URL: https://www.plop.at/en/hfsprescue/full.html

HFS+ Rescue - Hfsprescue

Table of Contents

1. Introduction

HFS+ (HFS Plus, Hierarchical File System Plus) is a file system, developed and used by Apple®.

hfsprescue is a program to recover files from a HFS+ formatted partition. You can restore your files and directories even when it's not possible to mount it with your operating system. As side effect, the program also restores deleted files. Your files and directories will be recovered to the directory 'restored/' in the directory where you start hfsprescue. The damaged HFS+ file system will be opened as read only to not change the data on the damaged drive. So you need enough space to copy out the files from the HFS+ file system.

This program has no graphical user interface. You have to run it from the command line.

hfsprescue runs under Linux, Mac OS X and FreeBSD.

hfsprescue supports HFS+ compression (resource fork).

Latest version: 3.5-rc1, 26/Apr/2020

2. Download

HFS+ Rescue recovered your data and you want to say thank you or support the development of this free software? Just donate the amount you think, your data is worth to you :)

Donate EUR ▼ Donate with PayPal or Bitcoin

hfsprescue-3.5-rc1.tar.gz (2020/04/26) Source code
hfsprescue-3.5-rc1-precompiled.tar.gz Precompiled for Linux, Mac OS X, FreeBSD

Outdated, should not be used anymore.

hfsprescue-3.4.tar.gz (2018/02/16) Source code
hfsprescue-3.4-precompiled.tar.gz Precompiled for Linux, Mac OS X, FreeBSD
hfsprescue-3.3.tar.gz (2017/03/31) Source code
hfsprescue-3.3-precompiled.tar.gz Precompiled for Linux, Mac OS X, FreeBSD
hfsprescue-3.2.tar.gz (2016/11/29) Source code
hfsprescue-3.2-precompiled.tar.gz Precompiled for Linux, Mac OS X, FreeBSD
hfsprescue-3.1.tar.gz (2016/09/14) Source code
hfsprescue-3.1-precompiled.tar.gz Precompiled for Linux, Mac OS X, FreeBSD
hfsprescue-3.0.tar.gz (2016/07/26) Source code
hfsprescue-3.0-precompiled.tar.gz Precompiled for Linux, Mac OS X, FreeBSD

hfsprescue-2.2.tar.gz (2015/12/19) Source code
hfsprescue-2.2-precompiled.tar.gz (2015/12/19) Precompiled for Linux, Mac OS X, FreeBSD
hfsprescue-2.1.tar.gz (2015/11/19) Source code
hfsprescue-2.1-precompiled.tar.gz (2015/11/19) Precompiled for Linux, Mac OS X, FreeBSD
hfsprescue-2.0.tar.gz (2015/09/01) Source code
hfsprescue-2.0-precompiled.zip (2015/09/01) Precompiled for Linux, Mac OS X, FreeBSD

hfsprescue-1.1.tar.gz (2015/02/02)
hfsprescue-1.0.tar.gz (2015/01/12)

hfsprescue-0.3.tar.gz (2013/01/30)
hfsprescue-0.2.tar.gz (2011/11/25)
hfsprescue-0.1-patched.tar.gz (2011/10/05)
hfsprescue-0.1.tar.gz (2010/11/30)

3. Recover your files

You have to complete 6 steps to restore your files:

1. Scan for your files.
   hfsprescue -s1 <device node|image file> [-b <block size>] [-o <offset in bytes>] [-d <working / destination directory>] [-f|--force]

2. Cleanup file database.
   hfsprescue -s2 [--utf8len <value 1 to 5>] [--future-days <days>] [-d <working directory>]

3. Restore your files.
   hfsprescue -s3 <device node|image file> [-b <block size>] [-o <offset in bytes>] [-d <working directory>] [--vh-file <file name>] [--eof-file <file name>] [-c <file number>] [--alternative]

4. Restore your directory structure.
   hfsprescue -s4 [-d <working directory>]

5. Move the restored files to the correct directories.
   hfsprescue -s5 [-d <working directory>]

6. Last step, finalize and cleanup.
   hfsprescue -s6 [-d <working directory>] [-k]

hfsprescue will guide you through every step and is telling you the command for the next step.

A simple example for the 6 steps:

```
hfsprescue -s1 /dev/sdb2
hfsprescue -s2
hfsprescue -s3 /dev/sdb2
hfsprescue -s4
hfsprescue -s5
hfsprescue -s6
```

4. Additional features

hfsprescue has some additional features to help you

- Search Unicode string (useful on damaged/lost partition table).
- Search bytes from a file (useful on damaged/lost partition table).
- List files that have been found.
- CSV export of the list of files that have been found.
- Recover one file instead of all files.
- Recover files from a list.
- Find possible positions of the Extents Overflow File.
- Extract the Extents Overflow File.
- Find HFS+ Volume Header and partition start.
- Find HFS+ Alternate Volume Header.
- Extract a HFS+ Volume Header.
- Remove empty directories.

