSE system host that you want to restore. Restore external FSE clients before restoring the FSE server:

1. Stop FSE processes:

```
fse --stop
```

2. Start the FSE processes that have to be running to allow the restore process:

```
fse --startrestore
```

3. If you are operating on a Windows FSE client, detach the HSM file system filter from all disk volumes that store HSM file systems. For each such volume, run the following command:

```
C:\>fse --dismount-ntfs VolumeName
```

4. On the FSE server, initiate the restore process with the following command:

```
fserestore
```

The FSE system starts restoring data from the backup media. In case that multiple media were used for backup, you must provide the media when requested. You must insert the media in the same sequence as they were used for backup. Note that the same drive must be used for all media.

Example FSE restore

```
2003/12/16 11:22:57 alfa.company.com Started on host
alfa.company.com
2003/12/16 11:23:01 alfa.company.com Restore will use tape
device /dev/sg5
2003/12/16 11:25:39 alfa.company.com Rewind tape & check medium
header...
2003/12/16 11:25:39 alfa.company.com Restoring client
alfa.company.com...
2003/12/16 11:25:40 alfa.company.com Attempting restore of
client alfa.company.com
2003/12/16 11:25:40 alfa.company.com Got these cmdline
arguments: /opt/fse/lbin/fse-restore.py --restore-fseclient
stratol vgraid5/lvstratol ext3 alfa.company.com /dev/sg5 4
2003/12/16 11:25:40 alfa.company.com Restoring partition
stratol on client alfa.company.com...
2003/12/16 11:25:40 alfa.company.com Mounting
vgraid5/lvstratol...
2003/12/16 11:25:40 alfa.company.com Mounted.
2003/12/16 11:25:40 alfa.company.com Reading HSMFS of partition
stratol...
2003/12/16 11:25:56 alfa.company.com Changing medium...
2003/12/16 11:25:56 alfa.company.com Changing tape 050004 with
050005...
2003/12/16 11:25:56 alfa.company.com Waiting for user
confirmation...
2003/12/16 11:27:11 alfa.company.com Medium changed.
2003/12/16 11:25:40 alfa.company.com Attempting restore of
client alfa.company.com
2003/12/16 11:25:40 alfa.company.com Got these cmdline
arguments: /opt/fse/lbin/fse-restore.py --restore-fseclient
stratol vgraid5/lvstratol ext3 alfa.company.com /dev/sg5 4
2003/12/16 11:25:40 alfa.company.com Restoring partition
stratol on client alfa.company.com...
2003/12/16 11:25:40 alfa.company.com Mounting
vgraid5/lvstratol...
2003/12/16 11:25:40 alfa.company.com Mounted.
2003/12/16 11:25:40 alfa.company.com Reading HSMFS of partition
stratol...
2003/12/16 11:25:56 alfa.company.com Changing medium...
2003/12/16 11:25:56 alfa.company.com Changing tape 050004 with
050005...
2003/12/16 11:25:56 alfa.company.com Waiting for user
confirmation...
2003/12/16 11:27:11 alfa.company.com Medium changed.
2003/12/16 11:28:56 alfa.company.com Setting extended
attributes...
```

```
2003/12/16 11:28:56 alfa.company.com Unmounting partition
filesystem...
2003/12/16 11:28:58 alfa.company.com Reading HSM DB of partition
strato1...
2003/12/16 11:28:58 alfa.company.com Restore complete for
partition strato1 on client alfa.company.com.
2003/12/16 11:28:58 alfa.company.com Reading FSC of partition
strato1...
2003/12/16 11:28:58 alfa.company.com Attempting restore of
client alfa.company.com
2003/12/16 11:28:58 alfa.company.com Got these cmdline
arguments: /opt/fse/lbin/fse-restore.py --restore-fseclient
strato2 vgraid5/lvstrato2 ext3 alfa.company.com /dev/sg5 4
2003/12/16 11:28:58 alfa.company.com Restoring partition
strato2 on client
2003/12/16 11:28:58 alfa.company.com Mounting
vgraid5/lvstrato2...
2003/12/16 11:28:58 alfa.company.com Mounted.
2003/12/16 11:28:58 alfa.company.com Reading HSMFS of partition
strato2...
2003/12/16 11:29:57 alfa.company.com Setting extended
attributes...
2003/12/16 11:29:57 alfa.company.com Unmounting partition
filesystem...
2003/12/16 11:29:59 alfa.company.com Reading HSM DB of partition
strato2...
2003/12/16 11:29:59 alfa.company.com Restore complete for
partition strato2 on client alfa.company.com.
2003/12/16 11:29:59 alfa.company.com Reading FSC of partition
strato2...
2003/12/16 11:29:59 alfa.company.com Attempting restore of
client alfa.company.com
2003/12/16 11:29:59 alfa.company.com Got these cmdline
arguments: /opt/fse/lbin/fse-restore.py --restore-fseclient
strato3 vgraid5/lvstrato3 ext3 alfa.company.com /dev/sg5 4
2003/12/16 11:29:59 alfa.company.com Restoring partition
strato3 on client alfa.company.com...
2003/12/16 11:29:59 alfa.company.com Mounting
vgraid5/lvstrato3...
2003/12/16 11:29:59 alfa.company.com Mounted.
2003/12/16 11:29:59 alfa.company.com Reading HSMFS of partition
strato3...
2003/12/16 11:31:13 alfa.company.com Setting extended
attributes...
2003/12/16 11:31:15 alfa.company.com Unmounting partition
filesystem...
2003/12/16 11:31:16 alfa.company.com Reading HSM DB of partition
strato3...
```

```
2003/12/16 11:31:16 alfa.company.com Restore complete for
partition strato3 on client alfa.company.com.
2003/12/16 11:31:16 alfa.company.com Reading FSC of partition
strato3...
2003/12/16 11:31:16 alfa.company.com Reading RMDB...
2003/12/16 11:31:16 alfa.company.com Reading CfgDB, FRI...
2003/12/16 11:31:17 alfa.company.com Reading etc...
2003/12/16 11:31:17 alfa.company.com Recreating RMDB from
rmdb.bckp...
2003/12/16 11:31:23 alfa.company.com Restore complete.
```

Post-restore Steps

After the restore is finished, perform the following steps on each FSE system host that has been restored starting with external FSE clients and finishing with the FSE server:

1. Stop the FSE processes that have been running to allow the restore process:
`fse --stop`
2. Start all FSE processes:
`fse --start`
3. If you are operating on a Windows FSE client, detach the HSM FS filter from all disk volumes that store HSM file systems. For each such volume, run the following command:

```
C:\>fse --dismount-ntfs VolumeName
```

Run the consistency check of the File System Catalog (FSC) against the corresponding HSM file system to verify the consistency of the data. For details, see [“Consistency Check of FSC vs. HSM File System”](#) on page 157.

Recovery

FSE provides functionality for recovering from disasters that can happen to the FSE installation, and which can be used as an alternative or as a supplement to the conventional FSE restore process. HSM file systems and File System Catalogs (FSCs) can be restored using specially tailored recovery methods.

If one or more HSM file systems are damaged, you can use HSM file system recovery to recreate an HSM file system from the Name Space Catalogs, as an alternative to FSE backup and restore.

File System Catalog (FSC) recovery is designed for recreation of the damaged or missing FSC files from the Fast Recovery Information (FRI) data and the data migrated to FSE media. It can also be used in less severe situations, for example in case entries in the FSC are inconsistent with the corresponding HSM file system.

See the recommended ways of using backup and restore and FSE recovery in “[Backup](#)” on page 195.

File System Catalog Recovery

The File System Catalog is recreated either from Fast Recovery Information (FRI), if it is available, or from the migrated metadata. To perform FSC recovery, use:

```
fserecover --fsc PartitionName
```

For example, to start recovery of the File System Catalog for FSE partition `fse_fs01` enter:

```
fserecover --fsc fse_fs01
```

When running the `fserecover --fsc` command the following occurs:

1. If FRI is located on disk, which is the case for all data volumes that are not yet full, recovery is done from these temporary FRI files.
2. If a data volume is full and its FRI has been written to the system volume on the medium, the recovery is done from the system volume FRI. This is the fastest method since all FRI is located on the system volume.
3. If a medium volume is not partitioned to include the system volume, each data volume has its own FRI, and recovery is done accordingly.
4. If FRI is not available, the recovery process needs to recreate it from file data on the data volume. This is a time-consuming process since all data need to be read first, and only then can FRI be recreated and used to recover FSC.

Once FSC recovery is completed, you should run the FSC consistency check as described in “[Consistency Check of FSC vs. HSM File System](#)” on page 157.

HSM File System Recovery

An HSM file system is recovered directly from the Name Space Catalog (NSC), which is a part of the File System Catalog. Since the Name Space Catalog contains complete file system structure data (such as the directory structure, file attributes, and security), the HSM file system structure can be recreated according to this information.

HSM file system recovery is necessary in cases where the HSM file system is damaged and has not been backed up, therefore it cannot be restored. However, it is also a recommended procedure for recreating the file system when a backup copy does exist, as described in “[Backup](#)” on page 195.

How it works

Prerequisite

- The Name Space Catalog (NSC) of the FSE partition whose HSM file system will be recovered must exist.

Linux specific

To recover an HSM file system on a Linux FSE client:

1. From the local `/etc/fstab` file, determine the device (block special device node) of the HSM file system, and run the following command:

```
# umount Device
```
2. Format the HSM file system using the `mke2fs` command. Refer to “[Creating File Systems On Top of LVM Volumes](#)” on page 39 of the *FSE Installation Guide* for more information.
3. Start the recovery of the HSM file system:

```
# fserecover --hsmfs PartitionName
```

Windows specific

To recover an HSM file system on a Windows FSE client:

1. Using the `fsepartition --show` and `mountvol` commands, determine the corresponding volume name and run the following command:

```
C:\>fse --umount MountPoint VolumeName
```
2. Format the HSM file system using **Computer Management** or the `format` command.
3. Unmount the HSM file system once again as described in step 1.
4. Start recovery of the HSM file system. Use the following command for this purpose:

```
C:\>fserecover --hsmfs PartitionName
```


Assuming the prerequisites are met, this will start recreating the file system structure as follows:

- a. The recovery job checks if the recovery has already been started so it can be resumed from the last recovered fileID.
- b. It verifies that the HSM file system is not mounted and then mounts it on the temporary mount point.
- c. It removes the existing HSMDB.
- d. It scans the FSC and recovers the file system.

The FSC scanner retains the last recovered fileID from

`/var/opt/fse/tmp/.<partition_name>-_hsmfs_recover` on Linux, or
`%InstallPath%\var\tmp\<partition_name>-_hsmfs_recover` file on Windows. This enables file system recovery to be resumed after a power failure or other interruptions. In case this file is lost, recovery is started from the beginning. If you want to start an interrupted recovery from the start, you need to delete this file.

- e. It unmounts the temporarily mounted HSM FS and removes temporary mount points and device files.

Once the file system is recovered, the file state is as follows:

- File index (Windows) and I-node number (Linux) are changed after the recovery
- File access and creation time are proper, but other file time attributes may be modified
- File is offline
- Directory may not have proper change, modify, and access time set.

Note: Once the recovery process is finished, offline files need to be recalled to recreate the complete HSM file system. When the files are back online, their state is updated.

Limitations

- Recovery cannot properly handle user changes on an HSM file system that occurred after the last migration is finished. To prevent this situation, you need to format the HSM file system.
- Symbolic links cannot be recovered because they are not migrated to FSE media.
- Recovery of Windows Alternate Data Streams (ADS) data is not supported, except for the names of ADS, which are stored in NSC.

Troubleshooting



In this chapter, the most common errors regarding installation, configuration, and use of FSE are presented, including instructions on how to collect debug traces for different kinds of FSE activities. It also contains a list of all errors that can be issued by the FSE installation, with a one-line explanation for each.

This chapter includes the following topics:

- [HSM File System Filter Problems](#), page 218
- [HSM File System Mounting Problems](#), page 219
- [FSE Media Management Problems](#), page 222
- [FSE Migration Problems](#), page 223
- [FSE Recall Problems](#), page 223
- [FSE Backup and Restore Problems](#), page 224
- [FSE Start-up and Shut-down Problems](#), page 225
- [Communication Problems](#), page 226
- [Firebird Server Problems](#), page 228
- [SCSI Problems](#), page 229
- [Tape Library Problems](#), page 230
- [FSE Troubleshooting Tools](#), page 231
- [Debugging FSE Processes](#), page 232
- [Contacting Support](#), page 238
- [FSE Error Codes](#), page 240

If you have a problem with FSE, you should check the FSE documentation before contacting your support representative. This chapter gives you some clues on what is happening with FSE and how to solve your problem. Usually you first need to check the FSE event log `fse.log` for indications of problems, located in

`/var/opt/fse/log/` on Linux and in `%InstallPath%\var\log` on Windows. (The default `%InstallPath%` is `Program Files\Hewlett-Packard\FSE`, but you can change this during installation by selecting Custom setup.)

If you have already analyzed your problem and could not find a solution in FSE and other relevant documentation, you must collect relevant information about your FSE system configuration, your specific situation, and the symptoms that you are experiencing. Such a prepared data set will speed up the problem investigation and its resolution when contacting your support representative. For more information, see “[Contacting Support](#)” on page 238.

HSM File System Filter Problems

Wrong Kernel Version on the FSE Client

Linux specific

Sometimes during FSE client installation the following error message is displayed, and the installation fails:

```
Setting HSM File System Filter module:
ERROR: FS Filter (hsmfs.o) not available for kernel 2.4.19-340
...
Loading HSM FS Filter module: [FAILED]
```

This message is issued because the HSM file system filter module is only available for the Linux kernel version 2.4.21 and FSE will not run with any other kernel. See the *FSE Release Notes* for the list of supported kernels, and, if needed, install a supported version.

Checking HSM File System Kernel Messages

To see the messages or debug trace information that the HSM file system filter left among the kernel messages, do the following:

Linux specific

On a Linux FSE client, to see the messages or debug trace information the HSM file system filter left among the kernel messages:

```
# dmesg | grep hsmfs
```

Windows specific

To see these messages on a Windows client, proceed as follows:

1. In Windows Control Panel, double-click **Administrative Tools**, and then double-click **Event Viewer**.
2. Click **System Log** and under the Source column look for `hsmfs` source.

HSM File System Mounting Problems

HSM File System cannot be Mounted

Linux specific

On a Linux FSE client, there is an entry in the `/etc/fstab` file for the `/fse/hsmfs_01` file system with type `hsmfs`. Running the `mount /fse/hsmfs01` command produces an error:

```
# mount /fse/hsmfs_01/
mount: wrong fs type, bad option, bad superblock on
/dev/vg_fse/hsmfs_01, or too many mounted file systems
```

This error is reported if any of the following is true:

- FSE daemons are not running.
- There is no FSE partition configured for the specified HSM file system or logical volume.
- The **Major:Minor** parameter in the FSE partition configuration is incorrect.

Determine the cause of this error and act accordingly:

- Check if FSE is running using the `fse --status` command; if needed, start the FSE services using the `fse --start` command.
- Check if the partition for the mount point is configured in the FSE installation.

The logical volume is configured with its **Major:Minor** parameter specified in the FSE partition configuration file. You need to verify this parameter matches the actual **Major:Minor** parameter assigned to the logical volume (file system). For this purpose, run the following commands and compare their outputs; the retrieved **Major:Minor** parameters should be the same:

```
# fsepartition --show PART_01 | grep FileSystemID | sed
's/FileSystemID = "\\(.*)"/\1/'
# ll /dev/vg_fse/hsmfs_01 | awk '{print $5 $6}'
```

If the numbers displayed are not the same for both commands, see the *FSE Installation Guide* for information on proper logical volume and partition configuration.

Note: There are also a number of useful HSM file system messages that you can see by running the command `dmesg | grep hsmfs`. Check the output of the `lsmod | grep hsmfs` command to determine if the HSM file system module is loaded and how many processes are using it.

Windows specific

On a Windows FSE client, while running the `fse --mount` command to mount an HSM file system, the command may fail and will report an error.

The most probable cause for this problem is that NTFS has already attached to the disk volume that stores the HSM file system, and the HSM file system filter cannot gain control over it. Run the following command to detach NTFS and then retry mounting the HSM file system:

```
C:\>fse --dismount-ntfs VolumeName
```

Another possible cause for the problem is that the mount point directory for the HSM file system is not empty. In this case, delete the directory contents and try to mount the file system again.

Partition 'ready' or 'mounted' but cannot be Accessed

Regardless of the operating system, the output of the `fsepartition --list` command shows the status of the corresponding `fse-pm` and `fse-hsm` processes, and not the status of the corresponding HSM file system. If the `fse-pm` process is running, the status is `ready`. If the `fse-hsm` process is running, the status is `mounted`. However, the `fse-hsm` process should normally be running only when the HSM file system is mounted.

To see the detailed status of the partition and its corresponding HSM file system, use the following command:

```
fsepartition --status PartitionName
```

Linux specific

On a Linux FSE client, if the file system is not mounted, you can mount it manually using the following command. The command requires that the HSM file system has an appropriate entry in the `/etc/fstab` file:

```
mount /fse/hsmfs_01
```

For details, see “[Mounting on Linux](#)” on page 60.

Windows specific

On a Windows FSE client, you may experience the following symptoms:

- The HSM file system is mounted, but the Hierarchical Storage Manager is not running.

In such a case, proceed as follows:

Check that the state of the FSE partition on the FSE server is `ready` (and not disabled or stopped) using the command:

```
C:\>fsepartition --status PartitionName
```

- `fse.log` reports that no HSM file system is ready for host CLIENT on the respective FSE server.
 - Check that client host is accessible from the FSE server with the same fully-qualified domain name as it is accessible locally.
 - Check the DNS suffix:
 1. On the Windows desktop, right-click **My Computer**, then right-click **Properties** and select **Network Identification**.
 2. Click **Properties**, and then click **More...** and enter the suffix.
- `fse.log` reports that a volume is configured, but no mount point is defined.

When such a warning is reported and the `mountvol` command shows the line `*** NO MOUNT POINTS ***` for that volume, proceed as follows:

- In case you have not yet defined the mount point, define the mount point with the command `fse --mount command`; see “[Configuring an FSE partition for a Windows FSE client](#)” on page 56.
- If you have already defined the mount point, check its volume ID, written in the `%InstallPath%\var\log\fse.log`. Verify that the volume ID, written in the FSE server partition configuration file, which you specified before installing a Windows client, matches the volume ID of the configured FSE volume on the respective client. For more information on required volume ID format, see “[Configuring an FSE partition for a Windows FSE client](#)” on page 56.

FSE Media Management Problems

Media Initialization Fails with Large Block Size

Windows specific

Media initialization fails with:

```
ERROR: Medium 052007, volume 3: Init failed (12053: SCSI driver error. 'Buffer too large. See error.log for details.').
```

This happens because some Windows host bus adapter (HBA) drivers (particularly Adaptec) do not support large block sizes by default. You can enlarge the scatter/gather list value to increase the transfer size beyond 64 KB.

You need to change the registry setting that allows the lists to be expanded to up to 255 elements. Manually update the registry as follows.

1. Run **REGEDIT32** to edit the system registry.
2. Go to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > *DriverName* > Parameters > Devicen** (where the *DriverName* is the name of the miniport driver, for example AIC78XX, and the *n* in *Devicen* is the bus number assigned at initialization).
3. Go to **Edit > Add Value** for **REG_DWORD** Data Type and enter **MaximumSGList** for value. Enter **FF** for Value data (FF is equal to 255) and make sure the **Hex** option is set.
4. Exit the Registry Editor and restart the system.

Media Format Fails with Message “illegal request - No MIC detected”

An AIT medium cannot be formatted if it does not contain MIC or it is corrupted. The format operation fails with the error:

```
"Illegal Request ([0x83,0x02] AIT: No MIC detected.)"
```

FSE can use such a medium as a “single-volume” medium, similarly to an LTO medium. To prepare it for use, you must initialize it without formatting it as follows:

```
# fsemedium --init Barcode
```

FSE Migration Problems

Migrations all Waiting for Disk Buffer

Regardless of the operating system, if jobs are aborted, killed or they die for reasons unknown, it can happen that the disk buffer file they created will remain on the system, using up disk space, and will never be removed by FSE. The file `fse.log` will show jobs reporting Disk buffer full. They will be in a waiting state and none of them will start writing to tape.

If this happens, the solution is to stop FSE using the `fse --stop` command, remove all the files in the `/var/opt/fse/diskbuf` directory on Linux or in `%InstallPath%\var\diskbuf` on Windows, and restart FSE using the `fse --start` command.

fse-hsm Goes Down if all I-nodes on the HSM FS are Used

Linux specific

If all I-nodes on the HSM file system are used it could happen that `fse-hsm` is stopped. If this happens, you should extend the HSM file system using the command `e2fsadm`.

FSE Recall Problems

Recall Job does not Continue if Medium is Marked Unusable During the Job

If several recalls are waiting for the same resource (medium), which in the meantime has been marked as unusable, all recall jobs associated to this medium will permanently wait for resources. This is because the medium/media for a recall job is determined before the resource allocation is requested. The recall job is unaware of changes in a particular medium state if it should become unusable during this process.

All recall jobs that are waiting for a corrupted media must be aborted by the FSE administrator. If an additional copy of the unusable medium exists, then the recall job uses it instead of the unusable one the next time it is started.

Device Problem is Reported During a Failed Recall

When a recall fails and logs the error message (on Linux systems)

```
device busy
```

or (on Windows systems)

```
31: A device attached to the system is not functioning.
```

into the `fse-hsm_<PID>.log` file on the FSE client, try to identify the problem as follows:

- Determine the status of the corresponding FSE partition with the command:

```
fsepartition --status <PartitionName>
```

- Check if the `fse-hsm` process is running on the corresponding FSE client. Use the `ps -ef | grep fse-hsm` command (on Linux) or Windows Task Manager (on Windows) for this purpose.

If the partition status is set to any other value than `mounted`, and the `fse-hsm` process is running on the client, the problem may be caused by an improper configuration of the `omniORB.cfg` files. For information on which `omniORB.cfg` settings are correct and how to configure `omniORB.cfg` appropriately, see [“Reconfiguring FSE Interprocess Communication”](#) on page 62.

FSE Backup and Restore Problems

Backup is Aborted, but its Job Hangs

Backup uses an archiving tool (`tar` on Linux and `fse-arch` on Windows) to pack all data that needs to be backed up. If the archiving tool has already been started and in the meantime a backup job is aborted, the job hangs until the tools completes the archiving. This can be time consuming, especially if FSE contains a lot of data to be backed up.

In such a situation, an FSE administrator must clean up the FSE server after the `abort` command. To clean up the FSE server, proceed as follows:

1. On the FSE server system, terminate the auxiliary processes that are involved in the backup process.

Linux specific

Using the `kill` command, terminate all processes named `fse-dd` or `tar`.

Windows specific

Using Windows Task Manager, end all processes named `fse-arch`.

2. Run `release_snap.py` and follow the instructions it displays. The script will remove obsolete volume snapshots.

The script is located in the `/opt/fse/sbin/tools` (on Linux systems) or `%InstallPath%\bin` directory (on Windows systems). Note that `%InstallPath%` defaults to `%ProgramFiles%\Hewlett-Packard\FSE`, but it may be different according to your choice in the FSE installation wizard.

Backup does not Start because of Previous Unfinished Backup Jobs

Any unfinished backup job will prevent a new backup to be started. The FSE administrator must clean up the system using the procedure described in case of a hanging backup job (see the previous section) before a new backup job can be started.

FSE Start-up and Shut-down Problems

The `fse --stop` Command does not Finish and Leaves FSE Processes Running

After running the `fse --stop` command to shut down FSE, you may experience the following situation:

- The `fse` command does not finish (does not return to the command prompt) in a reasonable time.
- Local HSM file systems are still mounted.
- Local FSE processes are still running.

If a pre-stop script has been configured on this system, the above symptoms may indicate problems with commands in the script. Most probably one of the script commands has not finished. If so, identify the problematic command and reconfigure the script, as follows:

Linux:

For each command that is specified in the pre-stop script (the `pre_stop.sh` file located in the local `/opt/fse/sbin` directory), run the following command on the local system:

```
ps -ef | grep <CommandName>
```

This will help you identify which command caused the shut-down problem.

Windows:

If debugging of the FSE processes is enabled, the `fse-svc` process logs the invocation of each command specified in the pre-stop script. In this case, identify the problematic command by inspecting the `fse-svc_<PID>.log` debug file located in the local `%InstallPath%\var\log\debug` directory. Otherwise, identify the command by examining the process list in Windows Task Manager.

For information on configuring the pre-stop script, see the *FSE Installation Guide*.

Communication Problems

IP Address of an FSE Installation has Changed

When changing the IP address of an FSE installation, proceed as follows:

1. Stop the FSE processes with `fse --stop`.
2. Stop the omniORB naming service with `omniNames --stop`.
3. Remove all files in the `/var/opt/fse/omniorb` directory (on Linux) or in the `%InstallPath%\var\omniorb` directory (on Windows).
4. Change the system's IP address using the operating system tools.
5. If the `omniORB.cfg` file is configured, change its contents appropriately. For more information, see [“Reconfiguring FSE Interprocess Communication”](#) on page 62.
6. Start the omniORB naming service and FSE processes with `fse --start`.

CORBA Cannot be Initialized

If the following error is reported after invocation of an FSE command, verify the status of the CORBA Naming Service daemon (`omniNames`):

```
6000  Cannot initialize CORBA ORB. 'CORBA::COMM_FAILURE'
```

On the FSE server, check if `omniNames` is running. Invoke the `omniNames --status` command for this purpose, and restart it using the `omniNames --restart` command, if needed.

On the FSE client, if the `error.log` file reports “Failed to initialize CORBA”, most probably the FSE server is not accessible. Proceed as follows:

- On the FSE client, verify the settings specified in the `omniORB.cfg` file located in the `%InstallPath%\etc\` directory. Note that `%InstallPath%` defaults to `%ProgramFiles%\Hewlett-Packard\FSE` but may be different according to your choice in the FSE installation wizard.
- Check the network connection between the FSE client and the FSE server.

CORBA Naming Service Cannot be Accessed

When `omniNames` (CORBA Naming Service daemon) is not running on FSE, the following error is reported:

```
6001 Problems accessing CORBA Naming Service.
```

Check if `omniNames` is running by inspecting the output of the `omniNames --status` command. If `omniNames` is stopped, proceed as follows:

1. Stop the FSE daemons/services with the `fse --stop` command.
2. Start the `omniNames` naming service and FSE daemons/services using the `fse --start` command.

Connection to Management Interface Fails

If the following error is reported after running an FSE command, you should verify the status of the `fse-mif` process:

```
6008 Cannot connect to Management Interface - probably not
running.
'Requested name does not yet exist in the NameService.
Giving up. FSE/mgr/<host@domain@com>/mif'
```

Check if the `fse-mif` process is running using the `fse --status` command. If the process is stopped, you must restart FSE with the `fse --restart` command.

Wrong Network Configuration

Improper network configuration can cause improper FSE system operation. If the network is not configured properly to resolve its host name to the fully-qualified domain name (FQDN) and IP, one of the following warnings can appear in the FSE error log:

```
This machine (localhost.localdomain) has only localhost IP.
```

```
Misconfigured DNS on host: localhost.localdomain. Official name
(hostent::h_name) is NULL.
```

Misconfigured DNS on host: localhost.localdomain. Official name (hostent::h_name) should contain FQDN.