5. Command line parameters

```
hfsprescue [-h|--help] [--version]
```

```
hfsprescue -s1 <device node|image file>
                  [-b <block size>]
                  [-o <offset in bytes>]
                  [-d <working / destination directory>]
                  [-f|--force]
```

```
hfsprescue -s2 [--utf8len <value 1 to 5>] [--future-days <days>]
```

```
hfsprescue -s3 <device node|image file>
                  [-b <block size>]
                  [-o <offset in bytes>]
                  [-d <working directory>]
                  [--vh-file <file name>]
```

```
                         [--eof-file <file name>]
                         [-c <file number>]
                         [--file-list <file name>]
                         [--file-list-csv <file name>]
                         [--alternative]
                         [--ignore-blocks]
                         [--ignore-file-error]


hfsprescue -s4 [-d <working directory>]


hfsprescue -s5 [-d <working directory>]


hfsprescue -s6 [-d <working directory>] [-k]


hfsprescue --find <device node|image file>
                         [-ff <num bytes> <file1> [file2] [...]]
                         [-fs <string>]
                         [-o <offset in bytes>]


hfsprescue --list [--slash] [-d <working directory>]


hfsprescue --csv <file name> [--slash] [-d <working directory>]


hfsprescue --one-file <device node|image file> <file number>
                         [-b <block size>]
                         [-o <offset in bytes>]
                         [-d <working directory>]
                         [--vh-file <file name>]
                         [--eof-file <file name>]
                         [--alternative]


hfsprescue --find-eof
                         [-b <block size>]
                         [-o <offset in bytes>]
                         [--vh-file <file name>]


hfsprescue --extract-eof <device node|image file>
                         [ [--start-block <number>] < [--last-block <number>] | [--num-blocks <number>] > ]
                         [--eof-file <output file>]
                         [--vh-file <file name>]


hfsprescue --find-vh
                         [-o <offset in bytes>]
                         [--first]
                         [-f|--force]
                         [-v|--verbose]


hfsprescue --find-avh
                         [--first]
                         [-f|--force]
                         [-v|--verbose]
```

```
hfsprescue --extract-vh <device node|image file> <LBA sector>
                        [--vh-file <output file>]
```

```
hfsprescue --remove-empty-dirs
                        [--dir <directory>]
                        [-f|--force]
```

Parameter description

Display help

-h, --help	Display help and exit.
--version	Show program version.

Step 1 '-s1'

Scan for your files.

-s1 <device node|image file>	Run step 1. You have to tell the device node or image file.
-b <block size>	Set the block size in bytes. Useful when the Volume Header has been lost.
-f, --force	Overwrite current log files.
-o <offset>	Set the start offset of the partition in bytes. Useful when the partition table is lost or damaged. See here for a description how to calculate the offset.
-d <working / destination directory>	Use this directory instead of current directory.

Step 2 '-s2'

Cleanup file database.

-s2	Run step 2.
--utf8len <value 1 to 5>	Used for false file name detection. When you don't have file names with asian chars, then you don't have to use this option. The default value 1. Use 2 when you have asian file names. The values 3 to 5 should not be used.
--future-days <days>	Used for false file detection. Files with a creation date in the future are ignored. Default value is 7.
-d <working directory>	Use working directory directory.

Step 3 '-s3'

Restore your files.

-s3 <device node|image file>	Run step 3. You have to tell the device node or image file.
-b <block size>	Set the block size in bytes. Useful when the Volume Header has been lost.
-c <file number>	Continue the file restore and skip the files before <file number>.
-o <offset>	Set the start offset of the partition in bytes. Useful when the partition table is lost or damaged. See here for a description how to calculate the offset.
-d <working directory>	Use working directory directory.
--vh-file <file name>	Volume Header file. See also here

--eof-file <file name>	Extents Overflow File. See also here
--file-list <file name>	Restore files listed in file. See also here
--file-list-csv <file name>	Restore files listed in a CSV file. See also here
--alternative	Find a new name when the file already exists in it's directory. Can happen with older versions or deleted files. See here for details.
--ignore-blocks	Do not skip file restore when more blocks are allocated as needed. This is related for files bigger than 1 GB. Affected files are marked with '_too_many_blocks_skipped_' in the log file of step 3.
--ignore-file-error	Do not stop when there is a file creation error. Affected files are marked with '_file_create_error_' in the log file of step 3.

Step 4 '-s4'

Restore your directory structure.

-s4	Run step 4.
-d <working directory>	Use working directory directory.

Step 5 '-s5'

Move the restored files to the correct directories.

-s5	Run step 5.
-d <working directory>	Use working directory directory.

Step 6 '-s6'

Last step, finalize and cleanup.

-s6	Run step 6.
-d <working directory>	Use working directory directory.
-k	Keep mkdir.sh and hfsprescue_dir_id.tmp files.