To resolve this problem, check if the network settings (DHCP settings, the `/etc/hosts` file configuration (on Linux), DNS resolution configuration, and so on) are correct, and modify them if necessary.

fsefile Fails with Error Message “6004 Caught Corba exception”

If, after running the `fsefile` command with any of the following options: `--history`, `--recall`, `--migrate`, `--trigger-migration`, the command fails and reports

```
6004 Caught Corba exception.
```

try to identify the problem as follows:

- Find out the status of the FSE partition with the command:

```
fsepartition --status <PartitionName>
```

- Check if the `fse-hsm` process is running on the corresponding FSE client. Use the `ps -ef | grep fse-hsm` command (on Linux) or Windows Task Manager (on Windows) for this purpose.

If the partition status is set to any other value than `mounted`, and the `fse-hsm` process is running on the client, the problem may be caused by an improper configuration of the `omniORB.cfg` files. For information on which `omniORB.cfg` settings are correct and how to configure `omniORB.cfg` appropriately, see “[Reconfiguring FSE Interprocess Communication](#)” on page 62.

Firebird Server Problems

Firebird is third-party server software used by the Resource Manager for managing resources in FSE. Firebird software must be installed on the system before you can start FSE processes. For instructions on how to install the Firebird package, see the *FSE Installation Guide*. You can find more detailed information on Firebird on the web site <http://firebird.sourceforge.net>.

If Firebird is not installed or running, the following error is reported:

```
ERROR: Firebird SuperServer (Interbase) is not up and running!
```

If you receive this error, proceed as follows:

Linux:

Check if the Firebird server is running with the following command:

```
# /etc/init.d/firebird status
```

If it is running, the output should look like this:

```
ibserver (pid 896 895 581 577 576) is running...
```

If the Firebird server is not running, start it with the following command:

```
# /etc/init.d/firebird start
```

Windows:

Check if the Firebird server is running using the Windows Services tool:

- In Windows Control Panel, double-click **Administrative Tools**, and then double-click **Services**.
- Look for the Firebird Server. If it is not running, right-click it, and then click **Start**.

SCSI Problems

Various SCSI Errors

SCSI errors that look like those listed below usually indicate hardware problems. FSE only reports them to show the status returned from the hardware layer.

```
[<time>, <host>, BEA:SCSI, <drive>, 0, id:0:0:2:0]
ERROR: SCSI host status code: <code>
[<time>, <host>, LAS:SCSI, <library>, 0, id:0:0:5:0]
ERROR: Hardware Error ([0x40,0x91] Diagnostic failure on
component 145)
[<time>, <host>, BEA:SCSI, <drive>, 0, id:0:0:2:0]
ERROR: Hardware Error ([0x83,0x09] AIT: MIC write fault.)
[<time>, <host>, LAS:SCSI, <library>, 0, id:0:0:1:0]
ERROR: Medium Error ([0x52,0x00] Cartridge fault)
```

Check `/var/opt/fse/log/fse.log` and `/var/log/messages` on Linux, or `%InstallPath%\var\log\fse.log` and System Event Log on Windows, for SCSI subsystem error diagnostics to get an idea of the problem and act accordingly.

If the errors occur after changing hardware system configuration (such as a different cable connection), make sure that the device files used for library and drive configuration are correct.

You should also physically check the SCSI connections (cables, terminators) and make sure that all the connectors are correctly and firmly connected.

Blocked Access to a SCSI Device

On a Windows consolidated FSE implementation or a Windows FSE server, if you experience major problems with FSE drives, check the FSE error log for the following message:

```
Access to SCSI device is prevented. Make sure that the device
driver is disabled.
```

If this error message is logged, the cause of the problem with the FSE drive is the operating system driver for the tape drive, which has not been disabled.

For information on disabling the device drivers for tape drives, see the *FSE Installation Guide*.

Tape Library Problems

The Tape Library Reports the Same Barcode in Different Locations

It could happen that the tape library reports the same medium barcode in a drive and in a slot at the same time. In this case FSE writes an error message to the `error.log` file, and marks both the medium and the drive as unusable. This happens when the FSE administrator manually moves the medium from the drive to the slot, and then starts the library inventory rescan.

To resolve this situation, an FSE administrator has to move the medium back to the same drive manually and clear the slot. Another option is to clear the drive and inventory the slot to which the cartridge was moved. Note that a full inventory scan is not recommended because it can be very time-consuming.

FSE Troubleshooting Tools

FSE provides a set of commands and scripts called FSE tools, designed to perform advanced low-level FSE servicing tasks. These commands communicate directly with FSE daemons/services and FSE agents, and are usually used for low-level modifications and more extensive monitoring or troubleshooting tasks.



Caution: Wrong use of commands or scripts can cause data corruption!

FSE tools are located in the `/opt/fse/sbin/tools` directory on Linux and in `%InstallPath%\bin` on Windows.

The following table lists commonly used FSE tools with a brief description:

Tool name	Description
<code>fse_sysinfo.sh</code>	Collects important system information about the system the FSE software is running on (Linux only).
<code>fse_cleantmp.sh</code>	Helps clean up the temporary FSE disk buffer and FSE debug files (Linux only).
<code>fse_net</code>	Performs the following tasks for network-communication debugging inside FSE, according to supplied options: <ul style="list-style-type: none"> ■ Resolves host names to an IP number and vice versa. It shows for a specified host name or IP number its fully qualified domain name (FQDN), name aliases (Linux only), address type, and IP number. ■ Verifies the connection from an arbitrary FSE host to the CORBA naming service running in FSE. ■ Verifies the connection between the FSE server and an arbitrary FSE client.
<code>fse_hsm</code>	Retrieves statistical data from the Hierarchical Storage Manager (HSM) process.
<code>fse_hsmattr</code>	Lists HSM attributes stored in HSM file system Extended Attributes.
<code>fse_hsmdb</code>	Navigates the HSMDB.
<code>fse_la</code>	Passes manual commands to the Library Agent (FSE server only).
<code>fse_scsi</code>	Performs low-level SCSI commands on tape drives and media changers (FSE server only).
<code>fse_df</code>	Parses the FSE data format (stored on FSE media, Fast Recovery Information) (FSE server only).

Tool name	Description
<code>fse_pmjob</code>	Prints a list of active migration, release, and recall jobs on a specified FSE partition, together with their current statuses (FSE server only).
<code>fse_fsc</code>	Navigates the File System Catalog (FSC) database (FSE server only).
<code>Dumpprmdb.sh</code>	Dumps the Resource Management Database into the <code>rmdb.dump</code> file, located in the <code>/var/opt/omni/sbin/tools</code> directory (on Linux FSE servers) or <code>%InstallPath%\bin</code> (on Windows FSE servers).

Debugging FSE Processes

There are two ways in which debugging of the FSE processes can be enabled:

Configure the `trace.cfg` file by setting the values of the appropriate debugging variables (options), and then restart the FSE processes.

This way, debugging will be enabled for all subsequent FSE sessions. `trace.cfg` is located in the `/etc/opt/fse/` directory (on Linux systems) or in `%InstallPath%\etc` (on Windows systems).

Run the `fssystem --enable-dbg` command with the appropriate debugging options while FSE processes are running.

This will enable debugging for the currently running session only. If FSE is restarted, the debugging mode will fall back to the settings specified in the `trace.cfg` file.

The following is the syntax of the `fssystem --enable-dbg` command:

```
# fssystem --enable-dbg SysDbgLevel
                        [SysDbgFile [SysDbgFlags [SysDbgHSMFS]]]
```

Debugging Options

Most of the following options can be specified in the `trace.cfg` file as well as in the command line for the `fssystem --enable-dbg` command.

HSM_DEBUG_LEVEL

This variable represents the level of detail that FSE processes will track in their debug files. Note that each debugging level includes all debugging information from lower levels.

The following table represents the values the HSM_DEBUG_LEVEL variable can take, their corresponding levels, and their descriptions.

Option value	Debugging level	Description
L	1	Low —very little trace information will be logged in files, producing less debug overhead and small debug files.
N	2	Normal —the usual level of detail in debug traces.
A	3	API —trace third-party API calls (CORBA, Firebird...).
D	4	Detail —trace all process details, which can produce a lot of debug output.

Example:

```
HSM_DEBUG_LEVEL=A
```

HSM_DEBUG_FILE

The following table represents the values the HSM_DEBUG_FILE variable can take and their descriptions.

Option value	Description
" "	(empty string or variable commented out; default value) Debugs will be written to the standard output <i>stdout</i> .
.	(dot) Debugs will be written to files (one per FSE process) in the directory: /var/opt/fse/log/debug on Linux systems, or %InstallPath%\var\log\debug on Windows systems. Each FSE process will use a unique filename of the form: <prog>_<pid>.log.
<FileName>	All FSE processes will print their trace data into the file: /var/opt/fse/log/debug/<FileName> on Linux systems, or %InstallPath%\var\log\debug\<FileName> on Windows systems.

It is recommended to use the value . (dot), which results in each process producing its own trace file. The other two cases (" " and <FileName>) are for special purposes only and should not be used unless required by HP support personnel.

Example:

```
HSM_DEBUG_FILE=.
```

Debugs can occupy a lot of disk space when troubleshooting an unpredictable problem. On Linux, if you need more disk space than available in the `/var/opt/fse/log/debug` directory, change your working directory to `/var/opt/fse/log` and create a symbolic link to a location with more disk space:

```
# mkdir /tmp/BIG_DISK
# cd /var/opt/fse/log
# rmdir debug
# ln -s /tmp/BIG_DISK debug
```

HSM_DEBUG_FLAGS

Select the detailed information you want to log in the debug trace files. Use multiple letters if you want different items to be logged:

Option value	Description
I	Function In & Out —log FSE function entries and exits. Note that this type of logging will create large debug files.
N	No Header —print no header in the trace output line.
C	Compact —print compact headers in the trace output line. This is the recommended type of logging.
T	Exception Throw —log exception throwing events in the trace output.
E	Error Logging —log errors in the trace output also. By default, errors are only logged to the file: <code>/var/opt/fse/log/error.log</code> on Linux systems or <code>%InstallPath%\var\log\error.log</code> on Windows systems.
V	Verbose third-party debugs —produce verbose omniORB debugging information.
O	Log only the specified level —cause the debugging level specified by the <code>HSM_DEBUG_LEVEL</code> variable to omit the debugging information from the lower levels (which would normally be included).

Example:

```
HSM_DEBUG_FLAGS=CTE
```

HSM_DEBUG_FS

The `HSM_DEBUG_FS` variable is a numeric value composed of several HSM file system filter debug flags. Set the flags if you want to trace specific information about the HSM file system filter driver.

The following tables list the flags (big flags) that can be set:

Linux:

Flag value	Description
0x00000001	main filter execution flow
0x00000002	file system filter
0x00000004	tracing of VFS methods
0x00000008	pseudo device communication
0x00000010	logging of events
0x00000020	HSM attributes
0x00000040	locking
0x00040000	tracing of VFS methods (page cache)
0x00080000	pseudo device communication - detail tracing
0x00100000	logging of events - detail event structure dump
0x00200000	inode info/HSM attributes - detail tracing
0x00400000	locking - detail tracing
0x40000000	tracing of Invisible IO (all operations that are hidden)
0x80000000	tracing of all VFS methods (even those that are not needed)

Windows:

Flag value	Description
0x00000001	main filter execution flow
0x00000002	file system filter
0x00000004	tracing of FS methods
0x00000008	device communication
0x00000010	logging of events
0x00000020	file object info
0x00000040	locking
0x00000080	HSM attributes/structures

Example:

```
HSM_DEBUG_FS=0x0000007f
```

HSM_DEBUG_DISKBUF

This debugging variable cannot be used with the `fssystem --enable-dbg` command. It can only be specified in the `trace.cfg` file.

Use this variable if you want to control what happens to the disk buffer files after they have been migrated to media, or after the recall process is completed.

The following tables list the values that the variable can take:

Option value	Description
none	Do not leave any disk buffer files (default).
corrupt	Leave only corrupted disk buffer files on disk.
all	Leave all disk buffer files on disk.

Example:

```
export HSM_DEBUG_DISKBUF=corrupt
```

HSM_USE_MIC

This debugging variable cannot be used with the `fssystem --enable-dbg` command. It can only be specified in the `trace.cfg` file.

The `HSM_USE_MIC` option is designed to ease debugging related to Memory-in-Cassette (MIC) chips, which are integrated in the AIT tape cartridges. You can prevent FSE from using the MIC functionality by specifying the option as follows:

```
export HSM_USE_MIC=no
```

HSM_MEDIA_LOGGING

This debugging variable cannot be used with the `fssystem --enable-dbg` command. It can only be specified in the `trace.cfg` file.

Specifying this variable, you can prevent FSE from logging the actions performed on the FSE media, which are normally logged into the FSE event log. Use the following line to disable media logging:

```
export HSM_MEDIA_LOGGING=no
```

Note: Excessive logging may result in a FSE system performance becoming noticeably degraded. It may be a good idea to disable media logging when you expect that many media operations will be logged in a short period of time.

HSM_CALL_TIMEOUT

This debugging variable cannot be used with the `fssystem --enable-dbg` command. It can only be specified in the `trace.cfg` file.

In the FSE interprocess communication, the default timeout period for all omniORB (CORBA) calls is 2 days. For various reasons, you may want to change the default period to a different value. The timeout value must be specified in milliseconds. Value 0 means no timeout.

Example:

```
export HSM_CALL_TIMEOUT=300000
```

HS_PT_DIRECT

By default, the FSE system uses direct SCSI pass-through mode for controlling SCSI devices. Some SCSI and Fibre Channel host bus adapters (HBAs) are not compatible with this mode, which may result in problems using the attached devices. The variable *HSM_PT_DIRECT* in *trace.cfg* allows such HBAs to be used in the FSE system. You can switch communication with devices from direct SCSI pass-through mode to buffered SCSI pass-through mode.

To use buffered SCSI pass-through mode, uncomment the following line in *trace.cfg*, or add it if necessary:

```
# export HSM_PT_DIRECT=no
```

Debugging Backup and Restore Jobs

If you want to debug backup and restore jobs, use the following procedure:

1. Restart FSE services on the FSE client and keep the console open. The console will report all client backup and restore debug info.
2. Start the *fsebackup* or *fserestore* script with the *-dbg* option. This will produce additional output in the console where script is started, and in the console where FSE was started on the client (see the previous step).
3. Backup and restore progress can be tracked with fewer details in the files *backup.log* or *restore.log*, located in the */var/opt/fse/log* directory on Linux, and in *%InstallPath%\var\log* on Windows.

Potential Problems with Debugging

How do I check the debugging level of the HSM file system filter?

On Linux, use the command `dmesg | grep hsmfs_debug` to list the HSM file system filter debugging level.

On Windows, check the System Event Log (**Start > Settings > Control Panel > Administrative Tools > Computer Management > Event Viewer**).

On a Linux system, the HSM file system filter debugging level is always 0x0000 after restarting the system.

If the HSM file system partition is not listed as `noauto` in the `/etc/fstab` file, the HSM file system filter will be loaded at boot time with the debugging level set to `0x0000` regardless of the settings specified in the `trace.cfg` file.

1. Add the following line into the `/etc/modules.conf` file:

```
options fsefs fsefs_debug=0x007f
```

where `0x007f` is an example of the specified debug level.
2. Restart FSE for the changes to take effect.

Backup or restore debugging results in job failure.

Backup and restore debugging using the `-dbg` option does not work with the `fssystem --enable-dbg` triggered process debugging.

Make sure you turn on the FSE processes debugging by setting the `trace.cfg` file and restarting FSE.

Contacting Support

Gathering Information About Your Problem

You need to gather specific information needed for troubleshooting an FSE installation. Technical support needs to know exactly what your problem is and as much information about it as possible. You should describe if any changes were made to the environment before the problem occurred; include software version numbers, the nature of the problem (whether the problem occurs intermittently, what errors are being observed), and a list of third-party vendors.

Checklist for required information

When collecting data about the problem, follow this list:

Problem description

Think about the keywords of your problem. Write this down as a one-line description. Defining what the problem is is very important, because technical support will be focused on solving that precise problem. If you specify the wrong problem, you might end up with the original problem still present after the support put a lot of effort into solving something else.

Log and debug traces of the problem

Reproduce the problem with a debugging set—for details, see “[Debugging FSE Processes](#)” on page 232. Save the evidence, such as wrong command output or strange error messages, and send it for lab investigation. The list of files that must be sent must be changed to `/var/opt/fse/log` on Linux and `%InstallPath%\var\log` on Windows.

Note that without knowing the contents of the `fse.log` and `error.log` files, technical support cannot start investigating the problem.

Reproduction procedure

Write down the procedure for reproducing the problem. This is important for technical support to be able to investigate and solve the problem.

Workaround discovered?

If you managed to develop your own workaround, describe it and add the description as a part of your problem report. It might be a useful hint to technical support and other users, or it can lead to an even better solution.

System information

On Linux, use the `/opt/fse/sbin/tools/fse_sysinfo.sh` script to collect data about the system the FSE software is running on. The output is shown on `stdout`, and stored in the file `/var/opt/fse/log/<hostname>_info.log`. Send this file and the contents of the directory `/var/opt/fse/cfg` to technical support.

On Windows, use **Start > Programs > Accessories > System Tools > System Information** to collect data about the system. From the **Action** menu, select the **Save As System Information** option and send the file to technical support.

One archive package

Pack everything collected in previous steps in a tar/zip file. It is easier to handle one file per problem. This way you make sure technical support receives everything you provide for problem investigation.

Contact

See “[HP Technical Support](#)” on page 12.

Technical support will contact you as soon as possible, depending on the data you provide and especially on the problem you reported.

FSE Error Codes

Code	Description
0	No error
1	Singleton violation: Class instance already exists
2	File access error
3	Out of memory
4	Invalid arguments passed to a function
5	MAJOR INTERNAL ERROR: Data corruption detected
6	Functionality not yet implemented
7	String too long
8	Argument missing
9	Shouldn't happen at all
10	Invalid file name
11	Mutex usage error
12	Environment variable not set
13	UUID cannot be parsed
14	Timeout expired
15	Pointer is NULL
16	String contains spaces at the beginning or at the end or contains new lines
17	Unknown Type
18	std::exception
19	Aborted
20	Operation is not supported
21	Maximum size exceeded
22	Precondition check failed
23	Process execution problem
24	Not found
25	Network connectivity problem
26	Internationalization problem
999	FATAL ERROR
1000	Resource database error
1001	Invalid foreign key

Code	Description
1002	Duplicate unique value
1003	Problem with connection
1004	Not found
1005	Duplicate drive index
1010	Disk buffer allocation error
2000	Cannot insert object
2001	Cannot find object
2002	List is empty
2003	Wrong method call
2004	Event is empty or bad
2005	Object exists
2006	Object is NULL or is not created
2007	I-node is 0
2008	Device file is not found or device file name is bad
2009	No job list or list is empty
2010	Hash is zero Cannot create HSM cache
2011	File has no file ID
2012	One or more configuration parameters are wrong
2013	Read error
2014	Write error
2015	Cannot get mount event
2016	STL container ERASE method failed
2017	Wrong list position
2018	No file name
2019	Job indexes do not match
2020	No migration file object
2021	File is not on expected list
2022	Path is not valid
2023	I-node is locked too long
2024	Entry is not dirty
2025	File is locked by I-node
2026	Invalid FS entry type

Code	Description
2027	Job index exceeds vector size
2028	Job element index exceeds vector size
2029	HSM is not ready to handle events from filter
2030	Premount event is waiting for same device
2031	No premount event is waiting for device
2032	Mount point is already defined
2033	Invalid job status
2034	HSM FS recovery error
2035	Volume UUID not found.
2036	HSM already started. Unmount first.
2037	Volume is disabled.
2038	NTFS is attached without HSMFS filter.
2099	Other HSM errors
3000	Parser error
3001	Unknown variable encountered in validation
3002	Target class does not have any subgroups
3003	Input value not valid in conversion
3004	Type not found in table of possible types
3005	Missing entry for mandatory variable
3006	Entry parsed too many times
3100	Not enough parameters for option or command
3101	Syntax error: cannot resolve argument
3102	Missing option
3201	Internal error! Invalid argument type. Check apcpp.
3202	Internal error! Same option already exists.
3203	Internal error! Invalid group type.
4000	Not enough space to store whole record to file
4001	Data cannot fit in record
4002	Invalid relative file header
4003	Invalid relative file record index
4004	Invalid vector size
4005	Transaction is not started

Code	Description
4007	Duplicate relative file ID
4008	Transaction object missing
4009	Invalid alteration record value
4010	Invalid end transaction structure version
4011	Invalid start transaction structure version
4012	Invalid alter structure version
4013	Record is not end transaction structure
4014	Record is not start transaction structure
4015	Record is not alter structure
4016	Invalid transaction
4017	Relative file transaction must have pointer to DB
4018	Too many transaction instances
4019	Invalid DB info header
4020	Reserved space in journal file was filled by a transaction
4021	Unused vector is not in buffer
4022	Invalid unused vector structure
4023	Invalid header position
4024	Invalid circular file header
4025	Invalid journal file header
4026	Read after the end of relative file Index overflow
4027	Journal file missing.
6000	Cannot initialize CORBA ORB
6001	Problems accessing CORBA Naming Service
6002	Cannot resolve object name to reference
6003	Cannot register object reference in Naming Service
6004	Caught Corba exception
6005	Object that is registered in NS is still active
6006	Unknown CORBA exception
6007	Unknown FSE exception
6008	Cannot connect to Management Interface—probably not running
6009	Cannot setup UI message server
6999	FATAL IPC ERROR: Cannot continue

Code	Description
7000	Data block is full
7001	Error performing ASCII <-> UTF-8 conversion
7002	Invalid block contents detected
7003	Block already closed
7004	End of data detected
7005	Invalid state of Data Format library detected
7006	Invalid record sequence
7007	Invalid block magic cookie
7008	Invalid record magic cookie
7009	Invalid block sequence number
7010	Invalid record sequence number
7011	Byte stream in invalid format
7012	Corrupted volume header
7013	BLKMGR: Writer already running No more readers allowed
7014	Byte stream size mismatch
7015	Rethrown error from other thread
7016	Limited-stream missing
7017	Broken pipe
8004	Problems with Library Agent of this library
8005	Problems with Back End Agent
8006	Problems with Resource Manager
8007	Packer error
8008	Problems starting/stopping partition managers from MIF
8009	Remove Error
8010	Update Error
8011	Select Error
8012	Semantic Error
8013	Syntax Error
8014	Unknown Status
8015	Medium has FSE or ECMA volume header
8016	Medium has more than one volume
8017	Medium does not belong to any media pool

Code	Description
8050	Configuration file not recognized
8051	Configuration revision out of range (0–MaxCfgRevs) Could not store revision file in CfgDB repository
8052	This configuration revision does not exist
8053	Problem handling configuration files inside CfgDB repository
8054	There is already resource with same name configured
8055	Renaming objects in RMDB is not allowed yet
8056	Illegal name specified for an FSE object
8057	Semantic error in configuration file
8058	Syntax error in configuration file
8059	Erroneous configuration file header
8060	Error managing CfgDB
9000	Job type not valid or missing
9001	No resources found
9002	Partition already registered
9003	Cannot find partition
9004	You want to release more drives than actually allocated
9005	Job not found
9006	Table size not valid
9007	Migration ID generator should be initialized first
10000	Job ID does not exist
10001	Configuration error
10002	FSC did not return any set of copies for specific file ID
10003	Job was removed
10004	Job failed
10005	Not enough space for disk buffer
10006	Registration of HSM is not allowed
10007	Volume not found
10008	No files packed for migration
10010	HSM not registered
10011	HSM registered, but not accessible
11001	Blank check

Code	Description
11002	End of data
11003	Early end of medium
11004	Filemark/setmark detected
11005	Medium overflow (physical EOM)
11006	Medium protected
11007	Incorrect block length
11008	Medium error
11009	Hardware error
12000	Invalid SCSI status
12010	Critical sense key
12011	Device not ready
12014	Unit attention Medium changed or target reset
12022	Illegal request
12023	No medium in drive
12024	Move failed: Destination full
12025	Move failed: Source empty
12050	Retryable error (device busy or command aborted)
12051	Error opening SCSI device
12052	Error closing SCSI device
12053	SCSI driver error
12054	SCSI host error
12055	SCSI invalid result
13000	Invalid data type
13001	Split information is wrong
13002	Too many copies in the same generation
13003	No copies found
13004	Media volume number exceeds limit
13005	No media volume file found
13006	Successive change of data
13007	Number of generations does not match
13008	No data
13009	No NSC element object

Code	Description
13010	FSC is busy Possible: Reply from journal in progress
14000	Invalid volume position
14001	Invalid volume format (not FSE)
14002	Invalid volume ID
14003	Invalid device type (expected TAPE)
14010	Invalid size of parameter in MIC
14011	Invalid contents of parameter in MIC
14012	Initialization is not allowed
14013	Formatting is not allowed
14014	Hardware ID does not match
14015	WORM media/drive/pool mismatch
14016	Error while handling the FRI
15000	Invalid device type (expected MEDIA CHANGER)
15001	Invalid barcode in slot or drive
15002	Can't open tape library
15003	Drive index out of range
15004	Drive is not configured in the RMDB
15005	Mount request denied by user
15006	Unsupported tape library
16000	Error in HSMDB
17000	fsebak: Cannot perform suspend
17001	fsebak: Cannot perform continue
18001	Confirmation failed

Directory Layout



Directories created during the installation of FSE are presented in the following table. Note that the default Windows FSE installation path, `%InstallPath%`, is `Program Files\Hewlett-Packard\FSE`, but this can be changed during installation if Custom setup is chosen.

Linux path	Windows path	Contents
<code>/etc/opt/fse</code>	<code>%InstallPath%\etc</code>	Backup, trace and CORBA/omniORB configuration files
<code>/opt/fse/bin</code>		- currently not used -
<code>/opt/fse/doc</code>	<code>%InstallPath%\doc</code>	Readme files of the installed FSE hotfixes
<code>/opt/fse/sbin</code>	<code>%InstallPath%\bin</code>	Binaries of the FSE daemons, the CORBA/omniNames service binary
<code>/opt/fse/lib</code>	<code>%InstallPath%\bin</code>	Binaries of the FSE shared libraries and CORBA/omniORB libraries
<code>/opt/fse/lib/modules</code>	<code>%InstallPath%\bin</code>	Object files of the HSM file system module for various kernel versions
<code>/opt/fse/man</code>	<code>%InstallPath%\html</code>	Man pages/help files for the FSE commands
<code>/opt/fse/nls</code>		- currently not used -
<code>/opt/fse/newconfig</code>	<code>%InstallPath%\newconfig</code>	Templates of the FSE configuration files, the trace configuration file template, an empty skeleton for the Resource Management Database file
<code>/opt/fse/sbin</code>	<code>%InstallPath%\bin</code>	Binaries of the FSE commands, the omniNames command, the <code>fse</code> script
<code>/opt/fse/sbin/tools</code>	<code>%InstallPath%\bin</code>	Binaries of the FSE tools and scripts, the <code>nameclt</code> utility
<code>/var/opt/fse/cfg</code>	<code>%InstallPath%\var\cfg</code>	FSE configuration files of all configured FSE resources
<code>/var/opt/fse/dm/</code>	<code>%InstallPath%\var\dm</code>	FSE disk media
<code>/var/opt/fse/diskbuf</code>	<code>%InstallPath%\var\diskbuf</code>	Temporary data files of the FSE disk buffer

Linux path	Windows path	Contents
/var/opt/fse/fri	%InstallPath%\var\fri	Temporary data files of the Fast Recovery Information
/var/opt/fse/lock	%InstallPath%\var\lock	Synchronization (lock) files used by the FSE processes
/var/opt/fse/log	%InstallPath%\var\log	All FSE log files
/var/opt/fse/log/debug	%InstallPath%\var\log\debug	All FSE debug files
/var/opt/fse/omniorb	%InstallPath%\var\omniorb	Data files used by the CORBA/omniNames service
/var/opt/fse/part	%InstallPath%\var\part	File System Catalog and Hierarchical Storage Management database and associated journals for each configured FSE partition
/var/opt/fse/rmdb	%InstallPath%\var\rmdb	The Resource Management Database file
/var/opt/fse/tmp	%InstallPath%\var\tmp	Temporary files created and used by the FSE processes

Command Reference



B

This appendix contains FSE command-line reference pages available as UNIX man pages (*FSE man pages* on Linux systems), and Windows/HTML help (referred to as *HTML help for FSE commands* on Windows systems).

Note: FSE command-line reference is related to the final FSE release. FSE man pages and HTML help for FSE commands are updated whenever necessary, and may differ from the contents of this appendix.

Linux systems

To be able to access man pages on Linux, make sure that the MANPATH environment variable contains the `/opt/fse/man` directory.

- [Introduction to the FSE system \(fseintro\)](#), page 252
- [fse](#), page 257
- [fsebackup](#), page 265
- [fsebak](#), page 270
- [fsecheck](#), page 275
- [fsedrive](#), page 278
- [fsefile](#), page 283
- [fsejob](#), page 291
- [fselibrary](#), page 296
- [fsemedium](#), page 302
- [fsepartition](#), page 313
- [fsepool](#), page 322
- [fserecover](#), page 328
- [fserestore](#), page 332
- [fsesystem](#), page 334

The FSE command-line reference is related to the release. Manual pages and Windows/HTML help are updated whenever necessary, and may differ from this printed version.

Introduction to the FSE system (fseintro)

Description

HP StorageWorks File System Extender (FSE) is a mass-storage oriented software product (an HSM application), which provides a transparent, virtually infinite file system using tape libraries with high-capacity tape media for storing least accessed files and using a disk file system as a cache for recently used files. An FSE installation can be used, for example, as an online storage site or as a backup server.

FSE efficiently manages and stores extremely large quantities of data. Its storage capacity consists of a primary and a secondary storage space. These storage spaces present a successful combination of fast data access (disk storage speed) and economical data storage and protection (capacity and flexibility of tape media). Data archiving and retrieving can be customized and automated for each user of the FSE installation through logical partitioning. Each application or user can have a separate partition with its own set of rules. Policy-driven automated migration and recall processes make it possible for an FSE administrator to tune the system for best performance based on the type of data and the way applications handle that data.

FSE software is based on CORBA objects and CORBA's inherent client-server technology. It is implemented as a set of intercommunicating processes, which are responsible for particular FSE components. This way a sufficient modularity of the FSE installation is achieved. This modularity has been further extended, so that it is possible either to set up an FSE installation as a consolidated implementation (both client and server "all in one box"), or to distribute its functionality across one FSE server and an arbitrary number of FSE clients, or a mixed implementation of a consolidated set-up with further external FSE clients.

For information on FSE commands used for configuring, monitoring and managing the FSE installation, refer to the FSE man pages. For information about FSE features, functionality and principles of operation, refer to other product documentation items.

Commands

The FSE command-line interface is represented as a set of FSE commands. They belong to several command groups. On a distributed FSE implementation, some of these commands are available only on the FSE server (marked with an `s` character), some of them only on FSE clients (marked with a `c` character) and

some on both/all (marked with s +c characters). However, they are all present on a consolidated FSE implementation, so there is no special denotation for this kind of configuration.

Start-up and shut-down of FSE

- `fse (S + C)`

Initiate or terminate the FSE server or FSE client operation; show the current status of FSE daemons; prepare FSE for the restore process; configure mount points for HSM file systems (on Windows systems only); enable or disable Limited Access Mode for a particular HSM file system (on Windows systems only).

FSE configuration management

- `fssystem (S)`

Manage FSE system configuration; report state of FSE.

- `fsepartition (S)`

Manage configuration of the FSE partitions; scan the FSE partitions for obsolete data and report the corresponding statistics.

- `fselibrary (S)`

Manage configuration of the FSE libraries.

- `fsedrive (S)`

Manage configuration of the FSE drives.

- `fsepool (S)`

Manage configuration of the FSE media pools; report the FSE media pool related statistics gathered during the scan for obsolete data.

- `fsemedium (S)`

Manage configuration of the FSE media; format and initialize FSE media; perform maintenance tasks on the FSE media; duplicate or reorganize FSE media; recreate a copy of the unreadable data located on damaged FSE media.

Job monitoring and execution

- `fsejob (S)`

Monitor execution of running FSE jobs; show status of running FSE jobs.

File and file generation management

- `fsefile` (C)
Manage files on an FSE file system; recall older file generations from FSE media; add files to the migration candidate list; release files from an HSM file system; trigger migration on an FSE partition; trigger deletion of files from an HSM file system.

File system catalog check and recovery

- `fsecheck` (S)
Perform a File System Catalog consistency check against the HSM file system and FSE media.
- `fsererecover` (S)
Recover a File System Catalog from its redundant duplicate data sources (FSE media and Fast Recovery Information).

Backup and restore of the FSE installation

- `fsebackup` (S)
Back up a complete FSE system state to backup media.
- `fserestore` (S)
Restore a complete FSE system state from its backup.
- `fsebak` (S)
Achieve a consistent state of all HSMDDBs and FSCs to allow an eventual backup creation.

Files

There are certain files installed/created and used by the FSE installation that an FSE administrator must be familiar with. They are grouped together according to their purpose. On a distributed implementation, some of these files are present only on the FSE server (marked with an `s` character), some of them only on FSE clients (marked with a `c` character) and some on both/all (marked with `s + c` characters). Similarly, as with FSE commands, there is no special denotation for files on a consolidated FSE implementation.

FSE configuration file templates

- `/opt/fse/newconfig/*` (Linux)
`%InstallPath%\newconfig*` (Windows)
(S)
Templates for FSE configuration files that FSE administrators can modify and apply.

System configuration files

- `/etc/opt/fse/services.cfg` (Linux)
`%InstallPath%\etc\services.cfg` (Windows)
(S + C)
Configuration file with the fully-qualified domain name of the host on which the FSE server (with CORBA/omniORB Naming Service) is running.
- `/etc/opt/fse/backup.cfg` (Linux specific)
(S + C)
FSE backup configuration file.
- `/etc/opt/fse/trace.cfg` (Linux)
`%InstallPath%\etc\trace.cfg` (Windows)
(S + C)
FSE tracing configuration file. Used for technical support.

Log files

- `var/opt/fse/log/fse.log` (Linux)
`%InstallPath%\var\log\fse.log` (Windows)
(S + C)
FSE event log file. There is a single FSE event log in an FSE installation (client or server).
- `/var/opt/fse/log/error.log` (Linux)
`%InstallPath%\var\log\error.log` (Windows)
(S + C)
FSE error log file. There is a single FSE error log in an FSE installation.
- `/var/opt/fse/log/backup.log` (Linux)
`%InstallPath%\var\log\backup.log` (Windows)
(S)
FSE backup log file.

- `/var/opt/fse/log/hsmfs_*.log` (Linux)
`%InstallPath%\var\log\hsmfs_*.log` (Windows)
(C)
HSM file system log file. There is a single HSM file system log file for each configured HSM file system (partition).
- `/var/opt/fse/log/debug/*.log` (Linux)
`%InstallPath%\var\log\debug/*.log` (Windows)
(S + C)
FSE debug files (one per FSE process). Used by technical support.

FSE databases and system files

- `/var/opt/fse/cfg/*` (Linux)
`%InstallPath%\var\cfg*` (Windows)
(S)
Configuration Database (CfgDB)
- `/var/opt/fse/rmdb/rmdb.gdb` (Linux)
`%InstallPath%\var\rmdb\rmdb.gdb` (Windows)
(S)
Resource Management Database (RMDB)
- `/var/opt/fse/part/*/hsm/*` (Linux)
`%InstallPath%\var\part*\hsm*` (Windows)
(C)
Hierarchical Storage Management Database (HSMDB) files (one set per FSE partition).
- `/var/opt/fse/part/*/fsc/*` (Linux)
`%InstallPath%\var\part*\fsc*` (Windows)
(S)
File System Catalog (FSC) files (one set per FSE partition).

Note: FSE databases and system files must not be directly altered in any way!

See also:

[fse](#), [fsebackup](#), [fsebak](#), [fsecheck](#), [fsedrive](#), [fsefile](#), [fsejob](#), [fselibrary](#), [fsemedium](#), [fsepartition](#), [fsepool](#), [fserecover](#), [fserestore](#), [fsesystem](#)

fse

Initiate or terminate the FSE server or FSE client operation. Show current status of the FSE daemons. Prepare FSE system for the restore process. Configure mount points for HSM file systems (on Windows systems only). Enable or disable Limited Access Mode for a particular HSM file system (on Windows systems only).

Syntax

```
fse --start
fse --stop
fse --restart
fse --status

fse --startrestore

fse --mount MountPoint VolumeName
fse --umount MountPoint VolumeName

fse --dismount-ntfs VolumeName
fse --limit-access VolumeName
fse --full-access VolumeName
```

Description

The `fse` command (startup script on Linux systems, executable on Windows systems) affects core FSE components. It should be used with maximum care as its use can lead to improper FSE server or client operation or no operation at all. The FSE administrator must know exactly what he is doing when using this command.

The command is used to perform the following FSE administrator's tasks:

- start up or shut down FSE processes to enable or cease the FSE (server, client) operation,
- restart FSE (server, client) processes in case of trouble,
- print status information for the configuration-independent FSE processes (FSE daemons) running on the FSE installation (FSE server, FSE client),
- prepare the FSE installation for the restore process,
- on Windows systems, configure mount points for HSM file systems, mount or unmount HSM file systems.

- on Windows systems, dismount a newly formatted volume, enable or disable Limited Access Mode for a particular HSM file system.

FSE processes

Core FSE components, like the `fse` command, are running FSE processes that are responsible for proper FSE operation. There are two groups of FSE processes: configuration-independent and configuration-dependent. The *configuration-independent* FSE processes, also called FSE daemons, allow FSE software to operate and intercommunicate. The *configuration-dependent* FSE processes control a single configured FSE resource. The FSE administrator can configure more than one resource of the same kind, so there can be several FSE processes of the same kind running simultaneously. Currently, the only resources that are controlled by their own processes are FSE partition and FSE library.

Consolidated and distributed FSE implementations

It is possible either to set up an FSE installation on a stand-alone machine or to distribute its functionality across one FSE server and an arbitrary number of FSE clients. In the first case, all FSE processes are running on the same system; in the second set-up some processes run only on the FSE server as FSE server processes (marked with an `s` character), some run only on the FSE clients as FSE client processes (marked with a `c` character) and some on both/all (marked with `s + c` characters).

Configuration-independent FSE processes—FSE daemons are:

- *Service (S + C)*
- *Resource Manager (S)*
- *Management Interface (S)*
- *File System Event Manager (C)*

There is one File System Event Manager running on each FSE client host. It controls an arbitrary number of HSM file systems that reside on the same FSE client host.

Configuration-dependent FSE processes:

- *Partition Manager (S)*
- *Back End Agent (S)*
- *Library Agent (S)*

`fse status` prints info on the configuration-independent FSE processes only. For each FSE daemon, `fse status` prints its binary name, its current status (running, stopped) and the associated process IDs. If the command is executed on the FSE server (FSE client) it only prints info on the FSE server (FSE client) daemons. The daemon binary names are listed in the following table.

Process	FSE Daemon
omniNames	CORBA Naming Service
fse-svc	Service
fse-rm	Resource Manager
fse-mif	Management Interface
fse-fsevtmgr	File System Event Manager

Post-start and pre-stop scripts

The FSE administrator can set up two helper scripts which perform certain automated tasks at the FSE installation's (FSE server's, FSE client's) start-up and shut-down time. They are executed by the `fse` command.

Both helper scripts have to be placed in the `/opt/fse/sbin` directory (on Linux) or `%InstallPath%\bin` directory (on Windows) on a local machine, and have to be named as specified below:

- `post_start.sh` (Linux); `post_start.cmd` (Windows)

This is the post-start helper script. It is executed after all FSE processes have been started up and all existing HSM file systems with an entry in the `/etc/fstab` file (on Linux) or all existing HSM file systems with a defined mount point (on Windows) have been mounted, that is, immediately after the FSE installation is put into its fully operational state.

- `pre_stop.sh` (Linux); `pre_stop.cmd` (Windows)

This is the pre-stop helper script. It is executed before all mounted FSE file systems are unmounted and all running FSE processes are shut down, that is, immediately before the FSE installation is pulled out of its fully operational state.

If either of these scripts is missing, it is simply ignored.

Note: The commands that you specify in the post-start and pre-stop scripts must not block the execution of the `fse` command; therefore, they must conform to the following rules:

- They must not require interactive input.
 - They must finish in a reasonable amount of time and then return control to the script.
-

Options

--start

This option has different effects depending on whether the `fse` command is executed on a consolidated FSE implementation or on a distributed FSE implementation on an FSE server or client.

Before using this option, ensure that `omniNames` (the `omniORB` naming service daemon) is running on the consolidated FSE installation or on the FSE server.

If used on a consolidated implementation, the `start` option:

1. initiates `omniNames` — the `omniORB` naming service daemon, if it is not running,
2. initiates the FSE system operation starting up all FSE processes (see “[FSE processes](#)” on page 258 for details),
3. (on Windows) mounts all existing HSM file systems,
(on Linux) mounts all HSM file systems with an entry in the `/etc/fstab` file,
4. executes the `post-start` script, if it exists (see “[Post-start and pre-stop scripts](#)” on page 259 for details on the post-start script).

If used on the FSE server in a distributed implementation, it:

1. initiates the FSE server operation starting up all FSE server processes (see “[FSE processes](#)” on page 258 for details),
2. executes the `post-start` script, if it exists.

If used on an FSE client in a distributed implementation, it:

1. initiates `omniNames` — the `omniORB` naming service daemon, if it is not running,
2. initiates the FSE client operation starting up all FSE client processes (see “[FSE processes](#)” on page 258 for details),
3. (on Windows) mounts all existing HSM file systems,
(on Linux) mounts all HSM file systems with an entry in the `/etc/fstab` file,

4. executes the `post-start` script, if it exists.

--stop

As with the `start` option, the effects of this option depend on where the `fse` command is executed.

If used on a consolidated implementation, the `stop` option:

1. executes the `pre-stop` script, if it exists (see “[Post-start and pre-stop scripts](#)” on page 259 for details on the pre-stop script),
2. unmounts all mounted HSM file systems,
3. terminates the FSE system operation shutting down all running FSE processes (see “[FSE processes](#)” on page 258 for details).

If used on the FSE server in a distributed implementation, it:

1. executes the `pre-stop` script, if it exists,
2. terminates the FSE server operation shutting down all running FSE server processes (see “[FSE processes](#)” on page 258 for details).

If used on an FSE client in a distributed implementation, it:

1. executes the `pre-stop` script, if it exists,
2. unmounts all mounted HSM file systems,
3. terminates the FSE client operation shutting down all running FSE client processes (see “[FSE processes](#)” on page 258 for details).

--restart

This option has the same effect as a consecutive invocation of the `fse stop` and `fse start` commands.

Note: In a distributed FSE implementation, restarting the FSE processes on the FSE server requires restarting processes on a particular FSE client to allow normal operation of each FSE partition configured on that client.

--status

Show the status of each FSE (server, client) daemon, such as configuration-independent FSE (server, client) processes. See “[FSE processes](#)” on page 258 for FSE daemon status info details.

--startrestore

Prepare the FSE installation for the restore process. This option puts FSE in a special “restore” mode in order to disable the HSM file systems; the way it achieves varies according to the operating system:

On Linux, the `fse --startrestore` command starts all local FSE processes except the HSM File System Event Manager. Also the HSM file systems are not mounted in this mode.

On Windows, the `fse --startrestore` command starts all local FSE processes, but mounts the HSM file systems as ordinary NTFS file systems.

Note that you should invoke the `fse --startrestore` command only when the local FSE processes have been stopped. You must invoke it on each component of the FSE installation that you want to restore *before* running the `fserestore` command.

mount, --mount / umount, --umount

On Linux, the command `fse --mount|--umount` should be not used by the FSE administrator directly. It is called by the `fse-svc` daemon during recovery of the HSM file system.

On Windows, use the command `fse --mount|--umount` with great care. It should be used instead of the `mountvol` command if you want to manage an FSE partition (for example, to enable or disable it).

- `fse --mount` will create a mount point *MountPoint* for the volume *VolumeName* and trigger the HSM FS filter to connect to the volume device file and start FSE-HSM.
- `fse --umount` will stop FSE-HSM and switch the HSM FS filter operation for managing accesses to the volume *VolumeName* from Full Access Mode to Limited Access Mode. See the description of the `--limit-access` option for information on Limited Access Mode.

MountPoint

A directory path that represents the mount point to which the *VolumeName* NTFS volume will be mounted.

VolumeName

An NTFS volume that will store an HSM file system on a Windows client.

--dismount-ntfs

Windows systems only: This option dismounts the newly formatted NTFS volume *VolumeName*. It sends an appropriate code (the `FSCTL_DISMOUNT_VOLUME` ioctl code) to the volume. Later, when the volume is mounted again, the HSM FS filter can attach to it before NTFS, and thus gain control over it. Note that the `fse --dismount-ntfs` command will fail if the volume is in use, that is, when some processes have open handles on it.

--limit-access

Windows systems only: This option switches the HSM FS filter operation for managing access to the volume *VolumeName* from Full Access Mode to Limited Access Mode. In this mode, modifications on the corresponding HSM file system are not allowed. The volume remains mounted at its mount point, FSE-HSM continues to run for the HSM file system and online files are still migrated and released. Recalls of offline files are denied. When an offline file is accessed while the HSM file system operates in Limited Access Mode, the application that is accessing the file receives an error and the file is not recalled. For more information on Limited Access Mode, see “[HSM File System Access Modes](#)” on page 146.

--full-access

Windows systems only: This option switches the HSM FS filter operation for managing accesses to the volume *VolumeName* from Limited Access Mode to Full Access Mode. This change is possible only if the corresponding FSE-HSM process is running and the HSM file system on the volume operates in Limited Access Mode. For more information on Full Access Mode, see “[HSM File System Access Modes](#)” on page 146.

Examples

- Create a mount point for the volume and trigger the FSE FS filter to connect to a volume device file and start FSE-HSM (on Windows):

```
fse --mount
  I:\fs02\Volume{86eec883-a640-11d7-9be3-00096bb08bc0}
```

- Stop FSE-HSM and enable Limited Access Mode for the specified volume (on Windows systems):

```
fse --umount
  c:\emptydir Volume{d9568629-da27-11d7-bf21-0050da45607}
```

- Disconnect NTFS from the specified volume (on Windows systems):
fse --dismount-ntfs
Volume{d9568629-da27-11d7-bf21-0050da45607}
- Enable Limited Access Mode for the specified volume (on Windows systems):
fse --limit-access
Volume{d9568629-da27-11d7-bf21-0050da45607}
- Disable Limited Access Mode (switch back to Full Access Mode) for the specified volume (on Windows systems):
fse --full-access
Volume{d9568629-da27-11d7-bf21-0050da45607}

Files

On Linux:

- /opt/fse/sbin/post_start.sh
The post-start helper script.
- /opt/fse/sbin/pre_stop.sh
The pre-stop helper script.

On Windows:

- %InstallPath%\bin\post_start.cmd
The post-start helper script.
- %InstallPath%\bin\pre_stop.cmd
The pre-stop helper script.

See also:

[fssystem](#), [fsepartition](#), [fserestore](#), [Introduction to the FSE system \(fseintro\)](#)

fsebackup

Back up the FSE system state to backup media.

Syntax

```
fsebackup {-m | --media } Barcode... [{-c | --client }
[Client... ]] [--skip-hsmfs ] [{-l | --log-file } LogFileName ]
fsebackup {-h | --help }
```

Description

A backup of the FSE installation is performed in such a way that it preserves a consistent state of information stored in FSE databases, FSE system files, and user data on HSM file systems. Its operation is designed to have minimal impact on FSE system operations.

There are three levels of backup:

- **Server backup**
Backs up all the required databases stored on the FSE server, but does not back up any data on the FSE clients (HSM file system and HSMDB). The HSM file system can be recovered from the File System Catalog (FSC) with the `fserrecover` command. This is the fastest and preferred way of doing a backup.
- **Full backup**
Backs up all the required databases stored on the FSE server and all FSE clients. It is possible to recover the whole FSE installation (server & clients) by doing a restore.
- **Single client backup**
Backs up all the required databases stored on the FSE server and a specific FSE client. It is possible to recover the FSE server and specific client by doing a restore, while other clients would require running the `fserrecover` command to recover the HSM file system.

Prerequisites

FSE backup uses logical volume snapshot technology to keep the FSE system's operation almost unobstructed during the backup window. To allow the FSE backup to use this technology, HSM file systems, FSE configurations, and all FSE

databases and system files must reside on LVM logical volumes (on Linux), or special snapshot software must be installed (on Windows). This is an essential prerequisite.

FSE backup uses FSE backup media — FSE media that are assigned to a special media pool, distinguished by its BACKUP type. A backup pool should be configured with only one volume in order to use all the space on medium. Backup media in this pool should be formatted, but not initialized. If the backup process encounters an initialized FSE medium, it will return an error. This is applied to prevent accidental loss of data. For details on configuring an FSE backup media pool, see the `fsepool` man page.



Caution: It is the FSE administrator's responsibility to dedicate correct backup media for each backup session. `fsebackup` does not keep a record of which backup media has already been used for backup and when. It will only check if media with the specified barcodes are assigned to an FSE backup media pool before proceeding. It is also suggested to keep at least two backup sets in order to ensure that you always have at least one valid backup.

Backup procedure

After the prerequisites are checked and fulfilled, `fsebackup` first suspends the FSE installation and achieves a consistent state of all File System Catalogs (FSCs) and Hierarchical Storage Management Databases (HSMDBs). It then creates snapshots of all required disk volumes (`/var/opt/fse` and `/var/opt/fse/part` on Linux and the appropriate disk on Windows) before resuming the FSE system operation. Suspend and resume operations are performed using the `fsebak` command. Once the snapshots are successfully created the backup can continue.

All relevant data is gathered together and archived using the `tar` tape archiving utility on Linux and a proprietary FSE tool on Windows (`fse-arch`). The data are then written to the FSE backup media in the form of several archive files separated with file marks. The first archives on the medium are client specific data (HSM file system and HSMDB) in case client data is being backed up. This is followed by File System Catalog archives for each FSE partition, then the Configuration Database (configuration information for the FSE installation, FSE libraries, FSE drives, FSE media pools and FSE partitions), the Resource Management Database, Fast Recovery Information (FRI) in temporary disk files and global configuration (backup, trace and CORBA/omniORB configuration files). Offline files on the HSM file system are backed up as sparse files. These files practically do not occupy any space on the backup medium.

FSE backup usually runs in the background, started automatically by cron or similar utilities. A special lock file `backup.pid` is used to prevent starting multiple backup processes in parallel.

Activity logging

The FSE administrator usually consults the `fse.log` file to determine backup start and stop times. These times are used to estimate the required backup window. Additionally, `fsebackup` records its progress report to the `backup.log` file (see “Files” on page 269). When a backup successfully finishes, it reports the barcodes of the FSE backup media that the backup data was written to.

Configuration

On Linux, the administrator needs to configure the parameters in the `backup.cfg` configuration file. The configurable parameters and their default values are as follows:

- `HSM_LV_VAR = vg_fse/fsevar`
Name of the LVM volume group and the LVM logical volume used to store the file system for the `/var/opt/fse` mount point.
- `HSM_LV_PART = vg_fse/fsepart`
Name of the LVM volume group and the LVM logical volume used to store the file system for the `/var/opt/fse/part` mount point.
- `SNAPSHOT_PCT = 10`
Size of the LVM logical volume snapshot, expressed in percentage of the original LVM logical volume size.

If the `/etc/opt/fse/backup.cfg` file is not found, default values are used for the FSE backup process.

Options

- `-m, --media`
Perform a backup of the FSE server. Record the backup image data on one or more FSE media with barcodes `Barcode...`

- **Barcode...**
A sequence of barcodes of FSE media used to store backup image data. The media barcodes must be separated by spaces.
- **-c, --client**
Perform a backup of FSE clients, or if a specific client hostname is specified by the *Client...* argument, only a specific client. This option can only be used together with the `--media` option.
- **Client...**
A sequence of fully-qualified domain names (FQDN) of the FSE clients that are backed up. FQDNs must be separated by spaces.
- **--skip-hsmfs**
Do not perform a backup of the HSM file system. This option can only be used together with the `--media` option.
- **-l, --log-file**
Log FSE backup activity into the file *LogFile*. FSE backup activity is normally logged into the default FSE backup log file. This option can only be used together with the `--media` option.
- **LogFile**
File name (with optional path) of the additional backup activity log that will record the progress of the FSE backup process.
- **-h, --help**
Print the usage information for this FSE command.
- **-v, --version**
Print the copyright and version information for this FSE command.

Examples

- Back up the current FSE server state to FSE media with barcodes 000123, 000124, and 000125:

```
fsebackup --media 000123 000124 000125
```
- Back up the current FSE server state and all FSE clients to FSE media with barcodes 000123, 000124, and 000125:

```
fsebackup --media 000123 000124 000125 --client
```
- Back up the current FSE server state to an FSE medium with barcode 000202. Write an additional backup log to file `/tmp/backup_000202.log`:

```
fsebackup --media 000202 -l /tmp/backup_000202.log
```

- Back up the current state of an FSE server and FSE clients `fseclient_98.hpfsse.de`, `fseclient_99.hpfsse.de` to the FSE medium with barcode 000300:

```
fsebackup -m 000300 -c fseclient_98.hpfsse.de  
fseclient_99.hpfsse.de
```

Files

- `/opt/fse/newconfig/backup.cfg` (Linux specific)
Template for an FSE backup configuration file.
- `/etc/opt/fse/backup.cfg` (Linux specific)
FSE backup configuration file.
- `/var/opt/fse/log/backup.log` (Linux)
`%InstallPath%\var\log\backup.log` (Windows)
Log file with recorded information about the activity of the FSE backup process.
- `/var/opt/fse/log/fse.log` (Linux)
`%InstallPath%\var\log\fse.log` (Windows)
FSE event log file. It is used for measuring the required backup window.
- `/var/opt/fse/lock/backup.pid` (Linux)
`%InstallPath%\var\lock\backup.pid` (Windows)
FSE backup lock file.

See also:

[fserestore](#), [fsebak](#), [fserecover](#), [Introduction to the FSE system \(fseintro\)](#)

fsebak

Achieve a consistent state of all HSMDBs and FSCs to allow an eventual backup creation.