Find file bytes and/or an unicode string '--find'

Find data on sectors. See here for details.

--find <device node|image file>	Find data. You have to tell the device node or image file.
-ff <num bytes> <file1> [file2] [...]	Find number of bytes from given files.
-fs <string>	Find a given string. The string will be converted to Unicode.
-o <offset in bytes>	Start search from offset.

List files '--list'

List found files. See here for details.

--list	This parameter lists all files that have been found. You can run this after you completed Step 2.
--slash	Mac OS X allows the char '/' in file names in the GUI. For directory compatibility, '/' is converted to ':'. Use --slash when you want to display the '/' in the file name instead of ':'. Maybe when you search a file name which has '/'.

-d <working directory>	Use working directory directory.

CSV export of file list '--csv'

Export list to a CSV file. See here for details.

--csv	Export the file list to a CSV file. You can run this after you completed Step 2.
--slash	Mac OS X allows the char '/' in file names in the GUI. For directory compatibility, '/' is converted to ':'. Use --slash when you want to export the '/' in the file name instead of ':'.
-d <working directory>	Use working directory directory.

Restore one file '--one-file'

Restore just one file instead of all files that have been found. You can run this after you completed Step 2. See here for details.

--one-file <device node|image file> <file number>	You have to tell the device node or image file and the file number of the requested file. Both parameters are required.
-b <block size>	Set the block size in bytes. Useful when the Volume Header has been lost.
-o <offset>	Set the start offset of the partition in bytes. Useful when the partition table is lost or damaged. See here for a description how to calculate the offset.
-d <working directory>	Use working directory directory.
--vh-file <file name>	Volume Header file. See also here
--eof-file <file name>	Extents Overflow File. See also here
--alternative	Find a new name when the file already exists in it's directory. Can happen with older versions or deleted files. See here for details.

Find the Extents Overflow File '--find-eof'

Scan the device for possible start blocks. See here for details.

--find-eof <device node|image file>	You have to tell the device node or image file.
-b <block size>	Set the block size in bytes. Useful when the Volume Header has been lost.
-o <offset>	Set the start offset of the partition in bytes. Useful when the partition table is lost or damaged. See for a description how to calculate the offset.

Extract the Extents Overflow File '--extract-eof'

See here for details.

--extract-eof <device node|image file>	You have to tell the device node or image file.
--start-block <number>	Start block of the Extents Overflow File. Needs also '--last-block' or '--num-blocks'.
--last-block <number>	Last block of the Extents Overflow File. Needs also '--start-block'.
--num-blocks <number>	Number of blocks of the Extents Overflow File. Needs also '--start-block'.
--eof-file <output file>	Save file as file name.
--vh-file <file name>	Volume Header file name.

Find the HFS+ Volume Header '--find-vh'

Scan the device for possible Volume Headers and shows the start of the partition. See here for details.

--find-vh <device node|image file>	You have to tell the device node or image file.
-o <offset in bytes>	Start search from offset.
--first	Just show the first HFS+ Volume Header and quit.
-f, --force	Show Volume Header even when the lastMountVersion field reports another OS than Mac OS X or Linux.
-v, --verbose	Display detailed information.

Find the HFS+ Alternate Volume Header '--find-vh'

Scan the device for possible Alternate Volume Headers. See here for details.

--find-avh <device node|image file>	You have to tell the device node or image file.
--first	Just show the first HFS+ Volume Header and quit.
-f, --force	Show Volume Header even when the lastMountVersion field reports another OS than Mac OS X or Linux.
-v, --verbose	Display detailed information.

Extract the Volume Header'--extract-vh'

See here for details.

--extract-vh <device node|image file>	You have to tell the device node or image file.
LBA sector	Required input. LBA sector number of the Volume Header.
--vh-file <output file>	Save as file name.

Remove empty directories '--remove-empty-dirs'

See here for details.

--remove-empty-dirs	Without additional parameters, the directory './restored' is used.
--dir <directory>	Use the given directory to scan for and remove empty directories.
-f, --force	Don't ask a question to start.

The '--alternative' parameter

During the restore process, it's possible that files have the same name in the same directory. This happens because of deleted files or older versions of files. The default behavior is, that hfsprescue will keep only the latest version of the file, depending on the file date/time. But sometimes its needed to restore deleted and older versions of a file too. You can do this with the '--alternative' parameter.

With this parameter, hfsprescue creates an alternative name, which is a combination of the original name and the Catalog ID of the file (+ if needed an additional number), when there is already a file in the directory.

Note: Older versions of a file or deleted files may be corrupted.

6. List found files

You can display a list of files that have been found, after you completed Step 2 '-s2'.

List fields: File Number: File Name, File Size, File RAW Time, File Date/Time, File Start Block

Command: hfsprescue --list

Sample output:

```
1: permStore, 66097 bytes, 1438688944, Tue Aug  4 13:49:04 2015, Start block 360458
2: reverseDirectoryStore, 65536 bytes, 1438688940, Tue Aug  4 13:49:00 2015, Start block 48344
3: reverseDirectoryStore.shadow, 3136 bytes, 1435230435, Thu Jun 25 13:07:15 2015, Start block 48375
4: reverseStore.updates, 1 bytes, 1438688946, Tue Aug  4 13:49:06 2015, Start block 557113
5: shutdown_time, 4 bytes, 1438067146, Tue Jul 28 09:05:46 2015, Start block 393216
6: store.db, 118784 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 48294
7: store.updates, 3 bytes, 1438688946, Tue Aug  4 13:49:06 2015, Start block 557112
8: store_generation, 4 bytes, 1435230434, Thu Jun 25 13:07:14 2015, Start block 48361
9: tmp.Cab, 0 bytes, 1435230434, Thu Jun 25 13:07:14 2015, Start block 0
10: tmp.Lion, 0 bytes, 1435230434, Thu Jun 25 13:07:14 2015, Start block 0
11: tmp.SnowLeopard, 0 bytes, 1435230434, Thu Jun 25 13:07:14 2015, Start block 0
12: tmp.spotlight.loc, 9961 bytes, 1438141713, Wed Jul 29 05:48:33 2015, Start block 458757
13: tmp.spotlight.state, 4096 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 622598
14: fseventsd-uuid, 36 bytes, 1438141708, Wed Jul 29 05:48:28 2015, Start block 393217
15: reverseStore.updates, 1 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 622602
16: store.updates, 3 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 622601
17: live.1.shadowIndexGroups, 6 bytes, 1438688945, Tue Aug  4 13:49:05 2015, Start block 163913
18: live.1.shadowIndexHead, 4096 bytes, 1438688946, Tue Aug  4 13:49:06 2015, Start block 589824
19: live.2.directoryStoreFile, 4096 bytes, 1438688943, Tue Aug  4 13:49:03 2015, Start block 524338
20: live.2.directoryStoreFile.shadow, 1088 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 524374
21: live.2.indexArrays, 4096 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 524322
22: live.2.indexCompactDirectory, 1024 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 524321
==== CUT =====
```

Note:

- Files that need the Extents Overflow File are marked with '_F_EOF_'.
- Compressed files are marked with '_F_COMPRESSED_'.

Use '--list' with 'grep' to get the file number when you want to restore a single file.

Command: hfsprescue --list | grep mynotes.txt

Sample output:

```
2023: mynotes.txt, 966097 bytes, 1438688944, Tue Aug  4 13:49:04 2015, Start block 560458
```

The file number is 2023.

7. CSV export of file list

You can export a list of files that have been found to a CSV file, after you completed Step 2 '-s2'. The fields are separated by semicolon ';'.

Fields: Number, File Name, Parent ID, Catalog ID, File Size, File RAW Time, File Time, Start block, HFS+ Compressed, EOF (ExtentsOverflowFile)

Command: hfsprescue --csv files.csv

Sample file:

```
"Number";"File Name";"Parent ID";"Catalog ID";"File Size";"File RAW Time";"File Time";"Start block";"HFS+ Compressed";"EOF"
1;"permStore";26;106;66097;1438688944;"Tue Aug  4 13:49:04 2015";360458;"No";"No"
2;"reverseDirectoryStore";26;62;65536;1438688940;"Tue Aug  4 13:49:00 2015";48344;"No";"No"
3;"reverseDirectoryStore.shadow";26;78;3136;1435230435;"Thu Jun 25 13:07:15 2015";48375;"No";"No"
4;"reverseStore.updates";26;96;1;1438688946;"Tue Aug  4 13:49:06 2015";557113;"No";"No"
5;"shutdown_time";26;108;4;1438067146;"Tue Jul 28 09:05:46 2015";393216;"No";"No"
6;"store.db";26;60;118784;1438688947;"Tue Aug  4 13:49:07 2015";48294;"No";"No"
7;"store.updates";26;95;3;1438688946;"Tue Aug  4 13:49:06 2015";557112;"No";"No"
8;"store_generation";26;65;4;1435230434;"Thu Jun 25 13:07:14 2015";48361;"No";"No"
9;"tmp.Cab";26;33;0;1435230434;"Thu Jun 25 13:07:14 2015";0;"No";"No"
10;"tmp.Lion";26;31;0;1435230434;"Thu Jun 25 13:07:14 2015";0;"No";"No"
11;"tmp.SnowLeopard";26;30;0;1435230434;"Thu Jun 25 13:07:14 2015";0;"No";"No"
```

```
12;"tmp.spotlight.loc";26;98;9961;1438141713;"Wed Jul 29 05:48:33 2015";458757;"No";"No"
13;"tmp.spotlight.state";26;63;4096;1438688947;"Tue Aug  4 13:49:07 2015";622598;"No";"No"
14;"fseventsd-uuid";28;29;36;1438141708;"Wed Jul 29 05:48:28 2015";393217;"No";"No"
15;"reverseStore.updates";26;96;1;1438688947;"Tue Aug  4 13:49:07 2015";622602;"No";"No"
16;"store.updates";26;95;3;1438688947;"Tue Aug  4 13:49:07 2015";622601;"No";"No"
17;"live.1.shadowIndexGroups";26;132;6;1438688945;"Tue Aug  4 13:49:05 2015";163913;"No";"No"
18;"live.1.shadowIndexHead";26;221;4096;1438688946;"Tue Aug  4 13:49:06 2015";589824;"No";"No"
19;"live.2.directoryStoreFile";26;194;4096;1438688943;"Tue Aug  4 13:49:03 2015";524338;"No";"No"
==== CUT =====
```