Syntax

```
fsebak --suspend
fsebak --continue StatusCode

fsebak --reserve-medium Barcode
fsebak --change-medium Barcode JobIOR

fsebak --complete-job JobIOR

fsebak {-h | --help }
fsebak {-v | --version }
```

Description

Common usage

The `fsebak` command affects the core FSE components. Its primary use is from inside the `fsebackup` command and not as a stand-alone tool, except at the request of technical support personnel. Therefore, it should be used with great care, as its use can lead to improper FSE system operation or no operation at all.

Internal command operation

`fsebak` prepares Hierarchical Storage Management Databases (HSMDB) and File System Catalogs (FSC) of all configured FSE partitions for an eventual backup procedure. To achieve such database consistency, the operation of all currently running Partition Managers must be temporarily suspended. Duplicate copies of HSMDBs and FSCs can then be created using other system commands. After a successful backup creation Partition Managers are unblocked using `fsebak` once again. This action restores normal FSE system operation.

Each FSE command communicates with the FSE system components through the Management Interface (`fse-mif`). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the `fse status` command. If the Management Interface is not running, start it with the `fse start` command. Additionally, the Partition Manager processes (`fse-pm`) for all configured FSE partitions must be running. This can be checked with the `fsesystem` command.

After successful command execution, `fsebak` notifies the FSE administrator with a suitable message, which is printed to the standard output stream (stdout). If there was an error during the command's execution, an appropriate error message is printed to the standard error stream (stderr).

Achieving FSE database consistency

For each running Partition Manager, `fsebak --suspend` performs the following procedure:

- block the incoming job requests in Partition Manager;
- make the corresponding HSMDB and FSC incorporate all changes from their transaction logs into the main database files;
- write temporary files `hsm.ref` and `fsc.ref` with references to HSMDB and FSC for use with the `--continue` option;
- report action success or failure.

Note: The contents of the `hsm.ref` and `fsc.ref` files must not be directly altered.

If all these steps were successfully completed, duplicate copies of HSMDB and FSC can be made. Usually they are created with the `fsebackup` command. See the `fsebackup` man page for details.

Resuming a normal FSE system operation

For each running Partition Manager, `fsebak --continue StatusCode` performs the following procedure:

- read references to HSMDB and FSC from files `hsm.ref` and `fsc.ref` and delete these two files;
- start accepting incoming job requests in Partition Manager and resume normal operation.

If the FSE administrator ensured that the HSMDB and FSC backups were stored on a safe medium, he/she is advised to use value 0 (zero) in place of the `StatusCode` argument. This value means that all old transaction log files of HSMDBs and FSCs will be removed, as they are no longer required. If any other value is used as the `StatusCode`, these transaction logs will be preserved and optionally used again, if needed.

Options

- `--suspend`

Prepare each HSMDB and FSC for backing up. This action temporarily suspends normal FSE system operation in all running Partition Managers. See “[Description](#)” on page 270 for details of this operation.

■ `--continue`

Unblock the FSE system operation using the value `StatusCode`. This enables the incoming job requests in all running Partition Managers and restores normal FSE system operation.

■ `StatusCode`

The status code used with the option `--continue`. See “[Description](#)” on page 270 for an explanation of the meaning and proper use of this value.

■ `--reserve-medium`

Create a new administrative job for a backup purpose, print the interoperable object reference of the job and the name of the device where the backup archive will be written. The administrative job is a dummy job: it only allocates an FSE medium with the specified barcode `Barcode` from an FSE backup media pool and loads it into a free FSE drive. It then prints the device name of this FSE drive.

■ `Barcode`

The barcode of the FSE medium that will be used to store the backup archive. This medium must be assigned to an FSE backup media pool.

■ `--change-medium`

Allocate a new FSE medium to store the backup archive. Use of this option causes the job with the interoperable object reference `JobIDOr` to unload the currently allocated FSE medium from the FSE drive, release it, allocate the FSE medium with the specified barcode `Barcode` from an FSE backup media pool and load it into the same FSE drive. Use of this option is necessary if the previously reserved medium does not have enough free space to store the whole backup archive.

- `JobIOR`
The interoperable job reference, which is used as an “address” of the job in the CORBA Naming Service daemon. It is represented as a 160-byte text string, beginning with the prefix `IOR: .`
- `--complete-job`
Complete the administrative job with the interoperable object reference `JobIOR` and release the currently allocated FSE medium.
- `-h, --help`
Print the usage information for this FSE command.
- `-v, --version`
Print the copyright and version information for this FSE command.

Examples

- Suspend normal FSE system operation and prepare HSMDBs and FSCs for backing up:
fsebak --suspend
- Resume normal FSE system operation after backups of the FSE databases and system files were successfully created:
fsebak --continue 0
- Resume normal FSE operation after backups of all FSE databases and system files were created, but preserve the HSMDB and FSC transaction log files for optional future use:
fsebak --continue 1
- Create an administrative job. Allocate the FSE medium with the barcode `backup-003` for a backup procedure and load it into an FSE drive:
fsebak --reserve-medium backup-003
- Release the currently allocated FSE medium for the administrative job with the interoperable object reference `IOR:010000000e00...` (160-byte text string), unload it, allocate the new FSE medium with the barcode `backup-105` and load it into the same FSE drive:
fsebak --change-medium backup-105 IOR:010000000e00...
- Complete the administrative job with the interoperable object reference `IOR:010000000a00...` (160-byte text string) and release the allocated FSE medium:
fsebak --complete-job IOR:010000000a00...

See also:

[fse](#), [fssystem](#), [fsebackup](#), [fserestore](#), [Introduction to the FSE system \(fseintro\)](#)

fsecheck

Perform a File System Catalog consistency check against the HSM file system or FSE media.

Syntax

```
fsecheck  {-i | --fsc-hsmfs } PartitionName
fsecheck  {-m | --fsc-media } [--autocorrect ] PartitionName

fsecheck  {-h | --help }
fsecheck  {-v | --version }
```

Description

Each FSE command communicates with the FSE system components through the Management Interface (`fse-mif`). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the `fse status` command. If the Management Interface is not running, start it with the `fse start` command.

After a successful command execution `fsecheck` notifies the FSE administrator with a suitable message, which is printed to the standard output stream (stdout). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (stderr).

Checking the consistency of the File System Catalog

The `fsecheck` command creates a maintenance job on the specified FSE partition. The maintenance job is executed by the corresponding Partition Manager and performs a File System Catalog consistency check. This is a process of comparing file entries, which are present in the File System Catalog, either with files on the corresponding HSM file system or with Fast Recovery Information (FRI) on the corresponding FSE media. It detects the situations where:

- some files on the HSM file system or on the FSE media are missing from the File System Catalog,
- it is uncertain for recently created files on the HSM file system whether they are missing from File System Catalog or are too new to have their file ID recorded in File System Catalog yet.

`fsecheck` reports the consistency check findings to the shell output. These findings are also recorded in the FSE check log (see “Files” on page 277).

Note: The maintenance job for the consistency check of the File System Catalog against the HSM file system keeps track of its progress. If it is aborted with the `fsejob` command and later re-run with `fsecheck`, it resumes its operation from the interruption point onward.

The `fsecheck` command also offers automatic correction, which can be enabled as an addition to the comparison of the File System Catalog against FSE media. The correction procedure updates the File System Catalog with missing information about file splits found in Fast Recovery Information on FSE media.

Options

- `-i, --fsc-fsefs`
Perform a consistency check on the FSE partition *PartitionName* comparing the corresponding File System Catalog against the associated HSM file system. See “[Checking the consistency of the File System Catalog](#)” on page 275 for details on this process.
- `PartitionName`
The name of the FSE partition on which to perform the consistency check.
- `-m, --fsc-media`
Perform a consistency check on the FSE partition *PartitionName* comparing the corresponding File System Catalog against the associated FSE media. See “[Checking the consistency of the File System Catalog](#)” on page 275 for details on this process.
- `--autocorrect`
Automatically correct errors discovered by the consistency check of the File System Catalog against corresponding FSE media. See “[Checking the consistency of the File System Catalog](#)” on page 275 for details on this process. This option can only be used together with the `--fsc-media` option.
- `-h, --help`
Print the usage information for this FSE command.
- `-v, --version`
Print the copyright and version information for this FSE command.

Examples

- Trigger a File System Catalog consistency check against the HSM file system of the FSE partition `daily_reports`:

```
fsecheck --fsc-hsmfs daily_reports
```

- Trigger a File System Catalog consistency check against FSE media assigned to the FSE partition `archive_part_32`:

```
fsecheck --fsc-media archive_part_32
```

- Trigger a File System Catalog consistency check against FSE media assigned to the FSE partition `archive_part_45`. Enable autocorrection:

```
fsecheck --fsc-media --autocorrect archive_part_45
```

Files

- `/var/opt/fse/log/check_hsmfs_fsc_<PartitionName>.log` (Linux)
`%InstallPath%\var\log\check_hsmfs_fsc_<PartitionName>` (Windows)

Log file with the recorded findings of the consistency check of the File System Catalog against the HSM file system.

- `/var/opt/fse/log/check_media_fsc_<PartitionName>.log` (Linux)
`%InstallPath%\var\log\check_media_fsc_<PartitionName>` (Windows)

Log file with the recorded findings of the consistency check of the File System Catalog against the FSE media.

See also:

[fserecover](#), [Introduction to the FSE system \(fseintro\)](#)

fsedrive

Manage the configuration of the FSE drives.

Syntax

```
fsedrive {-a | --add } ConfigFileName
fsedrive {-r | --remove } Name [-F | --force ]

fsedrive {-s | --show } Name [RevisionNumber | {-H |
--history } ]
fsedrive {-m | --modify } Name ConfigFileName
fsedrive {-l | --list } [-D | --detail ]

fsedrive --disable Name [-F | --force ]
fsedrive --enable Name

fsedrive {-h | --help }
fsedrive {-v | --version }
```

Description

This FSE command manages the configuration of the FSE drives. It is used to perform the following FSE administrator's tasks:

- configure a new or remove an existing FSE drive,
- retrieve the configuration of an existing FSE drive,
- reconfigure an existing FSE drive,
- list configured drives in the FSE installation,
- disable or enable operation of an FSE drive.

Each FSE command communicates with FSE system components through the Management Interface (*fse-mif*). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the *fse status* command. If the Management Interface is not running, start it with the *fse start* command.

After a successful command execution, *fsedrive* notifies the FSE administrator with a suitable message, which is printed to the standard output stream (stdout). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (stderr).

Options

- `-a, --add`

Configure a new FSE drive with parameters, written in the configuration file *ConfigFileName*. This action checks the validity of the configuration file, copies the file to Configuration Database and inserts the parameter values into Resource Management Database. FSE drive names are unique within the FSE installation.
- *ConfigFileName*

The name of the FSE drive configuration file to use.
- `-r, --remove`

Remove an existing FSE drive from the FSE installation. This includes removal of all configuration file revisions of this FSE drive from the Configuration Database. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is "No".
- *Name*

The name of the FSE drive being configured.
- `-F, --force`

Set forced mode for the FSE drive remove or disable operation. In this mode, the requirement for confirmation of the FSE drive removal or disabling is suppressed. This option can only be used together with the `--remove` or `--disable` option.
- `-s, --show`

If the *RevisionNumber* argument is specified, print the appropriate configuration revision for the FSE drive *Name*, otherwise print the current configuration revision.
- *RevisionNumber*

A non-negative integer value which represents the configuration revision to print.

- `-H, --history`

Print the configuration history depth for the FSE drive *Name*. Configuration history depth is the number of configuration file revisions stored in the Configuration Database for this FSE drive. This option can only be used together with the `--show` option.
- `-m, --modify`

Modify the configuration of FSE drive *Name* with the configuration file *ConfigFileName*. This action checks the validity of the new configuration file, copies it to Configuration Database as a new configuration revision and triggers the FSE drive reconfiguration.
- `-l, --list`

Display status information for configured FSE drives within the FSE installation. The reported information for an FSE drive includes:

 - Drive ... drive name
 - Family ... drive family (AIT, LTO, AIT-WORM)
 - Library ... name of the corresponding FSE library
 - Host ... fully-qualified domain name (FQDN) of the host the drive is attached to
 - CtrlDev ... drive control device file
 - Status ... current drive status (online, in use, offline)
 - Medium ... barcode of the medium currently in the drive (optional)
- `-D, --detail`

Retrieve additional FSE drive information and display the retrieved data in *ParameterName: ParameterValue* format with each parameter in its own line. The following additional information is reported:

 - VendorID ... ID string of the drive manufacturer
 - ProductID ... ID string of the drive model
 - SerialNo ... serial number of the drive
 - FirmwareRev ... revision of the firmware the drive currently uses
 - ScsiID ... ID string of the corresponding SCSI device
 - Index ... index of the drive inside the FSE library
 - UsageCount ... usage counter for the drive
 - Capabilities ... code for the drive features

— Key ... internal RMDB key value assigned to the drive

This option can only be used in combination with the `--list` option.

- `--disable`
Disable operation of the FSE drive *Name*. Disabling the drive means putting it offline in case it needs cleaning or servicing. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is "No".
- `--enable`
Enable operation of FSE drive *Name*, which is currently disabled. Enabling the drive means making it available to the FSE installation again.
- `-h, --help`
Print the usage information for this FSE command.
- `-v, --version`
Print the copyright and version information for this FSE command.

Examples

- Configure a new FSE drive from the configuration file `standalone_drive.fse`:
`fse drive --add standalone_drive.fse`
- Remove the configured FSE drive named `lib_drive_01`:
`fse drive --remove lib_drive_01`
- Remove the configured FSE drive named `drive_AB`; skip confirmation:
`fse drive --remove drive_AB -F`
- Show the current configuration of the FSE drive named `system_drive`:
`fse drive --show system_drive`
- Show the configuration revision 4 of the FSE drive named `system_drive_backup`:
`fse drive --s system_drive_backup 4`
- Show the configuration history depth for the FSE drive named `standalone_drive`:
`fse drive --show standalone_drive --history`
- Copy the first (oldest) configuration revision of the FSE drive named `lib_drive_03` to a file `lib_drive_03_cfg_1.fse` in the current directory:

```
fse drive --show lib_drive_03 1 >  
lib_drive_03_cfg_1.fse
```

- Apply a new configuration for the FSE drive named `standalone` from the configuration file `standalone-new.fse`:

```
fse drive --modify standalone standalone-new.fse
```

- List names and states of the currently configured FSE drives:

```
fse drive --list
```

- Retrieve additional information about the currently configured FSE drives:

```
fse drive --list --detail
```

- Disable the FSE drive named `drive_5`:

```
fse drive --disable drive_5
```

- Enable the currently disabled FSE drive named `extra_drive`:

```
fse drive --enable extra_drive
```

File

- `/opt/fse/newconfig/fse-drive.cfg` (Linux)
`%InstallPath%\newconfig\fse-drive.cfg` (Windows)

Template for an FSE drive configuration file.

See also:

[fse](#), [fselibrary](#), [fsemedium](#), [fsepool](#), [Introduction to the FSE system \(fseintro\)](#)

fsefile

Manage the files on an HSM file system. Recall older file generations from FSE media. Add files to the migration candidate list. Release files from an HSM file system. Trigger migration on an FSE partition. Trigger deletion of files from an HSM file system.

Syntax

```
fsefile {-l | --list } FileName... [-D | --detail]
fsefile {-H | --history } {FileName... [-D | --detail]
  | --id PartitionName FileID... [-D | --detail]}

fsefile {-R | --recall } {FileName... [--date Date |
--migid MigID } --into Path ] | --id PartitionName FileID...
[--date Date | --migid MigID ] --into Path }

fsefile {-M | --migrate } FileName...
fsefile {-S | --release } FileName...

fsefile {-T | --trigger-migration } PartitionName
fsefile {-D | --trigger-deletion } PartitionName [-F | --force]
  [--no-monitor]

fsefile {-h | --help }
fsefile {-v | --version }
```

Description

This FSE command manages files on an HSM file system that belongs to a particular FSE partition. It is used to perform the following FSE administrator's tasks:

- show the file ID and current status of a file on an HSM file system,
- show the history with information about older migrated generations of a file on an HSM file system,
- explicitly recall a file from FSE media to the HSM file system,
- recall an older migrated generation of a file from FSE media to an arbitrary location on a non-HSM file system,
- explicitly put a file on the migration candidate list for the corresponding FSE partition,
- unconditionally release an already migrated file on an HSM file system,

- trigger the migration of files that are currently considered migration candidates on a particular FSE partition and trigger the migration of files which were not migrated because of some failed migration,
- trigger deletion of files from an HSM file system that are covered by the configured deletion policy in the relevant FSE partition configuration file.

Each FSE command communicates with the FSE system components through the Management Interface (`fse-mif`). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the `fse status` command. If the Management Interface is not running, start it with the `fse start` command.

After a successful command execution `fsefile` notifies the FSE administrator with a suitable message, which is printed to the standard output stream (`stdout`). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (`stderr`).

Job progress monitoring

`fsefile` can receive messages from the Management Interface to give you the ability to monitor the progress of certain actions triggered by the command.

Monitoring is enabled by default and can be disabled using the `--no-monitor` option.

When monitoring is enabled, the `fsefile` command reports events and other relevant information from the FSE processes as long as the job that `fsefile` has started is running. After the job terminates, the command ends. If monitoring is disabled, the command returns the control immediately after it has started a corresponding job. In this case no information from the job is received, even in case of errors.

Monitoring can be canceled by pressing `^C` in the shell where `fsefile` runs.

Note: On canceling monitoring, the job will continue to run and will write progress to the `fse.log` file.

Options

- `-l, --list`
Show the file ID, current status (dirty, online, offline) and file name of each file specified by `FileName...`
- `FileName...`

A sequence of file names for files on which the operation is to be executed. File names must be separated by spaces. All files must be located on the same HSM file system (except if used with option `--list`).

If you specify file names without paths, the current directory must be the one in which the specified file is located.

Note: If a specified file is located on an HSM file system other than the HSM file system of the first specified file, a suitable warning is printed and the file is not processed.

- `-D, --detail`

Display the retrieved data in *ParameterName: ParameterValue* format with each parameter in its own line. This option can only be used in combination with the `--list` or `--history` option.
- `-H, --history`

For each migrated generation, show the file ID, migration ID, migration date and time, total size of streams that were migrated, and file name. If the file has been renamed between migrations, this command displays the last file name for all the generations. On Linux, files have only one stream and the migrated sizes are equal to the file size. On Windows, this is not always the case. This information is printed for each file specified by the *FileName...* argument or the `--id PartitionName FileID...` argument combination.
- `--id`

Use the *PartitionName FileID...* argument combination for file identification instead of the *FileName* argument. This option can only be used together with the `--history` or `--recall` option.
- *PartitionName*

Name of the FSE partition with the files that are being managed or on which the migration job is triggered.

■ *FileID...*

A sequence of file IDs for files the particular operation is executed on. File IDs must be separated by spaces. All files must be located on the same HSM file system. A file ID is a positive decimal integer number. The file ID of a particular file can be obtained with the `--list` command or by searching in the HSM file system log file for the corresponding file name. The HSM file system log file is named `hsmfs_*.log` and is located in the `/var/opt/fse/log` directory (on Linux systems) and in `%InstallPath%\var\log` (on Windows systems).

Note: If a specified file is located on an HSM file system other than the HSM file system of the first specified file, a warning is printed and the file is not processed.

■ `-R, --recall`

Recall a migrated generation of a file that is currently present on an HSM file system or that was present at some moment in the past. A generation can be recalled:

- in its place in the corresponding HSM file system (allowed only if the latest generation is specified);
- into an arbitrary directory on a non-HSM file system (allowed for any generation).

Recall is done separately for each file specified either by the *FileName...* argument or by the `--id PartitionName FileID...` argument combination; each file triggers its own recall job. Independently of the file specification method used, a particular older migrated generation can be specified using either the `--date Date` or the `--migid MidID` argument combination.

Note: The latest migrated generation can only be recalled with the *FileName...* argument, when the file still exists. This limitation results in the obligatory use of the `--into` option if the file to be recalled is specified with the `--id PartitionName FileID...` argument combination.



Caution: If a non-latest generation of a file is recalled using the `--into` option, the two security precautions are:

- owner and group ownership of the recalled file are both set to root;
 - all time stamps of the recalled file are set to the current date and time.
-

■ `--date`

Specify that the last generation of the file migrated before the point in time defined by the *Date* argument is to be recalled. This option can only be used together with the `--recall` option.

■ *Date*

Date and time of the point in time used to specify a particular migrated file generation. Date and time strings have to conform to the format of `YYYY/MM/DD hh:mm:ss` and must be enclosed together within double quotes.

■ `--migid`

Specify that the generation, which had been migrated in the migration job with migration ID *MigID*, is to be recalled. This option can only be used together with the `--recall` option.

■ *MigID*

Migration ID of a migration job in which a file generation had been migrated. The migration ID is a twelve-digit positive decimal integer number. The migration ID of a particular file generation can be obtained from the output of the `--history` option.

■ `--into`

Specify that the recalled file generation has to be saved into the directory with the absolute path *Path*. The path must point to a directory on a non-HSM file system. This option can only be used together with the `--recall` option.

Note: The `--into` option is required for recall of an older file generation.

■ *Path*

Absolute path pointing to a directory on a non-HSM file system, where the recalled file generation will be saved.

■ `-M, --migrate`

Put files specified by the *FileName...* argument on the corresponding migration candidate list. If a specified file is offline, it will be recalled first. This option is useful in situations where:

- a file is present on the HSM file system but has not been recognized by FSE for some reason. Adding it to the migration candidate list with this command preserves its attributes;
- a file is already migrated to an FSE medium whose current status is set to unreliable or it is obviously worn out. In these cases the file can be re-migrated to another FSE medium.

Note: If a specified file is already migrating or present on the release candidate list it cannot be added to the migration candidate list.

■ `-S, --release`

Unconditionally release files specified by the *FileName...* argument. All files must be located on the same HSM file system. The purpose of this operation is to gain free space on an HSM file system by releasing (large) files from it. The files released this way must be on the release candidate list.

■ `-T, --trigger-migration`

Trigger a migration job for the files that are currently present on the migration candidate list on FSE partition *PartitionName*.

Trigger a migration job for files that were not migrated because migration of those files failed.

■ `-D, --trigger-deletion`

Trigger deletion of files on an HSM file system that corresponds to the deletion policy defined for the owning FSE partition. The deletion policy consists of a sequence of *FileExpiration - Path* parameter pairs that are specified in the FSE partition configuration file.

Although the deletion removes the corresponding files from the HSM file system, their migrated generations still exist on FSE media, unless media reorganization with appropriate parameters has been run on the media that hold the migrated copies. If this is not the case, these migrated generations can still be recalled.

- `-F, --force`
Set forced mode for triggering the deletion of files on an HSM file system. In this mode, the FSE administrator's confirmation of triggering the file deletion is suppressed. This option can only be used together with the `--trigger-deletion` option.
- `--no-monitor`
Disable monitoring of the file deletion job which is to be performed on the HSM file system. Monitoring is enabled by default. See "[Job progress monitoring](#)" on page 284 for details on monitoring.
- `-h, --help`
Print the usage information for this FSE command.
- `-v, --version`
Print the copyright and version information for this FSE command.

Examples

- List current status of the file `updateMe.gz` in the current directory. Display each of the retrieved parameters in a separate line:

```
fsefile --list updateMe.gz --detail
```
- List file IDs of the file `updateMe.gz` in directory `/mnt/vg_fse/fse1` and file `archive.gz` in the current directory:

```
fsefile -l /mnt/vg_fse/fse1/updateMe.gz archive.gz
```
- Show history information for files with file IDs 21544 and 21560 on an FSE partition named `partition_02`:

```
fsefile -H --id partition_02 21544 21560
```
- Recall the latest migrated generation of the file `fsepkg_1.1.68.tar.gz` into the current directory on an HSM file system:

```
fsefile --recall fsepkg_1.1.68.tar.gz
```
- Recall the file `fsepkg_1.1.55.tar.gz` whose latest generation was migrated by the migration job with ID 266142980096 into the directory `/tmp/fse_recalled_files` on a non-HSM file system:

```
fsefile -R fsepkg_1.1.55.tar.gz --migid 266142980096 --into /tmp/fse_recalled_files
```
- Recall the last generation of the file `fsepkg_206_update.tar.gz` that was migrated before 2525/01/01 10:00:00 into directory `/tmp/fse_older_files` on a non-HSM file system:

```
fsefile --recall fsepkg_206_update.tar.gz --date "2525/01/01 10:00:00" --into /tmp/fse_older_files
```

- Recall the last generation of the file with file ID 30600 that was migrated before 2002/12/10 15:36:50 on an FSE partition named `FSEPartition010` into the directory `/tmp/fse_deleted_files` on a non-HSM file system:

```
fsefile --recall --id FSEPartition010 30600 --date "2002/12/10 15:36:50" --into /tmp/fse_deleted_files
```

- Recall an older generation of the file with file ID 18769 on an FSE partition named `partitionH` that was migrated in migration job with ID 266142980096 into the directory `/tmp/fse` on a non-HSM file system:

```
fsefile -R --id partitionH 18769 --migid 266142980096 --into /tmp/fse
```

- Put the files `big_file_A` and `big_file_B` on the corresponding migration candidate list:

```
fsefile --migrate big_file_A big_file_B
```

- Unconditionally release the files `week_50`, `week_51` and `/mnt/vg_fse/weekly-archives/week_52` (the current directory has to be set to `/mnt/vg_fse/weekly-archives` in order to release all files specified):

```
fsefile -S week_50 week_51 /mnt/vg_fse/weekly-archives/week_52
```

- Trigger a migration job for files on the migration candidate list on the FSE partition named `partition_03`:

```
fsefile --trigger-migration partition_03
```

- Trigger a file deletion job on the FSE partition named `partition_delete_me`. Skip the confirmation question and disable monitoring:

```
fsefile --trigger-deletion partition_delete_me -F --no-monitor
```

See also:

[fsepartition](#), [fsejob](#), [Introduction to the FSE system \(fseintro\)](#)

fsejob

Monitor execution of running FSE jobs. Show status of running FSE jobs.

Syntax

```
fsejob {-l | --list } [-M | --migration ] [-R | --recall ]
[-A | --admin ] [-C | --recovery ] [-T | --maint ]
[PartitionName ] [-D | --detail ]
fsejob {-t | --status } JobID

fsejob {-a | --abort } JobID...
fsejob {-p | --priority } JobID {Number | {-I | --incr }
Number | {-D | --decr } Number }
fsejob {-h | --help }
fsejob {-v | --version }
```

Description

This FSE command monitors execution of FSE jobs. It is used to perform the following FSE administrator's tasks:

- List running FSE jobs in the whole FSE installation or on a particular FSE partition,
- Show the current status of an FSE job,
- Abort execution of specified FSE jobs,
- Change the priority of an FSE job.

Each FSE command communicates with the FSE system components through the Management Interface (*fse-mif*). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the *fse status* command. If the Management Interface is not running, start it with the *fse start* command. There should also be a Partition Manager process (*fse-pm*) running for FSE partition for which the jobs are monitored. This can be checked with the *fsesystem* and *fsepartition* commands.

After successful command execution, *fsejob* notifies the FSE administrator with a suitable message, which is printed to the standard output stream (stdout). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (stderr).