8. Restore one file

You can restore just one file instead of all files that have been found. This feature can be used after you completed Step 2 '-s2'.

hfsprescue --one-file <device node|image file> <file number> [-b <block size>] [-o <offset in bytes>] [-d <working directory>] [--vh-file <file name>] [--eof-file <file name>] [--alternative]

You need the file number to restore a file. You get this number with using 'hfsprescue --list' or from the CSV export file.

Example to restore a file called 'mynotes.txt':

Command 1: hfsprescue --list | grep mynotes.txt

Sample output:

```
2023: mynotes.txt, 966097 bytes, 1438688944, Tue Aug  4 13:49:04 2015, Start block 560458
```

The file number is 2023.

Command 2: hfsprescue --one-file /dev/sdb2 2023

On success, you find the file 'mynotes.txt' in the 'restored/' directory.

9. Restore files with a list

When you don't want to restore all files in step 3, then you can use a list of files that you want to restore. Use the parameter '--file-list <file name>' for a simple list or '--file-list-csv <file name>' for a list in CSV format.

Simple list: --file-list <file name>

The list file has a simple format. At the beginning of the line is the file number (from --list), then comes a colon. After the colon can be any text. The text after the colon will be ignored.

Sample file with 10 files to restore:

```
1003:
283:
553:
18: live.1.shadowIndexHead, 4096 bytes, 1438688946, Tue Aug  4 13:49:06 2015, Start block 589824
19: live.2.directoryStoreFile, 4096 bytes, 1438688943, Tue Aug  4 13:49:03 2015, Start block 524338
20: live.2.directoryStoreFile.shadow, 1088 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 524374
21: live.2.indexArrays, 4096 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 524322
22: live.2.indexCompactDirectory, 1024 bytes, 1438688947, Tue Aug  4 13:49:07 2015, Start block 524321
450:
451:
```

With this function its easy to select files to restore or exclude files from restoring.

As step 4 restores the whole directory structure (even when you restore only a few files), can you use at the end after '-s6' the command 'hfsprescue --remove-empty-dirs' to remove the empty directories.

Example: Exclude files

This example shows how to exclude files with the extensions '.download' and '.part' from restoring.

```
hfsprescue --list | grep -v ".download, " | grep -v ".part, " > files.list
```

• 'grep' is used with '-v' to exclude the pattern from the output.
• As only files with the expected extensions should be excluded, the text ', ' is added at the end, because '--list' writes ', ' at the end of the file name.
• '>' creates a new file with the output of 'grep'.

The command of step 3: hfsprescue -s3 /dev/sdb1 --file-list files.list

Example: Restore files with modification dates

This example creates a list with files that have a modification date of 4., 5. and 12. August 2015.

```
hfsprescue --list | grep " Aug   4 " | grep " 2015, " > files.list
hfsprescue --list | grep " Aug   5 " | grep " 2015, " >> files.list
hfsprescue --list | grep " Aug  12 " | grep " 2015, " >> files.list
```

• 'grep' is used get the text with the pattern.
• Two 'grep' commands are used, because the year is interrupted from the month and day by the time.
• The first command with '>' creates a new file with the output of 'grep'.
• The other commands append with '>>' the output of 'grep' to the output file.

The command of step 3: hfsprescue -s3 /dev/sdb1 --file-list files.list

Example: Exclude files with 0 bytes

This example creates a list with files that are bigger than 0 bytes.

```
hfsprescue --list | grep -v ", 0 bytes, " > files.list
```

• 'grep' is used with '-v' to exclude the pattern from the output.
• '>' creates a new file with the output of 'grep'.

The command of step 3: hfsprescue -s3 /dev/sdb1 --file-list files.list

CSV: --file-list-csv <file name>

To accomplish a more complex filter for files, use some kind of Excel/Spreadsheet software.

Export the file list with '--csv'. Import the file in your spreadsheet program and do the modfications. Then export/save the data to a new CSV file. The delimiter must be a semicolon. The header line (first line) will be ignored by hfsprescue.

Sample file:

```
"Number";"File Name";"Parent ID";"Catalog ID";"File Size";"File RAW Time";"File Time";"Start block";"HFS+ Compressed";"EOF"
14;"fseventsd-uuid";28;29;36;1438141708;"Wed Jul 29 05:48:28 2015";393217;"No";"No"
15;"reverseStore.updates";26;96;1;1438688947;"Tue Aug  4 13:49:07 2015";622602;"No";"No"
16;"store.updates";26;95;3;1438688947;"Tue Aug  4 13:49:07 2015";622601;"No";"No"
17;"live.1.shadowIndexGroups";26;132;6;1438688945;"Tue Aug  4 13:49:05 2015";163913;"No";"No"
18;"live.1.shadowIndexHead";26;221;4096;1438688946;"Tue Aug  4 13:49:06 2015";589824;"No";"No"
19;"live.2.directoryStoreFile";26;194;4096;1438688943;"Tue Aug  4 13:49:03 2015";524338;"No";"No"
```

10. Find the HFS+ Volume Header or Alternate Volume Header and the partition start offset

The Volume Header and Alternate Volume Header have important information about the HFS+ file system. Both can be used to find the start of a partition.

Parameter: --find-vh / --find-avh

The Volume Header

The location of the Volume Header is 1024 bytes after the partition start. Finding the Volume Header is a good way to find the partition start. The Volume Header has also important informations like the location and size of the Extents Overflow File.

When you lost the partition table or you are working with a hard disk image then it's required to know the start offset of the partition with your HFS+ file system. One solution to get the partition start offset is to find the HFS+ Volume Header with the '--find-vh' parameter. A HFS+ file system has a backup of the Volume Header (the Alternate Volume header). This means that hfsprescue will report a few hits. When you had only one HFS+ partition on the drive, then the first hit should be the right one. You can use the '--first' parameter to limit the output to the first detected HFS+ Volume Header.

A valid HFS+ Volume Header criteria of hfsprescue is the 'lastMountedVersion' field. This field is set by the operating system that mounts the partition. Only Mac OS X and Linux is seen as valid by hfsprescue. When the last mount of the partition has been done by another OS, then use '-f' to ignore the OS limitation. If you need more details during the search process then use the '-v' verbose parameter.

When you use '-o <offset in bytes>' then the search begins at the offset and skips the bytes before.

The result of the search will be logged in the file 'hfsprescue-data/find-vh.log'.

hfsprescue --find-vh [-o <offset in bytes>] [--first] [-f|--force] [-v|--verbose]

Example: Show only the first HFS+ Volume Header.

Command: hfsprescue --find-vh /dev/sdb --first

Sample output:

```
*** Stop searching after the first Volume Header has been found.

Scanned 200 MB.
============================================
A Volume Header has been found.

Partition start:     209735680 (Byte), 0xc805000, 409640 (LBA Sector), at 200 MB
Volume Header start: 209736704 (Byte), 0xc805400, 409642 (LBA Sector), at 200 MB

Signature:                      0x2b48, H+
LastMountedVersion:             H+Lx, last mount by Linux.
FileCount:                      4362
DirCount:                       144
BlockSize:                      4096
TotalBlocks:                    244190208
AllocationFile StartBlock:      1
ExtentsOverflowFile StartBlock: 7454
CatalogFile StartBlock:         10270
```

The partition start offset is '209735680'.

Use this value for the '-o <offset in bytes>' parameter in combination with '-s3' and '--one-file'.

Example: hfsprescue -s3 /dev/sdb -o 209735680'

You can extract the Volume Header. See here.

Example: hfsprescue --extract-vh /dev/sdb 409642

The Alternate Volume Header

The Alternate Volume Header is a backup of the Volume Header and is stored 1024 bytes before the end of the partition. When you know the location of the Alternate Volume Header, the block size and total blocks, then its possible to calculate the partition start.

The search for the Alternate Volume Header goes from the end of the hard disk to the beginning.

The resultof the search is stored in 'hfsprescue-data/find-avh.log'.

hfsprescue --find-avh [--first] [-f|--force] [-v|--verbose]

Example: Stop after the first HFS+ Alternate Volume Header.

Command: hfsprescue --find-avh /dev/sdb --first

Sample output:

```
*** Stop searching after the first Volume Header has been found.

Searching backwards for the Alternate Volume Header.

============================================
A Volume Header has been found.

Volume Header start: 1000412826624 (Byte), 0xe8ed404c00, 1953931302 (LBA Sector), at 954068 MB

Signature:                      0x2b48, H+
LastMountedVersion:             10.0, last mount by Mac OS X.
FileCount:                      0
DirCount:                       0
BlockSize:                      4096
TotalBlocks:                    244190208
AllocationFile StartBlock:      1
ExtentsOverflowFile StartBlock: 7454
CatalogFile StartBlock:         10270

Possible partition start: 209735680 (Byte), 0xc805000, 409640 (LBA Sector), at 200 MB
```

The possible start is at offset '209735680'.