Job list information

Job information that is displayed using the `--list` option is the following:

- JobID ... job identification number (JobID)
- Partition ... name of the FSE partition on which the job is running
- Type ... job type (migration, recall, admin, recovery, maintenance)
- Priority ... current job priority*
- Started ... job start date and time in the format YYYY/MM/DD
hh:mm:ss
- Status ... current job status (phase)

* A numerical value is shown for jobs which are in the phase of waiting for free resources and an “n/a” string for jobs that have already allocated the required resources.

The job ID is a numerical identification of a job that consists of the start date of the job and its sequence number for that particular day.

Job status information

Job status information printed using the `--status` option consists of two parts. The first part is the same information reported by the `--list` option, in this case laid out in rows rather than in columns. See “[Job list information](#)” on page 292 for a description of the reported data. The second part consists of one or more lines, each with the following information:

- Copy ... consequential number of migrated data copy
- Phase ... current phase the job is executing (for this copy)
- Drive ... name of the allocated FSE drive that will be used for writing (this copy)
- Medium ... barcode of the allocated FSE medium (that will store this copy)
- Volume ... current medium volume that is being written to
- Status ... descriptive current status of a job (copy)

Note: The “Copy” value is relevant only with migration jobs, and not with any other job type. With multiple copy migration jobs, this status line is printed for each copy that is migrating.

Options

- `-l, --list`

List FSE jobs currently running in the FSE installation or on the specified FSE partition if the *PartitionName* argument is specified. Additional option qualifiers can be used to narrow the scope of this option. See “[Job list information](#)” on page 292 for details on the information that is printed with this option.
- `-M, --migration`

List migration jobs.
- `-R, --recall`

List recall jobs.
- `-A, --admin`

List administrative jobs.
- `-C, --recovery`

List recovery jobs.
- `-T, --main`

List maintenance jobs.
- `PartitionName`

List FSE jobs running on the specified FSE partition only.
- `-D, --detail`

Display the retrieved data in *ParameterName: ParameterValue* format with each parameter in its own line. This option can only be used in combination with the `--list` option.
- `-t, --status`

Show status information for the FSE job with the specified identification number *JobID*. See “[Job list information](#)” on page 292 for details on the status information that is printed with this option.
- `JobID`

Identification number of the FSE job of which to show the status or change the priority.

- `-a, --abort`
Abort FSE jobs specified by the sequence of their identification numbers with the `JobID...` argument.
- `JobID...`
A sequence of identification numbers for FSE jobs to be aborted. Identification numbers must be separated by spaces.
- `-p, --priority`
If no additional option is specified, change the priority of the FSE job with identification number `JobID` to priority `Number`. If additional `--incr` option is specified, increase its current priority by value `Number`. If the additional `--decr` option is specified, decrease its current priority by value `Number`.
- `Number`
A value for target FSE job priority, priority increase or priority decrease.
- `-I, --incr`
Increase current priority of the FSE job with identification number `JobID` by value `Number`.
- `-D, --decr`
Decrease current priority of the FSE job with identification number `JobID` by value `Number`.
- `-h, --help`
Print the usage information for this FSE command.
- `-v, --version`
Print the copyright and version information for this FSE command.

Examples

- List FSE jobs which are running on FSE partition `partition_01`:

```
fsejob -l partition_01
```
- List all FSE migration and recall jobs which are running in FSE:

```
fsejob --list --migration --recall
```
- List FSE administrative jobs which are running on FSE partition `partition_new`. Display each of the retrieved parameters in a separate line:

```
fsejob -l -A partition_new -D
```
- Show status of the FSE job with identification number `20021203000092`:

```
fsejob --status 20021203000092
```


- Abort execution of FSE jobs with identification numbers 20021126000803 and 20021126000804:
`fsejob --abort 20021126000803 20021126000804`
- Set priority of FSE job with identification number 20021125000714 to 4096:
`fsejob --priority 20021206000614 4096`
- Increase priority of FSE job with identification number 20021127000049 by 1000:
`fsejob -p 20021206000614 -I 1000`

See also:

[fssystem](#), [fsepartition](#), [fsefile](#), [Introduction to the FSE system \(fseintro\)](#)

fselibrary

Manage configuration of the FSE libraries.

Syntax

```
fselibrary {-a | --add } ConfigFileName
fselibrary {-r | --remove } Name [-F | --force ]

fselibrary {-s | --show Name [RevisionNumber | {-H |
--history }]}
fselibrary {-m | --modify } Name ConfigFileName
fselibrary {-l | --list } [-D | --detail]

fselibrary {-i | --show-inventory } LibraryName
fselibrary {-u | --update-inventory } LibraryName [{-R |
--rescan }]}

fselibrary --disable Name [-F | --force ]
fselibrary --enable Name
fselibrary {-h | --help }
fselibrary {-v | --version }
```

Description

This FSE command manages the configuration of the FSE libraries. It is used to perform the following FSE administrator's tasks:

- configure a new or remove an existing FSE library,
- retrieve the configuration of an existing FSE library,
- reconfigure an existing FSE library,
- list names and status information for configured libraries in the FSE installation,
- show current status of the configured drives and slots in the operating FSE library (barcodes of the media in drives and the media assigned to slots),
- update Resource Management Database with current status of the drives and media in the operating FSE library,
- disable or enable operation of an FSE library.

Each FSE command communicates with the FSE system components through the Management Interface (`fse-mif`). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the `fse status` command. If the Management Interface is not running, start it with the `fse start` command.

After successful command execution, `fselibrary` notifies the FSE administrator with a suitable message, which is printed to the standard output stream (stdout). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (stderr).

Options

- `-a, --add`

Configure a new FSE library with parameters written in the configuration file *ConfigFileName*. This action checks the validity of the configuration file, copies the file to Configuration Database and inserts the parameter values into the Resource Management Database. FSE library names are unique within the FSE installation. If the operation is successful, a new Library Agent for this library is subsequently started. The Library Agent is one of the configuration-dependent FSE processes in the FSE installation, responsible for handling and interpreting requests addressed to the FSE library.
- *ConfigFileName*

The name of the FSE library configuration file to use.
- `-r, --remove`

Remove an existing FSE library from the FSE installation. This includes shutdown of the Library Agent for this library and removal of all configuration file revisions of this FSE library from Configuration Database. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is "No".
- *Name*

The name of the FSE library whose configuration is to be managed.
- `-F, --force`

Set forced mode for FSE library removal or disable operation. In this mode, the FSE administrator's confirmation of the FSE library removal or disabling is suppressed. This option can only be used together with the `--remove` or `--disable` option.
- `-s, --show`

If the argument *RevisionNumber* is specified, print the appropriate configuration revision for the FSE library *Name*, otherwise print the current configuration revision.

■ *RevisionNumber*

A non-negative integer value which represents the configuration revision to print.

■ -H, --history

Print the configuration history depth for the FSE library *Name*. The configuration history depth is the number of configuration file revisions stored in the Configuration Database for this FSE library. This option can only be used together with the --show option.

■ -m, --modify

Modify the configuration of the FSE library *Name* with the configuration file *ConfigFileName*. This action checks the validity of the new configuration file, copies it to Configuration Database as a new configuration revision and triggers the FSE library reconfiguration. During reconfiguration its Library Agent is temporarily disabled. It is shut down before reconfiguration and restarted afterward.

■ -l, --list

Display status information for configured FSE libraries within the FSE installation. The reported status information for an FSE library includes:

- Library ... library name
- Family ... library family (AIT, LTO)
- Type ... library type
- Host ... fully-qualified domain name (FQDN) of the host the library is attached to
- CtrlDev ... library control device file
- Status ... current library status (online, disabled)

■ -D, --detail

Retrieve additional FSE library information and display the retrieved data in *ParameterName: ParameterValue* format with each parameter in its own line. The following additional information is reported:

- ProductID ... ID string of the library model
- SerialNo ... serial number of the library
- VendorID ... ID string of the library manufacturer

- FirmwareRev ... revision of the firmware the library currently uses
- ScsiID ... ID string of the corresponding SCSI device
- LoadNr ... number of drive loads performed since the library was configured in the FSE system
- UnloadNr ... number of drive unloads performed since the library was configured in the FSE system
- Key ... internal RMDB key value assigned to the library

This option can only be used in combination with the `--list` option.

■ `-i` , `--show-inventory`

List names and status information for configured FSE drives in the FSE library *LibraryName* and indexes and status information for configured FSE slots in the same library. The reported status information for an FSE drive includes:

- Name ... drive name
- Type ... item type (drive)
- Status ... current drive status (online, in use, disabled, error)
- Medium ... barcode of the FSE medium that is loaded in the drive (`<empty>` string is displayed if the drive is empty)

Status information for an FSE slot includes:

- Name ... slot index that corresponds to the physical position of the slot inside the library
- Type ... slot type (repository, import/export, cleaning)
- Status ... current slot status (full, reserved, free)
- Medium ... barcode of the medium that is present in the slot. (`<empty>` string is displayed if the slot is free)

Note: A “reserved” slot is physically empty, but it is reserved for storage of a particular FSE medium that may be currently loaded in an FSE drive or put offline.

■ *LibraryName*

The name of the FSE library to show or whose inventory is to be updated.

■ `-u` , `--update-inventory`

Trigger an inventory update on the FSE library *LibraryName*.

■ `-R` , `--rescan`

Specify that a physical slot rescan must be done during the inventory update. This option can only be used together with the `--update-inventory` option.

- `--disable`
Disable operation of FSE library *Name*. Disabling the library means putting it offline in case it needs maintenance. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is "No".
- `--enable`
Enable operation of FSE library *Name*, which is currently disabled. Enabling the library means making it available to the FSE installation again.
- `-h, --help`
Print the usage information for this FSE command.
- `-v, -version`
Print the copyright and version information for this FSE command.

Examples

- Configure a new FSE library from the configuration file `main_library.fse`:

```
fselibrary -a main_library.fse
```
- Remove the configured FSE library named `lib_main`:

```
fselibrary -r lib_main
```
- Remove the configured FSE library named `lib_supplemental`; skip confirmation:

```
fselibrary -r lib_supplemental --force
```
- Show the current configuration of the FSE drive named `system_lib`:

```
fselibrary --show system_lib
```
- Show the configuration history depth for the FSE library named `lib_archive_2`:

```
fselibrary -s lib_archive_2 -H
```
- Copy current (latest) configuration revision of the FSE library named `storage_library` to a file `storage_library_cfg.fse` in the current directory:

```
fselibrary --show main_library >  
storage_library_cfg.fse
```

- Apply a new configuration for the FSE library named `lib_archive` from the configuration file `new_archive_library.fse`:
`fselibrary --modify lib_archive new_archive_library.fse`
- List names and states of the currently configured FSE libraries:
`fselibrary -l`
- Retrieve additional information about the currently configured FSE libraries:
`fselibrary --list -D`
- Show the inventory of the FSE library named `lib_email`:
`fselibrary -i lib_email`
- Trigger an inventory update on the FSE library named `lib_builds`:
`fselibrary -u lib_builds`
- Trigger an inventory update on the FSE Library named `lib_email`. Perform a physical barcode rescan:
`fselibrary --update-inventory lib_email --rescan`
- Disable the FSE library named `AIT_library`; skip confirmation:
`fselibrary --disable AIT_library --force`
- Enable the currently disabled FSE library named `LTO_library`:
`fselibrary --enable LTO_library`

File

- `/opt/fse/newconfig/fse-library.cfg` (on Linux)
`%InstallPath%\newconfig\fse-library.cfg` (on Windows)
Template for an FSE library configuration file.

See also:

[fse](#), [fse drive](#), [fse medium](#), [fse pool](#), [Introduction to the FSE system \(fseintro\)](#)

fsemedium

Manage configuration of FSE media. Format and initialize FSE media. Perform maintenance tasks on FSE media. Duplicate or reorganize FSE media. Recreate a copy of the unreadable data located on damaged FSE media.

Syntax

```
fsemedium {-a | --add } Barcode [{-O | --pool } PoolName ]
fsemedium {-r | --remove } Barcode [{-F | --force }]

fsemedium {-l | --list } [Barcode {-V | --volume }]
[-D | --detail]

fsemedium {-f | --format } Barcode [{-F | --force }]
[{-i | --init }] [--no-monitor ]
fsemedium {-i | --init } Barcode [{-V | --volume }]
VolumeNumber... [{-F | --force }] [--no-monitor ]

fsemedium --mark-unreliable Barcode [{-F | --force }]
fsemedium --mark-unusable Barcode [{-F | --force }]
fsemedium --mark-good Barcode [{-F | --force }]

fsemedium --close-volume Barcode VolumeNumber [--no-monitor ]
fsemedium --recreate-fri Barcode [{-V | --volume }]
VolumeNumber... [--no-monitor ]

fsemedium --duplicate SourceBarcode TargetBarcode
[--no-monitor ] [-F | --force ]
fsemedium --reorg Barcode [(-V | --volume ) VolumeNumber...]
[--no-monitor ]
fsemedium --copy-contents Barcode [--no-monitor ]

fsemedium {-h | --help }
fsemedium {-v | --version }
```

Description

This FSE command manages a single FSE medium. It is used to perform the following FSE administrator's tasks:

- configure a new or remove an existing FSE medium,
- list barcodes and status information for configured media in the FSE installation,
- show status information for medium volumes on configured FSE media,

- format and/or initialize a configured FSE medium,
- change an FSE medium status to unreliable, unusable or good,
- mark an FSE medium volume as full and handle its Fast Recovery Information appropriately,
- recreate missing Fast Recovery Information on disk from the migrated metadata,
- duplicate a medium to a medium of the same tape (copy all data and Fast Recovery Information from the source to the target medium),
- reorganize a medium or specified medium volumes preserving only the data that were determined to be important during the reorganizational scan job,
- recreate a redundant copy of the unreadable data located on damaged FSE media.

Each FSE command communicates with the FSE system components through the Management Interface (`fse-mif`). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the `fse status` command. If the Management Interface is not running, start it with the `fse start` command.

After a successful command execution, `fsemedium` notifies the FSE administrator with a suitable message, which is printed to the standard output stream (stdout). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (stderr).

Job progress monitoring

`fsemedium` is able to receive messages from the Management Interface to give you the ability to monitor the progress of the command.

Monitoring is enabled by default and can be disabled by using the `--no-monitor` option.

When monitoring is enabled, the `fsemedium` command reports events and other relevant information from the FSE processes as long as the job that `fsemedium` has started is running. After the job terminates, the command exits. If monitoring is disabled, the command returns the control immediately after it has started a corresponding job. In this case no information from the job is received, even in case of errors.

Monitoring can be canceled by pressing `^C` in the shell where `fsemedium` runs.

Note: The job will continue to run and will write progress to the `fse.log` file.

Options

- `-a, --add`

Add a new FSE medium with barcode *Barcode* to the FSE installation. If the additional option `--pool` with the argument *PoolName* is specified, assign this FSE medium to the specified FSE media pool. If this option is not specified, the medium entry is added to the Resource Management Database but the medium itself is left unassigned with its configurable parameters unset.

Note: An FSE medium must be assigned to some FSE media pool before it can be used.

- *Barcode*

The identifier of the FSE medium that is being managed. It is usually a numerical barcode. For disk media, the barcode is the relative part of the mount point (for example: `dm_001`, `dm_002`).

- `-O, --pool`

Assign a new FSE medium with barcode *Barcode* to the FSE media pool *PoolName*. This option can only be used together with the `--add` option.

- *PoolName*

The name of the FSE media pool to add the new medium to.

■ `-r, --remove`

Remove an existing FSE medium from the FSE installation. This is possible if all FSE medium volumes on the medium are empty. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is "No".

■ `-l, --list`

If this option is used alone it lists barcodes and status information of the FSE media that is configured within the FSE installation. If the additional argument *Barcode* with the `--volume` option is specified then it shows the status of all medium volumes on the medium with the specified barcode. If the argument combination `--list --volume` is used then the status of all medium volumes on all configured FSE media is displayed. The reported status information for an FSE medium includes:

- medium barcode
- medium type (AIT-1, AIT-2, AIT-3, SAIT, LTO 1, LTO 2, DDS, DISK, UNKNOWN)
- name of the FSE media pool the medium belongs to
- current status of the medium (uninitialized, free, open, in use, full, unreliable, unusable)
- current location of the medium in an FSE library slot (library name and slot number) or in an FSE drive (drive name)
- number of volumes on the medium
- index of system volume (value 0 is displayed if it is not present)

Status information for an FSE medium volume includes:

- name of the corresponding FSE medium
- index of the volume on the medium
- medium volume type (data, system)
- medium volume size
- current amount of volume free space
- percentage of currently occupied volume space
- current medium volume status (empty, open, full, recovery, scanned, in reorg, recycled, error)

■ `-V, --volume`

There are four kinds of usage of this option in combination with other options:

- if the `--list` option is specified, show the status of all medium volumes on the medium with the *Barcode* barcode, or status of the medium volumes on all media configured in FSE (if *Barcode* is omitted);
- if the `--init` option is specified, initialize only medium volumes *VolumeNumber...* (the medium must already be formatted);
- if the `--recreate-fri` option is specified, populate the system volume on the medium with Fast Recovery Information from this volume (if it is closed).
- if the `--reorg` option is specified, reorganize only medium volumes with consecutive numbers *VolumeNumber...*

■ `-D, --detail`

Retrieve additional FSE medium or FSE medium volume information and display the retrieved data in *ParameterName: ParameterValue* format with each parameter in its own line.

If this option is used with the `--list` option, the following additional information is reported:

- *LibName* ... name of the library the medium is located in
- *SlotAddr* ... address of the library slot assigned to the medium
- *SlotType* ... type of the library slot assigned to the medium
- *DrvName* ... name of the drive the medium is loaded into (if any)
- *SerialNo* ... medium serial number
- *BlockSize* ... block size used on FSE medium volumes on the medium
- *SysVolSize* ... size of the system volume of the medium (if it exists)
- *CurVol* ... consecutive number of the medium volume on which the medium tape is currently positioned
- *Key* ... internal RMDB key value assigned to the medium

If this options is used with the `--list` and `--volume` options, the following additional information is reported:

- *UUID* ... universal unique ID of the medium volume
- *AccessNum* ... number of accesses to the data on the volume
- *WriteNum* ... number of writings to the volume

- OwriteNum ... number of overwritings of the data on the volume
- InitTime ... date and time of the last volume initialization event
- AccessTime ... date and time of the last volume access event
- WriteTime ... date and time of the last volume write event
- OwriteTime ... date and time of the last volume overwrite event
- LastVer ... date and time of the last volume verification event
- ValidData ... amount of the data on the volume that was determined to be important by the reorganizational scan job (in MB, the value equals 0 if the volume has not been scanned)
- TotalData ... amount of the data on the volume (in MB)

■ `-f, --format`

Format a new or an already used FSE medium with barcode *Barcode*. If the additional option `--init` is specified then the initialization of the medium volumes is performed as well. Formatting an FSE medium means partitioning it into FSE medium volumes, according to the parameters specified by the corresponding FSE media pool configuration. For FSE disk media, formatting creates subdirectories for each of the volumes. The name of the directory will be the volume number.

Note: An FSE medium must be assigned to an FSE media pool before it can be formatted. If the medium is not empty, formatting it is allowed only if the additional option `--force` is specified. Backup media in the backup media pool should be formatted, but not initialized.

■ `-F, --force`

Set forced mode for FSE medium remove, format, initialize, mark unreliable, mark unusable or mark good operation. There are three kinds of usage of this option in combination with other options. This option will:

- if the `--format` option is specified, allow formatting of non-empty FSE media if it is not referenced by FSC;
- if the `--init` option is specified as the first option after the command name, allow initialization of non-empty FSE medium volumes if it is not referenced by FSC;

— if the `--remove`, `--mark-unreliable`, `--mark-unusable` or `--mark-good` option is specified, suppress the requirement for the FSE administrator's confirmation of FSE medium removal, marking the medium unreliable, marking it unusable or marking it good.

■ `-i`, `--init`

Initialize the specified or all FSE medium volumes on the FSE medium with barcode *Barcode*. Initializing an FSE medium volume means (over-)writing its volume header, thus preparing the volume for use in the FSE installation.

Note: An FSE medium volume must be associated (via its FSE medium and FSE media pool) to an FSE partition before it can be initialized. If the medium volume is not empty, its initialization is allowed only if the additional option `--force` is specified. Backup media should not be initialized, but formatted only.

■ `--no-monitor`

Disable monitoring of the formatting, initialization, maintenance, or reorganization job which is to be performed on the FSE medium. Monitoring is enabled by default. See “[Job progress monitoring](#)” on page 303 for details on monitoring.

■ `VolumeNumber...`

A non-negative integer value (or a sequence of values) which represents the positions of FSE medium volumes on the medium layout.

■ `--mark-unreliable`

Disallow writing to the FSE medium with barcode *Barcode*. From this moment on, the data on the medium can only be read. This is done by setting the status of this medium in the Resource Management Database to unreliable. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is “No”.

■ `--mark-unusable`

Disallow reading from and writing to the FSE medium with barcode *Barcode*. From this moment on, the data on the medium cannot be read, nor can the medium be written to. This is done by setting the status of the medium in the Resource Management Database to unusable. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is “No”.

■ `--mark-good`

Make the FSE medium with barcode *Barcode* fully operational for normal use. From this moment on, the data on the medium can be read and the medium can be written to. This is done by setting the status of the medium in the Resource Management Database to good. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is "No".

■ `--close-volume`

Mark the specified FSE medium volume as full and set its status in the Resource Management Database accordingly. This operation also saves Fast Recovery Information (FRI) to the end part of the medium volume and to the medium's system volume, if it is present. If writing FRI to the medium fails an appropriate warning is displayed.

Note: A full FSE medium volume is no longer written to. Marking an FSE medium volume as full is useful in situations where, owing to problems with a particular medium volume, the status of a whole FSE medium is set to unreliable. The medium remains unused despite the fact that other medium volumes are error-free. In such cases the unreliable medium volume is marked as full and its containing medium as good to make the other medium volumes available to the FSE installation.

■ `--recreate-fri`

Recreate missing Fast Recovery Information on disk from the migrated metadata on the specified FSE medium or FSE medium volume. For closed medium volumes, the recreated Fast Recovery Information is also copied to the end part of these volumes.

Note: Fast Recovery Information on disk speeds up the eventual File System Catalog recovery.

■ `--duplicate`

Duplicate a source medium to a target medium of the same tape. All data will be copied bit-by-bit: medium volume header (including medium volume UUID), medium volume data and Fast Recovery Information (from data volume as well as system volume). The target medium must not belong to any media pool that is assigned to a partition. If you want to replace a medium with its duplicate, label the duplicate medium with the original barcode and replace the original with the copy.

- `--reorg`

Start the media reorganization job for the specified FSE medium or FSE medium volumes. If the `--no-monitor` option is not specified, the progress of the reorganization job will be reported to the command output. Note that you can only start the reorganization job if the FSE medium volumes involved in the process have already been scanned for obsolete data. Note that the reorganization process can take a considerable amount of time.
- `--copy-contents`

Recreate a redundant copy of the unreadable data that is located on a damaged medium. The copy is recreated from the corresponding data in another FSE media pool assigned to the same FSE partition, if it exists. The recreated data is created on the open or empty medium volumes in the FSE media pool to which the damaged medium is assigned. If the `--no-monitor` option is not specified, the progress of the maintenance job that performs the recreation will be reported to the command output.
- `h, --help`

Print the usage information for this FSE command.
- `-v, --version`

Print the copyright and version information for this FSE command.

Examples

- Add a new FSE medium with barcode `backup_tape_05` to the FSE configuration:

```
fsemedium --add backup_tape_05
```
- Add a new FSE medium with barcode `backup_tape_10` to the FSE configuration and assign it to FSE media pool `pool_bkp`:

```
fsemedium -a backup_tape_10 --pool pool_bkp
```
- Remove the FSE medium with barcode `backup_tape_13` from the FSE configuration:

```
fsemedium -r backup_tape_13
```
- Remove the FSE medium with barcode `backup_tape_78` from the FSE configuration; skip confirmation:

```
fsemedium -r backup_tape_78 -F
```
- List the barcodes of the configured FSE media within the FSE installation:

```
fsemedium --list
```


- Show the status of all medium volumes on the FSE medium with barcode `archive_tape_ABB`:
`fsemedium -l archive_tape_ABB -V`
- Format the FSE medium with barcode `work_34`; disable monitoring:
`fsemedium -f work_34 --no-monitor`
- Format the FSE medium with barcode `work_35` in forced mode and initialize all FSE medium volumes on it:
`fsemedium --format work_35 -F --init`
- Initialize all medium volumes on the FSE medium with barcode `work_offline_45` in forced mode:
`fsemedium -i work_offline_45 --force`
- Initialize volume number 1 on the FSE medium with barcode `tape_4a`; disable monitoring:
`fsemedium --init tape_4a --volume 1 --no-monitor`
- Disallow further writing to the FSE medium with barcode `old_tape_23`:
`fsemedium --mark-unreliable old_tape_23`
- Pull the FSE medium with barcode `dailyFiles_A` out of operation; skip confirmation:
`fsemedium --mark-unusable dailyFiles_A --force`
- Make the FSE medium with barcode `dailyFiles_B` fully available for writing and reading:
`fsemedium --mark-good dailyFiles_B`
- Mark erroneous volume number 3 on the FSE medium with barcode `ait_60` as full and populate the volume and the medium's system volume with Fast Recovery Information:
`fsemedium --close-volume ait_60 3`
- Recreate Fast Recovery Information for data volumes 1, 2 and 4 on FSE medium with barcode `weeklyFiles01`; disable monitoring:
`fsemedium --recreate-fri weeklyFiles01 -V 1 2 4 --no-monitor`
- Duplicate source medium with barcode `tape_44` to target medium with barcode `copytape_44`; disable monitoring:
`fsemedium --duplicate tape_44 copytape_44 --no-monitor`

- Start the media reorganization job for all medium volumes on the medium with barcode `tape_55`; disable monitoring:
`fsemedium --reorg tape_55 --no-monitor`
- Start the media reorganization job for medium volumes 1, 2, 5, and 7 on the medium with barcode `tape_55`:
`fsemedium --reorg tape_56 --volume 1 2 5 7`
- Recreate a copy of the files located on medium with barcode `damaged_med`:
`fsemedium --copy-contents damaged_med`

See also:

[fselibrary](#), [fsedrive](#), [fsepool](#), [fsepartition](#), [Introduction to the FSE system \(fseintro\)](#)

fsepartition

Manage configuration of the FSE partitions. Scan the FSE partitions for obsolete data and report the corresponding statistics.

Syntax

```
fsepartition {-a | --add } ConfigFileName
fsepartition {-r | --remove } Name [-F | --force ]

fsepartition {-s | --show } Name [RevisionNumber |
    {-H | --history } ]
fsepartition {-m | --modify } Name ConfigFileName
fsepartition {-l | --list } [-D | --detail]

fsepartition {-t | --status } Name

fsepartition --disable Name [-F | --force ]
fsepartition --enable Name

fsepartition --reorg-scan Name ExpirationDate
    [NumberOfGenerations]
fsepartition --reorg-stat Name [SlackThreshold%]

fsepartition {-h | --help }
fsepartition {-v | --version }
```

Description

This FSE command manages the configuration of FSE partitions. It is used to perform the following FSE administrator's tasks:

- configure a new or remove an existing FSE partition,
- retrieve the configuration of an existing FSE partition,
- reconfigure an existing FSE partition,
- list names and current status for configured partitions in the FSE installation,
- show current status of an FSE partition,
- disable or enable an FSE partition,
- scan FSE media that belong to an FSE partition for obsolete data,
- display the reorganizational scan statistics for the FSE media that belong to an FSE partition.

Each FSE command communicates with the FSE system components through the Management Interface (`fse-mif`). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the `fse status` command. If the Management Interface is not running, start it with the `fse start` command.

After a successful command execution, `fsepartition` notifies the FSE administrator with a suitable message, which is printed to the standard output stream (`stdout`). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (`stderr`).

Partition type

When configuring a new FSE partition with the `--add` option, the FSE administrator can define two different FSE partition types: *regular* and *WORM*. The selected partition type affects the characteristics of the underlying HSM file system. A *regular* FSE partition type uses a common HSM file system with all its features and capabilities, whereas a *WORM* FSE partition type uses a modified HSM file system—a WORM HSM file system—which forbids changing the contents of files once they are written to the file system.

The preferred FSE partition type is specified by setting the appropriate value for a variable in the FSE partition configuration template.

Note: A WORM FSE partition does not necessarily need to have WORM FSE media pools assigned. See the `fsepool` man page for details on WORM FSE media pools.

Partition status information

Partition status information printed using the `--status` option is the following:

- partition name
- current partition status (disabled, stopped, running, mounted, unknown)
- fully-qualified domain name (FQDN) of the machine that hosts the partition
- mount point of the HSM file system that belongs to the partition
- number of currently running migration jobs on the partition
- number of currently running recall jobs on the partition
- number of currently running maintenance jobs on the partition
- number of currently running recovery jobs on the partition

For each FSE media pool currently assigned to the partition, the following information is printed:

- *Pool* ... media pool name
- *Size* ... total size on the FSE media in this media pool [MB]
- *Avail* ... total available space on the FSE media in this media pool [MB]
- *Used* ... total used space on the FSE media in this media pool [%]
- *SysSize* ... total size of all system volumes on the FSE media in this media pool [MB]
- *SysAvail* ... total available space on all system volumes on the FSE media in this media pool [MB]
- *SysUsed* ... total used space on all system volumes on the FSE media in this media pool [%]
- *#Good* ... number of good FSE media in this media pool
- *#Unusable* ... number of unusable FSE media in this media pool
- *#Unreliable* ... number of unreliable FSE media in this media pool
- *#Uninit* ... number of uninitialized FSE media in this media pool

Partition states have the following meaning:

Status	HSM File System	Partition	Partition Manager	Normal Operation
disabled	not mounted	disabled	not running	yes
stopped	*	enabled	not running	no
running	not mounted	enabled	running	yes
mounted	mounted	enabled	running	yes
unknown	*	*	*	no

* not relevant

Note: Situations indicated by rows with the word **no** in the column Normal Operation indicate that the FSE administrator's intervention is necessary to restore normal FSE operation.

Options

- `-a, --add`

Configure a new FSE partition with parameters written in the configuration file *ConfigFileName*. This action checks the validity of the configuration file, copies the file to the Configuration Database, and inserts the parameter values into the Resource Management Database. FSE partition names are unique within one FSE installation. A new Partition Manager for this partition is started after the validation checks pass. The Partition Manager is one of the configuration-dependent FSE processes in the FSE installation. It is responsible for managing the data flow between the HSM file system and FSE media in the FSE media pools that belong to this FSE partition.
- *ConfigFileName*

The name of the FSE partition configuration file to use.
- `-r, --remove`

Remove an existing FSE partition from the FSE installation. This includes shutdown of the Partition Manager for this partition and removal of:

 - all configuration file revisions of this FSE partition from the Configuration Database,
 - FSC and HSMDB database files,
 - media volume information from the RMDB.

Note: Unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is "No".



Caution: Before using this option, the FSE administrator must be familiar with concepts of FSE system operation and must be well aware of the existing data-loss risk that can result from improper use of the `--remove` option.

- *Name*

The name of the FSE partition whose configuration is to be managed or on which to perform an operation.
- `-F, --force`

Set forced mode for the FSE partition removal. In this mode, the FSE administrator's confirmation for FSE partition removal is suppressed. This option can only be used together with the `--remove` or `--disable` option.

- `-s, --show`

If the argument *RevisionNumber* is specified, print the appropriate configuration revision for the FSE partition *Name*, otherwise print the current configuration revision.
- *RevisionNumber*

A non-negative integer value which represents the configuration revision to print.
- `-H, --history`

Print the configuration history depth for the FSE partition *Name*. The configuration history depth is the number of configuration file revisions stored in the Configuration Database for this FSE partition. This option can only be used together with the `--show` option.
- `-m, --modify`

Modify the configuration of FSE partition *Name* with the configuration file *ConfigFileName*. This action validates the new configuration file, copies it to the Configuration Database as a new configuration revision and triggers the FSE partition reconfiguration. During reconfiguration its Partition Manager remains running and operational.
- `-l, --list`

Display status information for configured FSE partitions within the FSE installation. The reported information for an FSE partition includes:

 - *Partition ...* partition name
 - *Status ...* current partition status (disabled, stopped, running, mounted, unknown)
 - *Host ...* fully-qualified domain name (FQDN) of the machine that hosts the partition
 - *Mountpoint ...* mount point of the HSM file system that belongs to the partition
- `-D, --detail`

Retrieve additional FSE partition information and display the retrieved data in *ParameterName: ParameterValue* format with each parameter in its own line. The following additional information is reported:

 - *UUID ...* universal unique ID assigned to the partition
 - *FSID ...* file system identifier in the form of a **major:minor** string

- MigIdTime ... timestamp (in the internal FSE format) of the latest migration job that was started on the partition
- MigIdCount ... indicator of multiple migration jobs started on the partition in a period of one second

This option can only be used in combination with the `--list` option.

- `-t, --status`

Show the status of the FSE partition *Name*. See “[Partition status information](#)” on page 314 for details on the status information that is displayed with this option.

- `--disable`

Disable operation of the FSE partition *Name*. Note that unless the additional option `--force` is specified, the command requires the FSE administrator’s confirmation before performing this operation. The default response to the confirmation question is “No”.

- `--enable`

Enable operation of FSE partition *Name*, which is currently disabled.

- `--reorg-scan`

Scan FSE medium volumes on all media that belong to the FSE partition *Name* for obsolete data using the mandatory *ExpirationDate* and optional *NumberOfGenerations* parameters. The scan statistics are recorded in the corresponding File System Catalog, to be used by the reorganization job. Note that the scanning process can take a considerable amount of time.

- *ExpirationDate*

Treat all migrated file data generations created after *ExpirationDate* as important. You have to specify the *ExpirationDate* parameter in the form

"YYYY/MM/DD HH:mm:ss" or "YYYY/MM/DD". In the latter case, the time 00:00:00 is assumed. Note that depending on the value of the

NumberOfGenerations parameter (if it is specified), not all valid generations may be preserved by the reorganization process.

- *NumberOfGenerations*

Treat at most *NumberOfGenerations* latest migrated file data generations as important. With this parameter, you can reduce the number of generations that would be preserved using the *ExpirationDate* parameter alone.

- `--reorg-stat`

Display the statistics gathered by the reorganizational scan job that was run on the FSE partition *Name*. The reported values/statistics are displayed for each relevant FSE medium, and are the following:

- Barcode ... FSE medium barcode
- VolNum ... number of FSE medium volumes on the medium
- ValidData ... amount of important (non-obsolete) data stored on the medium (in MB)
- Total ... total amount of data stored on the medium (in MB)
- SlackSpace ... the percentage of storage space on the medium that is occupied by the data which was treated as obsolete by the reorganizational scan job

■ *SlackThreshold%*

Omit the media with a slack space percentage lower than *SlackThreshold%* from the report generated by the `--reorg-stat` option. This option can only be used in combination with the `--reorg-stat` option.

■ `-h, --help`

Print the usage information for this FSE command.

■ `-v, --version`

Print the copyright and version information for this FSE command.

Examples

- Configure a new FSE partition from the configuration file `client_02_part_05.fse`:

```
fsepartition --add client_02_part_05.fse
```

- Remove the configured FSE partition named `client_08_part_04`:

```
fsepartition -r client_08_part_04
```

- Remove the configured FSE partition named `client_03_part_02`; skip confirmation:

```
fsepartition -r client_03_part_03 --force
```

- Show the current configuration of the FSE partition named `client_01_part_02`:

```
fsepartition --show client_01_part_02
```

- Show configuration revision 2 of the FSE partition named `client_01_part_04`:

```
fsepartition -s client_01_part_04 2
```

- Show the configuration history depth for the FSE partition named `client_01_documents`:

```
fsepartition -s client_01_documents -H
```

- Copy the current (latest) configuration revision of the FSE partition named `mail_partition_01` to a file `mail_partition_01_cfg.fse` in the current directory:

```
fsepartition --show mail_partition_01 >  
mail_partition_01_cfg.fse
```

- Copy configuration revision 3 of the FSE partition named `fse_part_001` to a file `fse_part_001_cfg_3.fse` in the current directory:

```
fsepartition -s fse_part_001 3 > fse_part_001_cfg_3.fse
```

- Apply a new configuration for the FSE partition named `client_02_part_05` from the configuration file `client_02_part_05_modified.fse`:

```
fsepartition --modify client_02_part_05  
client_02_part_05_modified.fse
```

- List names and states of the currently configured FSE partitions:

```
fsepartition --list
```

- Show status information for FSE partition named `mail_partition_01`:

```
fsepartition --status mail_partition_01
```

- Disable the FSE partition named `client_03_astro_pics`; skip confirmation:

```
fsepartition --disable client_03_astro_pics -F
```

- Enable the currently disabled FSE partition named `client_01_temporary_store`:

```
vpartment --enable client_01_temporary_store
```

- Run a reorganizational scan job on the FSE partition named `temporary_part`; treat all migrated data generations created after noon of 15th July, 2004 as important:

```
fsepartition --reorg-scan temporary_part "2004/07/15 12:00:00"
```

- Run a reorganizational scan job on the FSE partition named `very_temporary_part`; treat at most 15 latest migrated data generations created after 10:00 AM of 20th September, 2003 as important:

```
fsepartition --reorg-scan very_temporary_part  
"2003/09/20 10:00:00" 15
```

- Display the reorganizational statistics gathered by the reorganizational scan job for the FSE partition named `client_03_astro_pics`; omit reporting the FSE media with FSE slack space percentage lower than 70%:

```
fsepartition --reorg-stat client_03_astro_pics 70
```

Files

- `/opt/fse/newconfig/fse-partition.cfg` (Linux)
`%InstallPath%\newconfig\fse-partition.cfg` (Windows)

Template for an FSE partition configuration file.

- `/opt/fse/newconfig/fse-partition-CLI-DISK.cfg` (Linux)
`%InstallPath%\newconfig\fse-partition-CLI-DISK.cfg` (Windows)

Template for an FSE partition configuration file, with the parameters preset so that the partition policies are more suitable for use with FSE disk media.

- `/opt/fse/newconfig/fse-partition-NEVER-RELEASE.cfg` (Linux)
`%InstallPath%\newconfig\fse-partition-NEVER-RELEASE.cfg` (Windows)

Template for an FSE partition configuration file, with the parameters preset so that releases do not occur on the FSE partition until the free storage space on the partition is exhausted.

See also:

[fse](#), [fsepool](#), [fsemedium](#), [fsesystem](#), [fsejob](#), [fserecover](#), [Introduction to the FSE system \(fseintro\)](#)

fsepool

Manage the configuration of FSE media pools. Report the FSE media pool related statistics gathered during the scan for obsolete data.

Syntax

```
fsepool {-a | --add } ConfigFileName
fsepool {-r | --remove } Name [-F | --force ]

fsepool {-s | --show } Name [RevisionNumber |
    {-H | --history } ]
fsepool {-m | --modify } Name ConfigFileName
fsepool {-l | --list } [-D | --detail ]

fsepool --reorg-stat

fsepool {-h | --help }
fsepool {-v | --version }
```

Description

This FSE command manages configuration of the FSE media pools. It is used to perform the following FSE administrator's tasks:

- configure a new or remove an existing FSE media pool,
- retrieve the configuration of an existing FSE media pool,
- reconfigure an existing FSE media pool,
- list names and status information for configured media pools in the FSE installation,
- report the FSE media pool related statistics gathered during the scan for obsolete data.

Each FSE command communicates with the FSE system components through the Management Interface (`fse-mif`). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the `fse status` command. If the Management Interface is not running, start it with the `fse start` command.

After a successful command execution, `fsepool` notifies the FSE administrator with a suitable message, which is printed to the standard output stream (stdout). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (stderr).

Media pool types

There are three FSE media pool types that can be specified when configuring a new FSE media pool: *regular*, *WORM* and *backup*.

Regular FSE media pools are used for storing data migrated from an HSM file system. FSE media that belong to this type of media pool can be overwritten, reorganized, re-initialized and re-formatted.

WORM FSE media pools differ from regular FSE media pools in one important characteristic: it is impossible to overwrite data on the media that belong to this kind of media pool. Such media are called WORM media. Data already migrated to a WORM medium cannot be overwritten, it can only be appended to. A WORM medium cannot be reorganized, re-initialized or re-formatted. A WORM FSE media pool is therefore suitable for storing permanent archives for security or legal purposes.

The *backup* FSE media pool type is essentially different from the other two FSE media pool types. FSE media in this type of media pool do not store user data migrated from an HSM file system; they hold backup copies of the FSE system state created by the `fsebackup` command. To be able to backup and restore the FSE installation, there must be a backup FSE media pool configured to allow FSE backup media allocation by the `fsebackup` and `fserestore` commands. A backup pool should be configured with only one volume. Backup media in this pool should be formatted, but not initialized. For details about FSE backup, see the `fsebackup` man page.

Note: There can be only one backup FSE media pool configured in an FSE system.

Options

- `-a, --add`

Configure a new FSE media pool with parameters written in the configuration file `ConfigFileName`. This action checks the validity of the configuration file, copies the file to the Configuration Database, and inserts the parameter values into the Resource Management Database. FSE media pool names are unique within the FSE installation.

- `ConfigFileName`

The name of the FSE media pool configuration file to use.

- `-r, --remove`

Remove an existing FSE media pool from the FSE installation. This includes removal of all configuration file revisions of this FSE media pool from the Configuration Database. Note that unless the additional option `--force` is specified, the command requires the FSE administrator's confirmation before performing this operation. The default response to the confirmation question is "No".
- Name
The name of the FSE media pool whose configuration is to be managed.
- `-F, --force`

Set forced mode for the FSE media pool removal. In this mode, the FSE administrator's confirmation of the FSE media pool removal is suppressed. This option can only be used together with the `--remove` option.
- `-s, --show`

If the *RevisionNumber* argument is specified, print the appropriate configuration revision for the FSE media pool *Name*, otherwise print the current configuration revision.
- *RevisionNumber*

A non-negative integer value which represents the configuration revision to print.
- `-H, --history`

Print the configuration history depth for the FSE media pool *Name*. The configuration history depth is the number of configuration file revisions stored in the Configuration Database for this FSE media pool. This option can only be used together with the `--show` option.
- `-m, --modify`

Modify the configuration of FSE media pool *Name* with the configuration file *ConfigFileName*. This action checks the validity of the new configuration file, copies it to the Configuration Database as a new configuration revision, and triggers the FSE media pool reconfiguration.

■ `-l, --list`

List names and status information for configured FSE media pools within the FSE installation. The reported information for an FSE media pool includes:

- *Pool* ... media pool name
- *Family* ... media pool family (AIT, SAIT, LTO, DISK)
- *Type* ... media pool type (regular, WORM, backup)
- *Partition* ... name of the FSE partition the media pool is assigned to
- *BlkSize* ... block size on the corresponding FSE media
- *#Vol* ... number of volumes on the corresponding FSE media
- *SysVol* ... system volume position (first, middle, last, none) on the corresponding FSE media
- *SysVol[MB]* ... size of the system volume on the corresponding FSE media (if it is present)

See “[Media pool types](#)” on page 323 for description and use of different FSE media pool types.

■ `-D, --detail`

Retrieve additional FSE media pool information and display the retrieved data in *ParameterName: ParameterValue* format with each parameter in its own line. The following additional information is reported:

- *UUID* ... universal unique ID of the media pool
- *PartUUID* ... universal unique ID of the FSE partition to which the media pool is assigned
- *Key* ... internal RMDB key value assigned to the media pool

This option can only be used in combination with the `--list` option.

■ `--reorg-stat`

Display the statistics gathered by the reorganizational scan job that was run on the FSE partition to which the FSE media pool belongs. The reported values/statistics are the following:

- *Name* ... media pool name
- *NumOfMedia* ... number of FSE media that belong to the media pool
- *ValidData* ... amount of important (non-obsolete) data stored on the media of the media pool (in MB)

- Total ... total amount of data stored on the media of the media pool (in MB)
- SlackSpace ... the percentage of storage space on FSE media that is occupied by the data which was treated as obsolete by the reorganizational scan job
- `-h, --help`
Print the usage information for this FSE command.
- `-v, --version`
Print the copyright and version information for this FSE command.

Examples

- Configure a new FSE media pool from the configuration file `mail_archive_pool.fse`:

```
fsepool --add mail_archive_pool.fse
```
- Remove the configured FSE media pool named `system_archive_pool`:

```
fsepool --remove system_archive_pool
```
- Remove the configured FSE media pool named `system_backup_pool`; skip confirmation:

```
fsepool --remove system_backup_pool -F
```
- Show the current configuration of the FSE media pool named `backup_pool`:

```
fsepool -s backup_pool
```
- Show the configuration history depth for the FSE media pool named `proj_archive_pool`:

```
fsepool --show proj_archive_pool -H
```
- Copy configuration revision 5 of the FSE media pool named `mail_archive_pool` to a file `mail_archive_pool_cfg_5.fse` in the current directory:

```
fsepool -s mail_archive_pool 15 > mail_archive_pool_cfg_5.fse
```
- Apply a new configuration for the FSE media pool named `system_archive_pool` from the configuration file `new_system_archive_pool.fse`:

```
fsepool --modify system_archive_pool new_system_archive_pool.fse
```


- Retrieve additional information about the currently configured FSE media pools:
`fsepool -l -D`
- Display the reorganizational statistics gathered by the reorganizational scan job:
`fsepool --reorg-stat`

Files

- `/opt/fse/newconfig/fse-pool-AIT.cfg` (Linux)
`%InstallPath%\newconfig\fse-pool-AIT.cfg` (Windows)
Template for a configuration file for an FSE media pool with AIT media.
- `/opt/fse/newconfig/fse-pool-LTO.cfg` (Linux)
`%InstallPath%\newconfig\fse-pool-LTO.cfg` (Windows)
Template for a configuration file for an FSE media pool with LTO media.
- `/opt/fse/newconfig/fse-pool-DISK.cfg` (Linux)
`%InstallPath%\newconfig\fse-pool-DISK.cfg` (Windows)
Template for a configuration file for an FSE disk media pool.
- `/opt/fse/newconfig/fse-pool-AIT-MANY-SMALL-FILES.cfg` (Linux)
`%InstallPath%\newconfig\fse-pool-AIT-MANY-SMALL-FILES.cfg` (Windows)
Template for a configuration file for an FSE media pool with AIT media, with the parameters preset so that the media pool is suitable for use with FSE partitions with a large number of relatively small files.

See also:

[fsepartition](#), [fsemedium](#), [fselibrary](#), [fsedrive](#), [fsebackup](#), [fserestore](#), [Introduction to the FSE system \(fseintro\)](#)

fserecover

Recover a File System Catalog from its redundant duplicate data sources. Recover an HSM file system from an existing Name Space Catalog.

Syntax

```
fserecover { -f | --fsc } PartitionName
fserecover { -i | --hsmfs } PartitionName

fserecover { -h | --help }
fserecover { -v | --version }
```

Description

`fserecover` is used for regenerating:

- a File System Catalog (FSC) for the specified FSE partition in situations where it is inconsistent, damaged, or missing.
- an HSM file system (HSM FS) for the specified FSE partition in situations where it is inconsistent, damaged, or missing.

FSC recovery procedure

A File System Catalog is regenerated either from Fast Recovery Information (FRI) blocks, if they are present, or from the migrated metadata. For safety reasons, FRI blocks have multiple copies on different levels of the FSE storage hierarchy. Their locations are the temporary files on disk, system volumes on FSE media, and the end part of each data volume on FSE media.

The regeneration procedure always tries to choose the fastest recovery method and achieve the shortest recovery time. It starts by searching for FRI in the following order:

1. on disk (the fastest method),
2. on system volumes on FSE media (fast method),
3. at the end of each data volume on FSE media (slow method).

If FRI is missing and none of the above methods can be used, the procedure reverts to recovering from migrated metadata (the slowest method). If even this method fails, the recovery process is stopped and an appropriate error is written to `FSE error.log`.

Recovery uses all available FSE drives for accessing multiple volumes at the same time in order to speed up the recovery operation. The FSE partition for which the File System Catalog recovery is running is blocked for file accesses and all incoming jobs.

Note: If FSE processes are stopped during the recovery operation, recovery will continue when the FSE processes are re-started.

HSM file system recovery procedure

An HSM file system is regenerated from the Name Space Catalog (NSC), which holds all necessary information about the HSM file system.

To recover an HSM file system from scratch, the FSE administrator must:

1. unmount the inconsistent HSM file system,
2. format the HSM file system,
3. unmount the empty HSM file system (on Windows FSE clients only),
4. start recovery of the HSM file system.

The recovery job can be stopped with the command:

```
fsejob -a <jobID>
```

If recovery of the HSM file system is re-started, it will resume recovery from the last recovered fileID. This fileID is retrieved from the following file:

Linux:

```
/var/opt/fse/tmp/.<partition name>_hsmfs_recover
```

Windows:

```
%InstallPath%\var\tmp\.<partition name>_hsmfs_recover
```

This file provides data for incremental recovery in the event of a power failure or other inconvenient situation. If the file does not exist then recovery will start from fileID 1. If the recovery must be re-started from the beginning, then this file must be deleted. At the end of a successful recovery operation the file will be deleted.

Recovery applies to native file systems (ext3 on Linux, NTFS on Windows).

The file states after recovery are as follows:

- After the recovery of HSM FS, I-node information will be inconsistent with the old data in FSC and FRI. I-nodes of the files are not the same as before recovery.

- File access and create times will be set properly, other file time attributes may not be set during recovery.
- Files will be offline.
- Directories may not have proper change, modify, and access times set. This depends on the files that are restored after each directory is restored.

Limitations

- Symbolic links cannot be recovered because they are not written to tape (there is no migration of symbolic links to FSE media).
- Windows Alternate Data Streams (ADS) is not fully supported for recovery (recall of ADS is not supported). The names of ADS will be stored in NSC, but not data, therefore they will not be recoverable until they can be recalled from tape.
- It is not possible to stop recovery, mount the HSM file system as hsmfs, use it for migration, and later resume recovery.

Options

- `-f, --fsc`
Start recovery of the File System Catalog for FSE partition *PartitionName*. See “[FSC recovery procedure](#)” on page 328 for details of this process.
- `PartitionName`
The name of the FSE partition whose File System Catalog recovery or HSM file system recovery is to be triggered.
- `-i, --hsmfs`
Start recovery of the HSM file system for FSE partition *PartitionName*. See “[HSM file system recovery procedure](#)” on page 329 for details of this process.
- `-h, --help`
Print the usage information for this FSE command.
- `-v, --version`
Print the copyright and version information for this FSE command.

Examples

- Start recovery of the File System Catalog for FSE partition `fse_fs01`:

```
fserrecover --fsc fse_fs01
```
- Start recovery of the HSM file system for FSE partition `fse_fs01`:

```
fserecover --hsmfs fse_fs01
```

See also:

[fsecheck](#), [fsepartition](#), [Introduction to the FSE system \(fseintro\)](#)

fserestore

Restore a complete FSE system state from the backup image data written on FSE backup media.

Syntax

```
fserestore
fserestore { -h | --help }
```

Description

`fserestore` is designed for restoring an FSE system state. Using this command, it is possible to restore:

- a complete state of the FSE server,
- a complete state of the FSE server and all FSE clients (if they were included at backup time),
- a state of a single FSE client.

The restore process recreates all FSE databases and system files including the Configuration Database. If the HSM file system was included in the backup, it will also be restored. If the HSM file system was not restored, the FSE administrator can use the `fserrecover` command to recover the FSE file system from File System Catalog.



Caution: Before using the command, the FSE administrator must be familiar with concepts of FSE system operation and must be well aware of the existing data-loss risk owing to improper use of `fserestore`. Use the command with extreme caution as it can destroy all user data on HSM file systems!

Prerequisites

Prior to using `fserestore` for restoring the FSE installation, the following conditions are checked by the FSE administrator:

1. FSE is properly installed and configured. See the *FSE Installation Guide* for details.
2. If the FSE restore is to be performed on a Linux system, `tar` (tape archiving package) is installed.
3. FSE is in restore mode (using the commands `fse --stop` and `fse --startrestore` on FSE server and FSE client systems).

- Media with valid backup data (such as the latest backup image) are available.
- A drive is available for reading the data from backup media. The backup media is loaded/unloaded manually by the administrator.

User interaction

`fserestore` guides an FSE administrator through the whole restore procedure requesting answers to interactive questions and requiring confirmations.

`fserestore` then starts restoring data from the backup media. If multiple media were used for backup, the FSE administrator is prompted to unload one backup medium and load another one when necessary. All backup media are loaded into the same drive during restore. The last (or only) backup media is inserted first—it contains the backup summary. Afterwards, backup media is inserted in the backup sequence: first media, second media, and so on.

Activity logging

`fserestore` records a progress report to the `restore.log` file (see “[File](#)” on page 333).

Options

- `-h, --help`
Print the usage information for this FSE command.

Note: The `fserestore` operation is not controlled by command line options. The command demands interactive user input and confirmation of the restore scope. See “[Description](#)” on page 332 for details on the command operation and its use.

File

- `/var/opt/fse/log/restore.log` (Linux)
`%InstallPath%\var\log\restore.log` (Windows)
Log file with recorded information about activity of the FSE restore process.

See also:

[fse](#), [fsebackup](#), [fserecover](#), [fsebak](#), [Introduction to the FSE system \(fseintro\)](#)

fsesystem

Manage FSE system configuration. Report states of FSE partitions.

Syntax

```
fsesystem {-s | --show } [RevisionNumber | {-H | --history }]  
fsesystem {-m | --modify } ConfigFileName  
  
fsesystem {-t | --status }  
  
fsesystem --enable-dbg SysDbgLevel [SysDbgFile [SysDbgFlags  
    [SysDbgHSMFS ]]]  
fsesystem --disable-dbg  
  
fsesystem {-h | --help }  
fsesystem {-v | --version }
```

Description

This FSE command manages global FSE system configuration and reports the states of the FSE partitions. It is used to perform the following FSE administrator's tasks:

- retrieve current FSE system configuration,
- apply a modified FSE system configuration,
- show the status of each configured FSE partition with the number of jobs running on them,
- show the total number of running administrative jobs,
- enable debugging or change current settings for debugging of FSE processes (for current FSE session),
- disable debugging of FSE processes (for current FSE session).

Each FSE command communicates with the FSE system components through the Management Interface (`fse-mif`). To execute an FSE command, the Management Interface must be running. If the connection to the Management Interface fails, check its status by running the `fse status` command. In case the Management Interface is not running, start it with the `fse start` command.