Use this value for the '-o <offset in bytes>' parameter in combination with '-s3' and '--one-file'.

Example: hfsprescue -s3 /dev/sdb -o 209735680'

You can extract the Volume Header. See here.

Example: hfsprescue --extract-vh /dev/sdb 1953931302

11. Extract the HFS+ Volume Header

You can extract the Volume Header from the drive to a file with the '--extract-vh' parameter.

hfsprescue --extract-vh <device node|image file> <LBA sector> [--vh-file <output file>]

Beside the device node or image file is the LBA sector number required to extract the Volume Header. When you don't know the LBA sector number, then it's possible to find the Volume Header. See Find the HFS+ Volume Header or Alternate Volume Header and the partition start offset.

You can use the Volume Header file in program modes like '-s1', '-s3', '--one-file' and others.

The default output file is './restored/VolumeHeader'. You can specify another output file name with '--vh-file'.

Sample command: hfsprescue --extract-vh /dev/sdb 409642

12. Find the Extents Overflow File

The Extents Overflow File is used to store the positions of file fragments when a file is split into more that 8 fragments on the file system. When the Volume Header is destroyed or has a wrong entry for the Extents Overflow File, then its not possible to restore strong fragmented files. You can use '--find-eof' to search for possible positions of the Extents Overflow

File. hfsprescue will report possible start blocks. I found no way to reduce the possible positions and exactly identify the Extents Overflow File. It should be one of the first 7 values.

hfsprescue --find-eof [-b <block size>] [-o <offset in bytes>] [--vh-file <file name>]

Note: When you use the '-o <offset in bytes>' parameter, then the reported start blocks are relative to the offset value.

Note: When the Volume Header is defect and you have a backup, then specify the file name with '--vh-file'.

When you found positions, then you can extract the Extents Overflow File to a file. See Extract the Extents Overflow File.

Example:

Command: hfsprescue --find-eof /dev/sdb2

Sample output:

```
Searching block positions of the Extents Overflow File...

1. Possible block: 217 | File position: 0xd9000
2. Possible block: 487 | File position: 0x1e7000  maybe ExtentsOverflowFile or CatalogFile
3. Possible block: 1023 | File position: 0x3ff000
4. Possible block: 1149 | File position: 0x47d000
5. Possible block: 2108 | File position: 0x83c000
6. Possible block: 3132 | File position: 0xc3c000
7. Possible block: 25660 | File position: 0x643c000
==== CUT =====
```

In this case, the correct start block was 2108 (the 5. entry). I tried all blocks until I got the correct one.

When you find many possible blocks, then you have to test them with restoring a file that is strong fragmented. You have to extract the Extents Overflow File before you restore a file with '--one-file' for testing.

Info: When you have strong fragmented files, then

- those files will be reported in the log file of Step 3. Search for the text '_has_extents_overflows_'.
- are marked with '_F_EOF_' flag when you use '--list'.
- are reported in the CSV export.

13. Extract the Extents Overflow File

The Extents Overflow File is required to restore heavy fragmented files. The Extents Overflow File is extracted automatically to a file by hfsprescue when the Volume Header is valid and you use '-s3' or '--one-file'.

You can extract the Extents Overflow File on demand with '--extract-eof'.

hfsprescue --extract-eof <device node|image file> [[--start-block <number>] < [--last-block <number>] | [--num-blocks <number>] >] [--eof-file <output file>] [--vh-file <file name>]

The default output file is './restored/ExtentsOverflowFile'. You can set another file name with '--eof-file <output file>'.

Note: When you formatted the partition, then the Extents Overflow File is lost in the most cases.

Valid Volume Header

When your Volume Header is valid, then '--extract-eof' does not need any additional parameters to extract the Extents Overflow File.

Example: 'hfsprescue --extract-eof /dev/sdb1'

```
Signature:                      0x2b48, H+
LastMountedVersion:             H+Lx, last mount by Linux.
FileCount:                      4362
DirCount:                       144
BlockSize:                      4096
```

```
TotalBlocks:                    244190208
AllocationFile StartBlock:      1
ExtentsOverflowFile StartBlock: 7454
CatalogFile StartBlock:         10270
Total size:                     931 GB

Extracting the ExtentsOverflowFile to 'restored/ExtentsOverflowFile'.

Size: 11534336 bytes, 11.00 MB
Clump Size: 11534336 bytes
Total Blocks: 2816
Extent 0: Start 7454, Num 2816
Extent 1: Start 0, Num 0
Extent 2: Start 0, Num 0
Extent 3: Start 0, Num 0
Extent 4: Start 0, Num 0
Extent 5: Start 0, Num 0
Extent 6: Start 0, Num 0
Extent 7: Start 0, Num 0

File created.
```

Extract with Volume Header problems

When your Volume Header is defect, then you can use '--find-eof' to locate possible positions of the Extents Overflow File. You can use those possible positions and try to extract the Extents Overflow File. When you extract with '--start-block' and '--last-block' / '--num-blocks' then only the given range will be extracted. When the Extents Overflow File itself has fragments, then its not possible to restore the whole file without a Volume Header. You can use a Volume Header file with '--vh-file <file name>' when the partition Volume Header is defect.