After successful command execution, `fsesystem` notifies the FSE administrator with a suitable message, which is printed to the standard output stream (stdout). If there was an error during the command execution, an appropriate error message is printed to the standard error stream (stderr).

Status information

Status information that is printed using the `--status` option is the following:

- number of configured FSE partitions,
- total number of currently running administrative jobs on all FSE partitions.

For each configured FSE partition the following information is printed:

- current partition status (disabled, stopped, running, mounted, unknown),
- number of currently running jobs on the partition (excluding administrative jobs),
- fully-qualified domain name (FQDN) of the machine that hosts the partition,
- mount point of the HSM file system that belongs to the partition.

Partition states have the following meaning:

Status	Partition	Partition Manager	HSM FS	Error
disabled	disabled	not running	not-mounted	
stopped	enabled	not running	not-mounted	X
ready	enabled	running	not-mounted	
ready	enabled	running	mounted	X
mounted	enabled	running	mounted	
unknown	*	*	*	X

* not relevant

Note: Situations indicated by rows with **X** in the column Error indicate that an error occurred and the FSE administrator's intervention is necessary to restore normal FSE operation.

Options

- `-s` , `--show`

If the argument *RevisionNumber* is specified, print the appropriate FSE system configuration revision, otherwise print the current configuration revision.

- *RevisionNumber*

A non-negative integer value which represents the configuration revision to print.

- `-H, --history`

Print the FSE system configuration history depth. The configuration history depth is the number of FSE system configuration file revisions stored in the Configuration Database. This option can only be used together with the `--show` option.
- `-m, --modify`

Modify the FSE system configuration with the configuration file *ConfigFileName*. This action checks the validity of the new configuration file, copies it to the Configuration Database as a new configuration revision, and triggers the FSE system reconfiguration.
- `-t, --status`

Show the current status of each configured FSE partition, the number of jobs running on it, and the total number of running administrative jobs. See “[Status information](#)” on page 335 for details on the status information that is printed with this option.
- `--enable-dbg`

Enable debugging or change the current settings for debugging FSE processes according to the *SysDbgLevel* argument and the *SysDbgFile*, *SysDbgFlags*, and *SysDbgHSMFS* arguments, if they are specified. Use of this option overrides the current debug settings in the FSE tracing configuration file (see “[Files](#)” on page 339), but its effect is limited to the period of time until FSE processes are stopped with the `fse` command (current FSE session). After the processes are restarted, the settings in the FSE tracing configuration file apply.

Note: Use of this option has no effect on debugging the currently running Back End Agents. However, it does have an effect on debugging the Back End Agents that are started later in the current FSE session.

■ *SysDbgLevel*

Debug level identifier whose value can be one of the following uppercase letters: L, N, A, or D. The letters have the following meaning:

- L ... Low debug level. Specify that only the most important information should be logged in FSE debug files. This level of logging results in practically no debug overhead and produces small debug files.
- N ... Normal debug level. Specify that information which is most frequently used for debugging should be logged.

- **A** ... API debug level. Specify that in addition to the most frequently used debugging information, all API calls to third-party components (omniORB/CORBA, Firebird SuperServer) of the FSE installation should be logged.
- **D** ... Detailed debug level. Specify that all details of the FSE processes should be logged. This level of logging causes a noticeable debug overhead and produces a lot of debug output. This argument can only be used together with the `--enable-dbg` option.

■ *SysDbgFile*

An alphanumeric string which specifies where the debug information is recorded. This argument can be used in two ways:

- If the “.” character (a dot) is specified, FSE processes will write their debug data into corresponding files with filenames of the form `<ProgramName>_<ProcessID>.log`. Each generated log file corresponds to a single running FSE process.
- If a string other than “.” is specified, it will be considered as a filename for a single FSE debug file, into which all running FSE processes will write their debugging information.

In both cases the log files will be put in the `/var/opt/fse/log/debug` directory. This argument can only be used together with the *SysDbgLevel* argument. If it is not used, each FSE process writes the debug data to its corresponding standard output stream (stdout).

■ *SysDbgFlags*

A string composed of one or more (different) uppercase letters from the set {**I**, **N**, **C**, **T**, **E**, **V**}. This argument specifies in detail which events in running FSE processes are logged and defines the form of the debug output in an FSE debug file. The letters have the following meanings:

- **I** ... Function in & out. Specify that entries and exits from functions should be logged. This option results in large amounts of debug information.
- **N** ... No-header form. Specify that no FSE debug file header is printed.
- **C** ... Compact-header form. Specify that a compact FSE debug file header is printed instead of the full one.
- **T** ... Exception throw. Specify that all events that cause an exception throw should be logged.

- `E` ... Error occurrence. Specify that major errors which occur in the FSE processes should be logged. Note that such errors are normally only logged in the FSE error log file.
- `v` ... Verbose third-party debugging.

This argument can only be used together with the `SysDbgFile` argument. If it is not used, no events are logged—only the FSE debug file header is printed.

■ `SysDbgHSMFS`

HSM file system debug flags, specified by a positive hexadecimal integer number of the form `0xNNNNNNNN`. These flags enable logging of the events in the HSM file system filter. The flags and the corresponding events are the following:

- `0x00000001` ... main HSM file system filter execution flow.
- `0x00000002` ... HSM file system filter's hooking to system functions.
- `0x00000004` ... virtual file system (VFS) method calls.
- `0x00000008` ... low level communication between user applications and HSM file system filter.
- `0x00000010` ... HSM file system events.
- `0x00000020` ... read/write/change operations on file HSM attributes.
- `0x00000040` ... locking.

This argument can only be used together with the `SysDbgFlags` argument. If it is not used, HSM file system filter event logging is not enabled.

■ `--disable-dbg`

Disable debugging of FSE processes. Use of this option overrides the current debug settings in the FSE tracing configuration file (see “Files” on page 339), but its effect is limited to the period of time until FSE processes are stopped with the `fse` command (current FSE session). When the processes are restarted, the settings in the FSE tracing configuration file apply.

Note: Use of this option has no effect on debugging the currently running Back End Agents. However, it does have an effect on debugging the Back End Agents which are started later in the current FSE session.

■ `-h, --help`

Print the usage information for this FSE command.

- `-v, --version`
Print the copyright and version information for this FSE command.

Examples

- Show the current system configuration:
`fsesystem --show`
- Show the fourth revision of the FSE system configuration:
`fsesystem -s 4`
- Show the configuration history depth for FSE system configuration:
`fsesystem --show --history`
- Copy the current (latest) configuration revision of the FSE system configuration to a file `fse_config.fse` in the current directory:
`fsesystem --show > fse_config.fse`
- Apply the modified FSE system configuration from the configuration file `fse_config_new.fse`:
`fsesystem -m fse_config_new.fse`
- Show the status of each configured FSE partition:
`fsesystem --status`

Files

- `/opt/fse/newconfig/fse-system.cfg` (Linux)
`%InstallPath%\newconfig\fse-system.cfg` (Windows)
Template for an FSE system configuration file.
- `/etc/opt/fse/trace.cfg` (Linux)
`%InstallPath%\etc\trace.cfg` (Windows)
FSE tracing configuration file.

See also:

[fse](#), [fsejob](#), [fsepartition](#), [Introduction to the FSE system \(fseintro\)](#)

FSE Configuration Templates



You can check the default templates for the FSE system configuration. The templates are located on:

- Linux: `/opt/fse/newconfig/`
- Windows: `%InstallPath%\newconfig`
(where the default `%InstallPath%` is `Program Files\Hewlett-Packard\FSE`, but you can change this path by selecting Custom Setup during installation)

The FSE templates provided in this section include:

- [Template for FSE Libraries](#), page 342
- [Template for FSE Drives](#), page 343
- [Template for an FSE System Configuration](#), page 344
- [Template for FSE Partitions—General Part](#), page 345
- [Template for FSE Partitions—Migration Part](#), page 346
- [Template for FSE Partitions—Release Part](#), page 347
- [Template for FSE Partitions—Recall and Allocation Part](#), page 348
- [Template for FSE Partitions—Deletion Part](#), page 349
- [Template for an AIT Media Pool](#), page 349
- [Template for an SAIT Media Pool](#), page 350
- [Template for an LTO Media Pool](#), page 351
- [Template for a Disk Media Pool](#), page 352

Template for FSE Libraries

```
[FSE Library]
[Version = 1]
[Date = ""]
[User = ""]

#
# Specify a name for the tape libray and a hostname, where
# it resides. Use full host name.
#
Name          = " USER DEFINED NAME "
Host          = " LIBRARY HOST NAME "

#
# Control device is identification of library's (media
# changer's) on the specified system.
#
# Tip: use fse_scsi --scan to list available SCSI devices
#
# Linux: /dev/sgNN
# Windows: id:P:B:T:L
#
ControlDevice = " CONTROL DEVICE NAME "

LibType       = SCSI

# AIT - Sony AIT Family
# LTO - LTO Ultrium
MediaFamily   = AIT

#
# Specify the slots that will be used with this library.
# You can specify ranges of slot numbers and individual
# slot numbers.
#
# Example:
# Slots = (1-15, 20, 22, 24, 26-30)
#
[Slots]
Slots = (1-15, 20)
```


Template for FSE Drives

```
[FSE Drive]
[Version = 1]
[Date = ""]
[User = ""]

#
# Specify a name for the tape drive.
#
Name          = " DRIVE NAME "

#
# Specify the tape library name where the drive is placed
# and the index of the drive in the library (IMPORTANT!)
# Index of the first drive in the library always starts with 1.
#
# TIPS:
#   * use fse_scsi --scan to get the available SCSI devices
#   * use fse_scsi <library device file> --ea to display
#     the element assignment
#
Library       = " FSE LIBRARY NAME "

# AIT - Sony AIT Family
# LTO - LTO Ultrium
MediaFamily   = AIT

DriveIndex    = " DRIVE INDEX NUMBER "

# WORM - Drive has WORM capability
Capabilities   = ()

#
# Specify the hostname and the control device
# file of the tape drive that will be used
# to communicate with the drive.
#
# Tip: use fse_scsi --scan to list available SCSI devices
#
# Linux: /dev/sgN
# Windows: id:P:B:T:L
#
Host " HOST NAME "
{
```

```
        ControlDevice = " CONTROL DEVICE FILE "  
    }
```

Template for an FSE System Configuration

```
[FSE System]  
[Version = 1]  
[Date = "2002/01/01 01:01:01"]  
[User = "username@host.domain"]  
  
[Allocation]  
#  
# Similar to the parameters in the partition cfg file, but  
# controls the allocation of drives for all partitions together.  
#  
SystemMaxNumDrivesRecall      = 20      # min=0, max=64  
SystemMaxNumDrivesMigration  = 20      # min=0, max=64  
SystemMaxNumDrivesAdmin      = 20      # min=0, max=64  
SystemMaxNumDrivesRecovery   = 20      # min=0, max=64  
SystemMaxNumDrivesMaint      = 20      # min=0, max=64  
  
#  
# Resource Manager tries to allocate resources to  
# all jobs that have priorities in the range of  
# ResourceAllocationThreshold% from the job  
# with the top priority.  
#  
ResourceAllocationThreshold = 50%      # min=0%, max=100%  
  
#  
# The influence of time factor on job allocation.  
#  
TimeStep      = 2                      # min=0, max=10000  
  
#  
# Influence of retries on the job priorities.  
#  
PhaseFactor = 500                      # min=0, max=100000  
  
[Jobs]  
#  
# Initial init/format job priority.  
#  
SystemAdminPriority = 1000      # min=0, max=10000  
SystemBackupPriority = 1000     # min=0, max=10000
```

Template for FSE Partitions—General Part

```
[FSE Partition]
[Version = 1]
[Date = ""]
[User = ""]

#
# User defined name for an FSE partition. FSE partition
# consists of:
#   * HSM filesystem
#   * media pool(s)
#
Name = " PARTITION NAME "

#
# Where is the HSM file system?
#
Client " HSM FS HOST NAME "

    FileSystemID = _HSM_FS_MAJOR_:_HSM_FS_MINOR_
# e.g. 58:1, Linux client
# FileSystemID = "Volume{xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx}" # Windows client

    Type = Regular
    # Regular - regular read/write file system
# WORM      - Write-Once-Read-Many file system

    # uncomment and configure if Type = WORM :
    # WORMTimeout = 450s
}
#
# Specify which media pool(s) will be used for storing data to.
#
# NOTE: The number of pools specified defines the number of
# copies to be done.
#
Pools      = (" POOL NAME 1 ", "POOL NAME 2")
```

Template for FSE Partitions—Migration Part

```
[Migration]

#
# The amount of time that a file must inactive to
# become a candidate for migration.
#
# min=1s, max=30m
MinFileAge      = 10s

[RegularMigration]

#
# Parameters that trigger a migration.
#
# Migration is started if:
#
# * no new files are added to migration candidate list in
# MinWaitTime AND
# * there are at least MinNumMigFiles to be migrated AND
# * their size is at least MinMigSize
# OR
# * elapsed time since the first file was added to
#   mig. candidate list is more than MaxWaitTime
# OR
# * there are more than MaxNumMigFiles waiting to be migrated
# OR
# * size of those files exceeds MaxMigSize
#
MinWaitTime     = 20s           # min=1s,   max=1h
MinNumMigFiles  = 1            # min=1,   max=1000
MinMigSize      = 2MB          # min=256b, max=1GB
MaxWaitTime     = 30m          # min=1s,   max=24h
MaxNumMigFiles  = 10000        # min=1,   max=10000
MaxMigSize      = 400MB        # min=1MB,  max=500GB
```

Template for FSE Partitions—Release Part

```
[Release]
#
# Parameters that control the release of files on HSM file
# system:
#
# Release starts:
#
# * on HighWaterMark for files that expire retention and are
#   larger than MinFileSize
#
# * on CriticalWaterMark for all already migrated files
#
# * ReleaseInterval means how often the release conditions are
#   checked.
#
CriticalWaterMark   = 95%    # min=80%, max=100%
HighWaterMark       = 85%    # min=10%, max=95%
LowWaterMark        = 35%    # min=0%,  max=90%
ReleaseInterval     = 30s    # min=1s,  max=24h

[ReleaseFiles]
#
# MigRetentionTime: How long should the file not be considered a
# candidate for release after it has been migrated.
#
# RecallRetentionTime: How long should the file not be
# considered a candidate for release after it is recalled from
# the tape.
#
# MinFileSize: Files smaller than this will not be released.
#
# ExcludeFromRelease: Filename patterns of those files that
# should stay on the HSM file system (i.e. should not be
# released) Filename patterns are composed of regular characters
# which may comprise a filename as well as special pattern-
# matching characters.
#
#   ?      Any.
#           Matches any single character.
#
#   *      Closure.
#           Matches zero or more occurrences of any character.
#
#   [abc]  Set.
#           Matches a single character in the set.
#           If the dash '-' character is to be included, it
#           must immediately follow the opening bracket '['.
```

```
#           If the closing bracket ']' character is to be
#           included, it must be preceded by a quote ``
#           [a-z] Range.
#           Matches a single character in the range 'a' to 'z'
#           Ranges and sets may be combined within the same
#           set of brackets (e.g. [a-c123] matches any
#           character in the set {a, b, c, 1, 2, 3})
#           \   Quote (linux).
#           `   Quote (Windows).
#           ^   Makes the next character a regular (nonspecial)
#           character. Note that to match the quote character
#           itself, it must be quoted.
#
#           NOTE: Filename pattern must be enclosed with quotation
#           marks (") in case it contains Set, Range or Quote.
#
#           Example:
#           ExcludeFromRelease = ( *.ini, ".*[xy]??" )
#
MigRetentionTime      = 2h      # min=1s, max=2^32s
RecallRetentionTime   = 24h     # min=1s, max=2^32s
MinFileSize           = 2kB     # min=0,  max=1MB
#ExcludeFromRelease   =
```

Template for FSE Partitions—Recall and Allocation Part

```
[Recall]
#
# Recall will be cancelled if it can't be performed in
# RecallTimeOut.
#
RecallTimeOut         = 15m     # min=1s, max=8h

[Allocation]
#
# MaxNumDrives can be used by this FSE partition at the same
# time
# MaxNumDrivesMigration can be used for migration at the same
# time
# MaxNumDrivesRecall can be used for recall at the same time
#
MaxNumDrives          = 10     # min=0, max=64
MaxNumDrivesMigration = 8      # min=0, max=64
MaxNumDrivesRecall    = 8      # min=0, max=64
```

```

MaxNumDrivesRecovery = 8 # min=0, max=64
MaxNumDrivesMaint = 8 # min=0, max=64

#
# PartitionPriority: How important is this partition compared
# to others?
#
# MigrationPriority: The priority of the migration jobs (should
# be much smaller than RecallPriority)
#
# RecallPriority: The priority of the recall jobs
#
PartitionPriority = 50 # min=0, max=100
MigrationPriority = 20 # min=0, max=100
RecallPriority = 80 # min=0, max=100
RecoveryPriority = 100 # min=0, max=100
MaintPriority = 80 # min=0, max=100

```

Template for FSE Partitions—Deletion Part

```

[Delete]
#
# FileExpiration age: min=1D max=UNLIMITED default=UNLIMITED
#
# Example:
#     FileExpiration 7D
#     {
#         Path = (/usr/tmp, "\Temp\Temporary Internet Files")
#     }
#
#

```

Template for an AIT Media Pool

```

[FSE Pool]
[Version = 1]
[Date = ""]
[User = ""]

#
# Specify the name of the media pool. Media pools are
# used by FSE partitions.
#
# NOTE: You have to assign media to pools with fsemedium

```

```
# command.
#
Name          = " MEDIA POOL NAME "

MediaFamily    = AIT

# Regular - regular media pool for FSE partition
# WORM      - media pool for AIT WORM media
# Backup    - media pool for backup media
PoolType      = Regular

#
# Change the values of the tape format if the defaults don't
# suit you.
#
# NOTE: Some of the parameters will be used only if the tapes
# are (re)formatted.
#
[DefaultFormat]

BlockSize     = 32KB #min=4KB max=1536KB

NumOfVolumes  = 9#min=1 max=32

# None      - medium will not use system volume
# First     - first volume on medium will be system volume
# Middle    - middle volume on medium will be system volume
# Last      - last volume on medium will be system volume
SysVolLocation = Middle

SizeOfSysVol  = 512MB #min=0B max=10GB
```

Template for an SAIT Media Pool

```
[FSE Pool]
[Version = 1]
[Date = ""]
[User = ""]

#
# Specify the name of the media pool. Media pools are
# used by FSE partitions.
#
# NOTE: You have to assign media to pools with fsemidium
# command.
```



```
#

Name                = " MEDIA POOL NAME "

MediaFamily         = SAIT

# Regular - regular media pool for FSE partition
# Backup  - media pool for backup media
PoolType           = Regular

#
# Change the values of the on-tape format if the defaults don't
# suit you.
#
# NOTE: Some of the parameters will be used only if the tapes
# are (re)formatted.
#
[DefaultFormat]

BlockSize          = 32KB #min=4KB max=1536KB

# Single volume only supported on SAIT. Therefore, no system
# volume is created on the SAIT tape.
NumOfVolumes       = 1
SysVolLocation     = None
SizeOfSysVol       = 0MB
```

Template for an LTO Media Pool

```
[FSE Pool]
[Version = 1]
[Date = ""]
[User = ""]

#
# Specify the name of the media pool. Media pools are
# used by FSE partitions.
#
# NOTE: You have to assign media to pools with fsemedium
# command.
#

Name                = " MEDIA POOL NAME "
```

```
MediaFamily      = LTO

# Regular - regular media pool for FSE partition
# Backup - media pool for backup media
PoolType        = Regular

#
# Change the values of the on-tape format if the defaults don't
# suit you.
#
# NOTE: Some of the parameters will be used only if the tapes
# are (re)formatted.
#
[DefaultFormat]

BlockSize       = 32KB #min=4KB max=1536KB

# Single volume only supported on LTO. Therefore, no system
# volume is created on the LTO tape.
NumOfVolumes    = 1
SysVolLocation  = None
SizeOfSysVol    = 0MB
```

Template for a Disk Media Pool

```
[FSE Pool]
[Version = 1]
[Date = ""]
[User = ""]

#
# Specify the name of the media pool. Media pools are
# used by FSE partitions.
#
# NOTE: You have to assign media to pools with fsemedium
# command.
#

Name            = " MEDIA POOL NAME "

MediaFamily     = DISK
PoolType        = Regular
```

```
#
# Change the values of the on-tape format if the defaults don't
# suit you.
#
# NOTE: Some of the parameters will be used only if the tapes
# are (re)formatted.
#
# Mountpoint should be 254G to store the capacity of 5 x AIT2
#
[DefaultFormat]

BlockSize      = 32KB  #min=4KB max=1536KB
VolumeSize     = 10GB  #min=100MB max=2000GB
```


This glossary defines terms used in this guide or related to this product and is not a comprehensive glossary of computer terms.

Administrative job (admin job)

A process, executed by the Management Interface, which performs formatting and initialization of the FSE media.

Advanced Intelligent Tape (AIT)

See also [FSE medium](#).

A magnetic tape format and a tape drive technology used by the FSE installation. FSE supports AIT-2, AIT-3, AIT-2 WORM, and AIT-3 WORM series of tape media, and AIT-2 and AIT-3 series of tape drives.

Back End Agent (BEA)

An FSE agent which executes migration, recall, maintenance and recovery jobs; it also formats and initializes FSE media. All read and write operations on the FSE media are handled by the Back End Agent. The Back End Agent process name is `fse-bea`.

Backup job

A process which makes a backup copy of the HSM file systems, FSE databases and system files onto FSE media in the FSE backup media pool. The backup job is started by the FSE administrator with the `fsebackup` command.

Capacity disk

A disk with high-capacity storage that stores FSE disk media file systems. Capacity disks can be located on the consolidated FSE implementation or the FSE server, or on a remote system where they are accessed via CIFS or NFS protocol. Optionally, capacity disks can store large FSE disk buffer files.

CIFS

The file sharing protocol used on Windows systems. This is one of the protocols used for accessing files on an HSM file system from a remote machine, particularly a Windows client. A smb server (Samba) must be running on the machine hosting the HSM file system, in order to allow file transfers.

Configuration Database (CfgDB)

A directory in the FSE directory layout which contains the currently used and old revisions of the FSE configuration files for configured FSE resources.

Consistency check of FSC

A process executed for a particular FSE partition that checks the consistency of the corresponding File System Catalog (FSC). It compares the contents of the FSC either with the corresponding HSM file system or with Fast Recovery Information (FRI) for that FSE partition. Any inconsistencies found are reported to the shell output and to the FSE check log.

The FSC consistency check is performed by an appropriate type of maintenance job.

CORBA

CORBA is the acronym for Common Object Request Broker Architecture. This is an architecture and infrastructure that computer applications use to work together over networks.

Critical watermark

A user-defined value for the critical level of disk space usage on an HSM file system. When this value is reached, a forced release process is started for all files on the Hierarchical Storage Manager's release candidate list.

Data Location Catalog (DLC)

See also [File System Catalog \(FSC\)](#).

A part of a File System Catalog. Data Location Catalog contains information about locations of the migrated file data on the FSE media. It maintains a full history of file locations.

Deletion policy

A set of rules that define expiration periods for groups of directories on an HSM file system. Files which are older than the expiration period defined by the corresponding deletion policy can be automatically deleted using the `fsefile` command.

Configuration parameters for the deletion policy are specified in the FSE partition configuration file. Each FSE partition has its own deletion policy.

Dirty file

A file on an HSM file system which has been recently created and not yet migrated, or a file that has changed since the last migration or recall.

Disabled drive

An FSE drive which is temporarily put out of operation and cannot be used by migration, recall, administrative, maintenance and recovery jobs. This is done by changing its status in the Resource Management Database to “disabled”. The status is set by an FSE administrator with the `fse drive` command. The example situation that calls for disabling the drive is when it needs cleaning or servicing.

Disk medium

See also [FSE medium](#).

A disk medium is one file system mounted to a subdirectory of `/var/opt/fse/dm/` on Linux systems and to `%InstallPath%\var\dm` on Windows systems. Disk media emulates tape media and is used for regular archiving of the migrated FSE installation. The advantage of using disk media is in shortened recall of offline files.

Duplication

See [Media duplication](#).

Enabled drive

An FSE drive which is put back in operation by changing its status in the Resource Management Database to “enabled”. This status is set manually using the `fse drive` command. An enabled drive is fully available for FSE system operation.

Erroneous drive

An FSE drive which is temporarily put out of operation and cannot be used by migration, recall, administrative, maintenance and recovery jobs. Its status is automatically set in the Resource Management Database to “erroneous” as soon as the problems in the drive are detected by the Back End Agent. Usually, such problems occur when the drive needs servicing.

Explicit release

Unconditional release, started for a file or set of files on the release candidate list, specified by an FSE user. It is triggered with the `fsefile --release` command and occurs regardless of the parameters in the release policy.

Fast Recovery Information (FRI)

File System Catalog related data, collected during migrations and used for eventual recovery of the FSE installation. Medium volumes that are not yet full have Fast Recovery Information stored on the disk in the `/var/opt/fse/fri` directory on Linux systems, and in `%InstallPath%\var\fri` on Windows systems.

Once medium data volumes are full (filled up with migrated files), Fast Recovery Information is written to the part of the medium volume that was reserved in advance for this kind of data. Typically, a medium is partitioned to have a system volume, which stores redundant copies of Fast Recovery Information from all data volumes on the medium.

File generation

A version in the history of a particular file on an HSM file system. Older (non-latest) file generations are only stored offline, as they have already been migrated to FSE media, whereas latest file generations can be present either online or offline, depending on the time of their last access and the corresponding FSE partition policies.

File System Catalog (FSC)

See also [Name Space Catalog \(NSC\)](#).

A database, which consists of the Data Location Catalog (DLC) and Name Space Catalog (NSC). The Data Location Catalog contains information about location of files on the FSE media (full history). The Name Space Catalog (NSC) contains metadata of files on an HSM file system (the last generation metadata).

Each FSE partition has its own File System Catalog.

File System Catalog journal

A file which contains transaction information for the File System Catalog. It is designed to increase the catalog robustness. FSC journals contain information about already applied and pending catalog transactions.

File System Catalog journals are located in the

`/var/opt/fse/part/<PartitionName>/fsc/journal` directory on Linux systems, and in `%InstallPath%\var\part\<PartitionName>\fsc\journal` on Windows systems.

File system event

A creation or modification of a file on an HSM file system, detected and handled by the FSE software. The following operations are identified as relevant file-system events for FSE: create, rename, move, delete, read of contents, change of contents (write, truncate) and change of file attributes (ownership, permissions, and so on).

File System Event Manager

A daemon/service on the FSE client which receives notification about mount events for the HSM file system and triggers the start-up of the appropriate Hierarchical Storage Manager.

The File System Event Manager process name is `fse-fsevtmgr`.

Forced release

See also [Release](#).

Release of all files on an HSM file system, which are on the release candidate list, regardless of their retention time and file size. Forced release is triggered when the critical watermark of usage is reached on the HSM file system. The process is stopped in the moment when the high watermark on the HSM file system is reached.

Formatting

A process that partitions the FSE medium into medium volumes according to the specifications defined in the FSE media pool configuration. If a medium contains FSE data and you want to format it, you have to use the `--force` option with the `fsemedium` command. During formatting all data on the medium is erased.

FSC recovery, FSC recovery job

See [Recovery, recovery job](#).

FSE administrator

A system administrator who installs, configures, monitors, and maintains an FSE installation.

FSE agent

An FSE process that executes tasks on request by FSE jobs and some FSE commands.

The FSE agents are the Back End Agent and the Library Agent.

FSE backup

Archived copies of FSE databases and system files, stored on FSE media, used for restore of the FSE installation. Backup copies represent a complete state of the FSE system components at a particular moment. The creation of the FSE backup is triggered by the `fsebackup` command.

FSE backup log

A file that records relevant information on events that took place during the FSE backup.

The FSE backup log, named `backup.log`, is located in the `/var/opt/fse/log` directory (on Linux systems) and in `%InstallPath%\var\log` (on Windows systems).

FSE backup media pool

See also [FSE media pool](#).

A special type of FSE media pool, used exclusively for backup and restore of the FSE installation. In contrast to regular FSE media pools, the media cartridges in the FSE backup media pool do not contribute to the secondary storage space available to FSE users.

FSE check log

A file that records all the inconsistencies found by the maintenance job, which performs a File System Catalog consistency check for a specified FSE partition.

FSE check logs, named `check_<PartitionName>`, are located in the `/var/opt/fse/log` directory (on Linux systems) and in `%InstallPath%\var\log` (on Windows systems).

FSE client

See also [FSE external client](#).

A set of FSE daemons/services responsible for managing the HSM file systems on the FSE partitions. These daemons/services are: FSE File System Event Manager, FSE Hierarchical Storage Manager, and FSE Service, with additional FSE Windows Service on Windows systems.

FSE command

An FSE command-line interface command that communicates with FSE system components through the Management Interface. FSE commands are used to configure, monitor, and manage the FSE installation.

FSE command-line interface (FSE CLI)

A set of FSE commands that are available to FSE users.

FSE configuration file

A plain text file used to configure or modify the configuration of an FSE resource. FSE configuration files are stored in the Configuration Database.

The Configuration Database is located in the `/var/opt/fse/cfg/` directory (on Linux systems) or in `%InstallPath%\var\cfg` (on Windows systems).

FSE configuration file template

A pre-designed document for an FSE configuration file, included in the FSE package. It contains default configuration values for configuring an FSE resource. These values should be modified before the FSE administrator applies the template.

FSE configuration file templates are located in the `/opt/fse/newconfig` directory (on Linux systems) or in `%InstallPath%\newconfig` (on Windows systems).

FSE configuration revision

Revision of an FSE configuration file. Each configured FSE resource has its own set of configuration revisions stored in the Configuration Database. This set represents the configuration history of that resource.

FSE daemon/service

An FSE process that runs on the FSE installation to provide basic functionality. FSE daemons/services are: FSE File System Event Manager, FSE Hierarchical Storage Manager, FSE Partition Manager, FSE Management Interface, FSE Resource Manager and FSE Service.

FSE database

A file or set of related files with important information on the current status of the FSE installation, characteristics of the FSE installation, or configuration of the FSE installation. FSE databases are the Resource Management Database, File System Catalogs, Hierarchical Storage Management Databases, and Configuration Database.

FSE debug file

A log file of a single FSE process, which records the execution trace of operations inside the process. FSE debug files are created only if tracing in FSE is enabled (typically on request of FSE support personnel).

FSE debug files are located in the `/var/opt/fse/log/debug` directory (on Linux systems) and in `%InstallPath%\var\log\debug` (on Windows systems).

FSE disk buffer

Disk space used to hold temporary files processed during migration, recall, and reorganization jobs. Temporary files are removed after they are written to the FSE media (migration), after they are recreated on the HSM file system (recall), and after the recycled medium volumes are initialized (reorganization). Usually, a separate logical volume or disk partition is allocated for the FSE disk buffer.

Temporary FSE disk buffer files are managed by the Back End Agent, and are located in the `/var/opt/fse/diskbuf/` directory (on Linux systems) and in `%InstallPath%\var\diskbuf` (on Windows systems).

FSE disk media pool

See also [FSE media pool](#).

An FSE disk media pool is a special type of FSE media pool where the migrated data is stored on disks rather than on tape media. FSE disk media therefore emulate FSE tape media, with the advantage of being much faster for migrations and recalls. Using and managing FSE disk media pools is similar to using and managing FSE tape media pools.

FSE drive

A tape drive inside the FSE library, configured to be used by the FSE installation.

FSE error log

A file that records error messages of the running FSE processes. It stores information on all major errors that occur during FSE system operation. These errors often require an intervention of the FSE administrator. Major errors are, for example, failed initialization of an FSE medium, erroneous FSE drive operation, migration job failure, and so on. The FSE error log also provides usage statistics that can be used for fine tuning of the FSE configuration.

The FSE error log, named `error.log`, is located in the `/var/opt/fse/log` directory (on Linux systems) and in `%InstallPath%\var\log\` (on Windows systems).

FSE event log

A file that records relevant information on events happening in the FSE processes during the operation of the FSE installation.

FSE event log, named `fse.log`, is located in the `/var/opt/fse/log` directory (on Linux systems) and in `%InstallPath%\var\log\` (on Windows systems).

FSE external client

See also [FSE client](#).

An external FSE client can be hosted by either a Linux or a Windows system. External FSE clients are connected to the FSE server through a LAN and host HSM file systems. The client runs only the processes that provide functionality for managing these HSM file systems and communication to the major services running on the FSE server.

The daemons/services running on the client are: FSE File System Event Manager, FSE Hierarchical Storage Manager, and FSE Service, with additional FSE Windows Service on Windows systems.

FSE file split

In some situations, the amount of available space on the currently open FSE medium volume is smaller than the amount of data in the currently migrated file. To use the remaining space on the open volume, FSE splits the migrated file into two or more parts, and stores each part on a separate volume. Such migrated file parts are called FSE file splits.

FSE installation

See also [FSE software](#).

HP StorageWorks File System Extender. Hardware with installed FSE software. FSE hardware includes a tape library with drives, slots, media, and a changer, which is connected to the FSE server host machine via a SCSI interface.

FSE library

A SCSI tape library attached to and configured for use by the FSE installation.

FSE media pool

A common storage for the migrated HSM file system data. It groups FSE media of the same type into a logical unit, which is tracked as a group. FSE media inside the same FSE media pool share such characteristics as number of medium volumes (tape partitions), system volume feature, capacity of medium volumes, and so on. Media pool types are: regular, WORM, and backup. Each FSE partition has at least one regular FSE media pool assigned to it. One media pool cannot be assigned to multiple partitions.

FSE medium

See also [Disk medium](#).

A high-capacity magnetic tape, or a disk medium used by the FSE installation. It is further divided into tape partitions that hold medium volumes. FSE media represent the secondary storage space in an FSE installation.

FSE medium volume

A physical partition on an FSE medium, with its own volume header. There are two types of medium volumes: data volume and system volume. Data volumes store migrated file data, while system volumes store redundant copies of Fast Recovery Information from all data volumes on the medium. Each medium can have only one system volume and one or more data volumes.

FSE package

A package with FSE software that will be installed to an FSE installation.

FSE partition

A configurable entity used by the FSE installation to split the storage and configure it according to user requirements. Each FSE partition is related to a single HSM file system and one or more FSE media pools. It has its own policies, File System Catalog, and Hierarchical Storage Management Database.

FSE process

A process running on an FSE installation that performs FSE-specific tasks.

FSE recovery

A process, executed by the Partition Manager, that recovers the HSM file system or the File System Catalog (FSC) without requiring FSE backup copies. The HSM file system is recreated from the Name Space Catalog (NSC), while the FSC is recreated either from Fast Recovery Information (FRI) or from migrated metadata. FSE recovery can be used as a fallback if restore of FSE is not possible due to missing FSE backup data.

A recovery job is triggered explicitly on an FSE user's request with the `fsererecover` command.

FSE resource

Either an FSE partition, FSE library, FSE drive, FSE media pool, or FSE medium including medium volumes. Each FSE resource is configured using an FSE configuration file and has an entry in the Resource Management Database.

FSE restore

See also [FSE recovery](#).

A process that recreates the FSE system components from the FSE backup, stored on one or more FSE media. The restore is needed after the user data on the HSM file system or FSE system data is damaged or lost due to a disaster. When the restore is completed, the FSE system is returned to a fully operational state.

A restore job is triggered explicitly on an FSE user's request with the `fserestore` command.

FSE server

A set of FSE daemons/services that are responsible for managing the configured FSE resources, accepting and executing user commands, executing all types of FSE jobs, and monitoring and operating the FSE installation. These daemons/services are: FSE Resource Manager, FSE Management Interface, FSE Partition Manager, FSE Library Agent, and FSE Back End Agent.

FSE shared library

A proprietary dynamically loaded library, included in the FSE package, that is used by the FSE processes.

FSE slot

A physical place in the FSE library that holds a single FSE tape medium when it is not loaded in an FSE drive. FSE slots have their entries in the Resource Management Database.

FSE software

See also [FSE installation](#).

The programs and data files that are included in the FSE package. Once installed, they actively control the FSE installation and provide functionality to its users.

FSE system file

A temporary file, created by the FSE software, which contains information about the current status of the FSE installation. FSE system files are Fast Recovery Information, File System Catalog transaction logs, Hierarchical Storage Manager lists, and the FSE disk buffer files.

FSE tape media pool

See [FSE media pool](#).

FSE tool

A command that communicates directly with FSE daemons/services and agents besides Management Interface, designed for low-level modifications and more extensive monitoring and troubleshooting tasks. FSE tools are only intended for use by experienced FSE users (typically on request of FSE support personnel). Note that incorrect use of FSE tools can cause data corruption.

FSE tools are located in the `/opt/fse/sbin/tools` directory (on Linux systems) and in `%InstallPath%\bin` (on Windows systems).

FSE user

A computer system user who uses one or more HSM file systems for managing data.

Full Access Mode (FAM)

One of the two HSM file system filter operational modes.

When HSM file system filter is operating in Full Access Mode for a particular HSM file system, you are allowed to make changes to the data located on the HSM file system, and use all the HSM file system functionality that the FSE system provides. For example, you can create, rename, move or delete directories and files, and change their data or metadata. In this mode, all data movement processes (migration, release, recall, deletion) can be executed—by being either triggered explicitly or started according to the configured FSE partition policies.

See also “[Limited Access Mode \(LAM\)](#)” on page 369.

Good medium

An FSE medium in its fully operation state. This state lasts as long as the medium is in a good condition, that is, no write and read errors are detected while writing data to and reading data from the medium.

Hierarchical Storage Management Database (HSMDB)

A database with migration candidate and release candidate lists for files on the HSM file system. Each FSE partition has its own Hierarchical Storage Management Database.

Hierarchical Storage Management Database journal

A file which contains transaction information for Hierarchical Storage Management Databases. It is designed to increase the database robustness. HSMDB journals contain information about already applied and pending database transactions.

Hierarchical Storage Management Database journals are located in the `/var/opt/fse/part/<PartitionName>/hsm/journal` directory on Linux systems and in `%InstallPath%\var\part\<PartitionName>\hsm\journal` on Windows systems.

Hierarchical Storage Manager (HSM)

A part of the FSE client that handles file system events relevant for the FSE installation and enforces migration, release, and recall policies.

The Hierarchical Storage Manager process name is `fse-hsm`.

Hierarchical Storage Manager list (HSM list)

A list of files, maintained by the Hierarchical Storage Manager, which are candidates for migration or release. HSM lists are: a dirty file list (the list of changed files), a migration candidate list, and a release candidate list.

High watermark

A user-defined value for the high level of disk space usage on an HSM file system. When this value is reached, a regular release process is started for all files that are on the Hierarchical Storage Manager's release candidate list and that have passed their migration or recall retention period, and are larger than the defined `MinFileSize`.

HSM file system (HSM FS)

A file system, controlled by the FSE software. It is used to store file metadata (name, attributes, permissions) and online files of an FSE partition. HSM file systems represent the primary storage space of an FSE installation.

On Linux systems, HSM file systems are based on the Linux Ext3 file system; the FSE specific attributes are stored as Ext3 extended attributes. On Windows systems, HSM file systems are based on the NTFS file system.

HSM file system filter

The function of this module is to intercept relevant file system events for the FSE installation and to report them to the Hierarchical Storage Manager.

On Linux systems, the filter is a kernel module providing the `hsmfs` file system, and is built on top of a Linux native Ext3 file system. On Windows systems, the filter is a kernel driver and intercepts all accesses to the NTFS file system.

Initialization

A process that writes the FSE medium volume headers to all volumes found on a medium and creates their volume entries in the Resource Management Database.

Job

A process, executed by Partition Manager or Management Interface, which executes migration, release, recall, deletion, and administrative, recovery, and maintenance processes. Migration and recall jobs and the release process are triggered by their respective policy, whereas other processes, such as explicit release and deletion, administrative, recovery, and maintenance jobs, are triggered on the FSE user's request with FSE commands.

Library Agent (LA)

An FSE agent that handles requests for moving the FSE media between tape drives and library slots and triggers media barcode rescans.

The Library Agent process name is `fse-la-s`.

Limited Access Mode (LAM)

One of the two HSM file system filter operational modes.

When the HSM file system filter is operating in Limited Access Mode for a particular HSM file system, you are only allowed to browse the directory tree on the HSM file system or read the online data from it. You cannot create, rename, move or delete directories and files, nor change their data or metadata. Offline data is not recalled from FSE media; instead, the application that is accessing the file receives an error.

Limited Access Mode does not need FSE processes to be running on the FSE client. However, if the processes are running, three data movement processes (migration, release, deletion) can be executed—by being either triggered explicitly or started according to the configured FSE partition policies.

See also “[Full Access Mode \(FAM\)](#)” on page 366.

Low watermark

A user-defined value for a level of disk usage on an HSM file system. When this value is reached, a regular release process is stopped.

LTO Ultrium

See also [FSE medium](#).

A Linear Tape-Open (LTO) tape format and tape drive product family from IBM and HP. FSE supports LTO Ultrium 1 and 2 series of tape media, and LTO Ultrium 1 and 2 series of tape drives.

Maintenance job

One of the FSE job types that performs maintenance tasks in FSE, namely, the File System Catalog (FSC) consistency check and recreation of redundant migrated data copies.

Management Interface (MIF)

An FSE daemon/service, responsible for accepting, handling, and executing requests for user actions, issued through the FSE command-line interface. It also starts and manages execution of the administrative jobs.

The Management Interface process name is `fse-mif`.

Media duplication

A process that makes an exact copy of an original FSE tape medium. Duplicates can replace the originals, and are used for vaulting purposes.

Memory-in-Cassette (MIC)

A memory chip integrated into magnetic tapes of the AIT family, designed to improve tape performance. FSE uses Memory-in-Cassette in an AIT medium to store information about the medium, its volumes, current position of the tape, and so on, thus reducing the access time when searching for requested data on the medium.

Migration, migration job

A process, executed by the Partition Manager, that copies the file contents from the HSM file system to the FSE media. When the process is completed, the file entries of the migrated files are moved from the migration candidate list to the release candidate list.

Migration candidate

An online, dirty file that is scheduled for migration.

Migration candidate list

A list of files that were changed and have passed a certain period of time defined by the migration policy, which are waiting for migration. The list is maintained by the Hierarchical Storage Manager.

Migration policy

A set of rules by which the migration jobs for files on an HSM file system are initiated. Configuration parameters for the respective policy are specified in the FSE partition configuration file. Each FSE partition has its own migration policy.

Migration retention time

See also [Retention time](#).

A period of time during which a migrated file is kept online before it is released.

Name Space Catalog (NSC)

See also [File System Catalog \(FSC\)](#).

A part of the File System Catalog. NSC includes metadata of files on an HSM file system, such as directory structure, standard attributes, ownership (on Linux systems), and ACL (Access Control List) and Alternate Data Streams (on Windows systems). This metadata enables recovery of an HSM file system.

Network File System (NFS)

The file-sharing protocol used in UNIX networks. This is one of the protocols that can be used for accessing files on an HSM file system from a remote machine, particularly a Linux client.

Offline file

A file whose contents were migrated to one or more FSE media (according to the number of copies defined), and then removed from the HSM file system. The header and extended attributes of an offline file are left intact and remain on the HSM file system. An offline file becomes online at an FSE user's request, when its contents are recalled from the FSE media.

Offline medium

An FSE medium which is temporarily put out of operation and physically removed from the FSE library.

Online file

A file on an HSM file system that is ready to be used by an application. A file is online immediately after it is stored on the HSM file system, regardless of whether it was newly created or recalled from FSE media. An online file becomes offline when it is released.

Online medium

An FSE medium that is put back in operation by being physically inserted into the FSE library and having its status changed in the Resource Management Database to “online”. This online status is set using the `fsemedium` command.

Partition Manager (PM)

An FSE daemon/service that executes and controls the migration, release, and recall processes for a single FSE partition, according to the policies defined in the FSE partition configuration file.

The Partition Manager process name is `fse-pm`.

Performance disk

A disk with high-performance characteristics that stores HSM file systems. Performance disks must be located on the systems that host FSE clients.

Primary storage space

A logical volume/disk partition used to store the HSM file system.

Recall, recall job

A process, executed by the Partition Manager, that copies the file contents from the FSE media to the HSM file system. Once the process is completed, the offline file is again online. Its file entry is then added to the release candidate list.

Recall policy

A rule by which recall jobs for files on an HSM file system are conducted. Since recall is started on an FSE user's request, the only configurable parameter is one that specifies the maximum time a user should wait for the file to become online. This parameter is defined in the FSE partition configuration file. Each FSE partition has its own recall policy.

Recall retention time

See also [Retention time](#).

A period of time during which a recalled file is kept online before it is released again.

Recovery, recovery job

See [FSE recovery](#).

Recycled medium volume

An FSE medium volume from which all important data was copied to another medium volume during media reorganization. The data on recycled medium volumes can no longer be retrieved, but the storage space on such volumes can be reused for new migrated data.

Redundant data copy recreation

A process in FSE that recreates a copy of a file set that has become unreadable, most probably because the FSE medium to which it was migrated is damaged. The process recreates the damaged data copy using copies of the same data that were migrated to other FSE media pools. Therefore, the main prerequisite to execute this process is a multiple media pool configuration of the corresponding FSE partition.

A redundant data copy recreation process is triggered manually by the FSE administrator.

Regular release

See also [Release](#).

Release of files on an HSM file system that are on the release candidate list and meet the predefined criteria, triggered when the high watermark of usage is reached on the HSM file system. The process is stopped when the low watermark on the HSM file system is reached.

Release

A process that usually follows a migration or recall, executed by the Hierarchical Storage Manager. This process removes the file data from the HSM file system, leaving only the file header. Release is triggered by the Partition Manager according to the defined watermarks. After release, files are offline and their file entries are removed from the release candidate list.

Release candidate

An online file that is scheduled for release and that has either been migrated or was brought online by recall.

Release candidate list

A list of online files that have already been migrated or that have recently been recalled and are waiting to be released. The list is maintained by the Hierarchical Storage Manager.

Release policy

A set of rules by which the release process for files on a release candidate list is initiated. Configuration parameters for the release policy are specified in the FSE partition configuration file. Each FSE partition has its own release policy.

Resource Management Database (RMDB)

A database with records of all configured FSE resources. It contains all relevant resource information, such as configuration parameters, resource characteristics, current status, usage statistics, and relations to other resources.

The Resource Management Database is located in the `/var/opt/fse/rmdb` (on Linux systems) or `%InstallPath%\var\rmdb` directory (on Windows systems).

Resource Manager (RM)

An FSE daemon/service responsible for managing data in the Resource Management Database, allocating and releasing FSE resources and providing resource information to other FSE daemons/services and processes.

The Resource Manager process name is `fse-rm`.

Retention time

A defined period of time for files on the release candidate list during which a file is kept online (after it was migrated or recalled) before it is released. The retention time is reset when any of following changes: file contents, standard attributes, time stamps, permissions, or ownership.

Samba

See also [CIFS](#).

Software that allows a UNIX server to act as a file server to Windows clients. A Samba server must be running on the machine that hosts the HSM file system to allow Samba clients to access the HSM file system.

Secondary storage space

A space on FSE media controlled by the FSE installation. Secondary storage space transparently extends primary storage space, which is the HSM file system. It has to be configured, formatted, and initialized before it can be used.

Service

An FSE daemon/service which launches other FSE daemons/services and agents.

The Service process name is `fse-svc`.

Super Advanced Intelligent Tape (SAIT)

See also [FSE medium](#).

A magnetic tape format and a tape drive technology. FSE supports SAIT-1 series of tape media and SAIT-1 series of tape drives.

Ultrium

See [LTO Ultrium](#).

Unreliable medium

An FSE medium to which further writing is disabled, but from which FSE is still able to read data. Its status is set to “unreliable” in the Resource Management Database automatically as soon as the first write error is detected while the Back End Agent is writing data to the medium. Usually, such errors occur because of an excessive use of the medium.

Unusable medium

An FSE medium which has been disabled for both writing to and reading from. Its status is set in the Resource Management Database to “unusable” automatically as soon as the first read error is detected while the Back End Agent is reading data from the medium. Usually, such errors occur because of an extremely excessive use of the medium.

Verified drive

An FSE drive that is put back in operation after its problems have been resolved. This is done by changing its status in the Resource Management Database to “online”. The online status is set manually using the `fsedrive` command.

WORM file system

(WORM: Write-Once Read-Many)

A file system on which a file can be written only once, but can be read many times. A WORM file system prevents both files and their metadata from being modified, regardless of their access permissions. Consequently, it is not possible to remove files from directories or rename them; it is only possible to add new files (as on a regular file system). Files and directory operations are restricted to read-only after a period of time. A WORM partition must be related to WORM media pools containing WORM media only.

WORM medium

On an AIT WORM medium, data can be recorded only once. Once recorded, the AIT WORM medium cannot be overwritten, reformatted, or reinitialized, but new data can be appended to the existing contents. An AIT WORM medium can only be initialized in an AIT WORM media pool.

A

access mode, HSM 154
agents 21
architecture, FSE 18
authorized resellers, HP 13

B

backup 195
 FSE system components 192
 fsebackup command 200
 media management 197
 sample backup
 Linux 201
 Windows 204
 strategy, defining 192
 types 194

C

capacity disk 79
client-server system 24
command reference 251
commands 38
 list 40
 reference 251
 retrieving detailed information 178
 to configure resources and policies 41
communication
 interprocess, configuring 62
configuration
 file elements 36
 FSE database 37
 general procedure 43

 templates 37
Configuration Database 37
configuring 33 to 68
 basics 35
 drives 44
 interprocess communication 62
 introduction 33
 libraries 44
 media pools 71
 partitions 55
 policies, using FSE commands 41
 resources, using FSE commands 41
 WORM partitions 58
consolidated implementation 24

D

daemons 21
debugging 232
 backup and restore jobs 237
 options 232
 problems 237
deletion 129
 examples of policy 131
 policy parameters 130
 recalling deleted files 133
 starting job 133
detail option
 examples 179
 retrieving detailed information 178
disk media 79
distributed implementation 24
document
 prerequisites 11

- related documentation [12](#)

- drives

- configuring [44](#)

- duplicating FSE media [85](#)

E

- error codes [240](#)

- exclusion from release [121](#)

- filename patterns [122](#)

- pattern-matching characters [123](#)

F

- FAM

- switching to LAM [154](#)

- Fast Recovery Information

- recreating from medium [106](#)

- File System Catalog (FSC)

- consistency check

- FSE media [160](#)

- HSM file system [157](#)

- recovery [213](#)

- files system access modes, HSM [146](#)

- formatting FSE media [83](#)

H

- help, obtaining [12](#)

- HP

- authorized resellers [13](#)

- contacting support [239](#)

- storage web site [13](#)

- HSM

- file system access modes [146](#)

- switching access mode [154](#)

- HSM file system

- components [18](#)

- consistency check vs. FSC [157](#)

- as disk media [80](#)

- mounting [60](#)

- preparing [52](#)

- recovery [213](#)

- HSM Lists [111](#)

I

- initializing FSE media [84](#)

J

- jobs

- monitoring [155](#)

L

- LAM

- switching to FAM [154](#)

- libraries

- configuring [44](#)

- multiple [29](#)

- log files, FSE

- error log [172](#)

- event log [166](#)

- viewing [166](#)

M

- media pools [69](#) to [76](#)

- checking state [76](#)

- configuring [71](#)

- multiple copying [144](#)

- types [70](#)

- media volumes

- closing [106](#)

- types [78](#)

- media, FSE [76](#) to [107](#)

- adding to media pool [83](#)

- consistency check vs. FSC [160](#)

- disk media [79](#)

- duplicating [85](#)

- formatting [83](#)

- initializing [84](#)

- managing media and media pools [69](#) to [107](#)

- media pools [69](#)

- media types [78](#)

- multiple copying [144](#)

- preparing [59](#)

- removing [107](#)

- reorganizing [88](#)

- status, changing 105
- status, checking 102
- volumes, closing 106
- volumes, types of 78
- migration 114
 - default policy 117
 - examples of triggering 116
 - policy parameters 115
- monitoring 149
 - jobs 155
 - partitions 150
- mounting
 - HSM file systems 60
- multiple copying 144
 - parallel copying 145
 - recreating redundant copies 97
 - sequential copying 145

O

- old files
 - recalling 163
- operations 109
 - overview 30

P

- partition policy 139
- partitions
 - configuring 55
 - managing 150
 - WORM, configuring 58
- performance disk 79
- planning an FSE implementation 33
- policies
 - deletion parameters 130
 - introduction 113
 - migration parameters 115
 - recall parameter 129
 - release parameters 125
- priority calculation 142

R

- recall 128
 - default policy 129
 - old files 163
 - timeout parameter 129
- recovery 212
 - File System Catalog 213
 - HSM file system 213
 - methods 195
- redundant copies
 - creating 144
 - recreating 97
- related documentation 12
- release 117
 - default policy 127
 - examples of triggering 127
 - excluding files 121
 - policy parameters 125
 - watermarks 118
 - critical 118
 - high 120
 - low 120
- reorganizing FSE media 88
 - parameters 92
- resource allocation 134
- Resource Management Database (RMDB) 38
- restore 207
 - methods 195
 - post-restore steps 212
 - procedure 208
 - requirements 208
 - sample restore 210

S

- SCSI identification 44
- services 21
- support
 - checklist 238
 - contacting 238
- switching access mode, HSM 154
- system deployment 24

system policy [136](#)

T

technical support

checklist [238](#)

contacting [238](#)

templates [341](#) to [353](#)

AIT media pool [349](#)

disk media pool [352](#)

drive configuration [343](#)

introduction [37](#)

library configuration [342](#)

LTO media pool [351](#)

partition configuration

deletion [349](#)

general [345](#)

migration [346](#)

recall and allocation [348](#)

release [347](#)

SAIT media pool [350](#)

system configuration [344](#)

troubleshooting [217](#) to [240](#)

backup and restore [224](#)

communication [226](#)

debugging [232](#)

error codes [240](#)

file system filter [218](#)

file system mounting [219](#)

Firebird Server [228](#)

media management [222](#)

migration [223](#)

recall [223](#)

SCSI problems [229](#)

start-up and shut-down [225](#)

support [238](#)

tools [231](#)

U

user interface [23](#)

W

web sites

HP storage [13](#)

WORM partitions

configuring [58](#)