A simple example:

• At first, search for possible locations.

hfsprescue --find-eof /dev/sdb -b 4096

```
*** Force block size: 4096
Signature:                      0x00,  (Unknown)
LastMountedVersion:             , last mount was not done by Mac OS X.
FileCount:                      0
DirCount:                       0
BlockSize:                      4096
TotalBlocks:                    0
AllocationFile StartBlock:      0
ExtentsOverflowFile StartBlock: 0
CatalogFile StartBlock:         0
Total size:                     931 GB

Searching block positions of the Extents Overflow File...

1. Possible block: 7710 | File position: 0x1e1e000
2. Possible block: 10526 | File position: 0x291e000    maybe ExtentsOverflowFile or CatalogFile
```

• Then extract the Extents Overflow File. We use as start block the first entry and as last block the second entry value -1.

hfsprescue --extract-eof /dev/sdb -b 4096 --start-block 7710 --last-block 10525

```
*** Force block size: 4096
Signature:                      0x00,  (Unknown)
LastMountedVersion:             , last mount was not done by Mac OS X.
FileCount:                      0
DirCount:                       0
BlockSize:                      4096
TotalBlocks:                    0
AllocationFile StartBlock:      0
ExtentsOverflowFile StartBlock: 0
CatalogFile StartBlock:         0
Total size:                     931 GB

Extracting the ExtentsOverflowFile to 'restored/ExtentsOverflowFile'.
*** Warning. No extents of the ExtentsOverflowFile will be restored.
Extracting blocks from 7710 to 10525. 2816 blocks.
File created.
```

You have to test the extracted Extents Overflow File with restoring a strong fragmented file.

Use 'hfsprescue --list | grep _F_EOF_' to find an affected file.
Use '--one-file' to restore the file. Then check the restored file.

When you find more possible positions, then you have to test them until you find the correct start/end of the Extents Overflow File.

14. Remove empty directories

The whole directory structure will be recovered in step 4. Even, when you recover only a few files ins step 3. Now, you have a lot of empty directories at the end of step 6 '-s6'. You can remove them with '--remove-empty-dirs'.

hfsprescue --remove-empty-dirs [--dir <directory>] [-f|--force]

The standard start directory is './restored'. You can set a directory with '--dir'.

Before the program removes the empty directores, it shows the full start path and ask you to continue. You have to enter 'y' and the program starts. You can skip the question with the parameter '-f' or '--force'.

15. Find bytes from a file and/or an Unicode string

An additional debug/search feature is to find bytes from a file and/or Unicode strings. -ff and -fi can be used with the same command.

hfsprescue --find <device node|image file> [-ff <num bytes> <file1> [file2] [...]] [-fs <string>] [-o <offset in bytes>]

- **Find bytes of one or more files -ff**

You can find the bytes of a file. This can be very useful when you lost the partition table and you have to calculate the offset of the partition. It makes no sense to search for the whole file. It can be fragmented and you will never get a hit. The file size of the source file is not limited by hfsprescue. Regardless to the file size is the maximum search buffer size 1MB. This this is really large. I suggest to use the block size as maximum search length.

The result of the search will be logged in the file 'hfsprescue-data/find.log'.

When you use '-o <offset in bytes>' then the search begins at the offset and skips the bytes before.

Find bytes example 1:

Command: hfsprescue --find /dev/sdb2 -ff 4096 PerfectPicture.jpg

Sample output:

```
File PerfectPicture.jpg: Bytes found at offset 746586112 + 28672 = 746614784 (0x2c800000 + 0x7000 = 0x2c807000)
```

Find bytes example 2: Multiple files

Command: hfsprescue --find /dev/sdb2 -ff 4096 PerfectPicture.jpg anyfile.doc

Sample output:

```
File anyfile.doc: Bytes found at offset 0 + 401408 = 401408 (0x0 + 0x62000 = 0x62000)
File PerfectPicture.jpg: Bytes found at offset 746586112 + 28672 = 746614784 (0x2c800000 + 0x7000 = 0x2c807000)
```

- **Find string -fs**

This can be used to find an Unicode string. File names are encoded in Unicode. Its possible to find directory entries with this feature. The search string will be converted to Unicode by hfsprescue.

The result of the search will be logged in the file 'hfsprescue-data/find.log'.

When you use '-o <offset in bytes>' then the search begins at the offset and skips the bytes before.

Find string example:

Command: hfsprescue --find /dev/sdb2 -fs myimportantfile.doc

Sample output:

```
String "myimportantfile.doc" found at offset 1048576 + 326870 = 1375446 (0x100000 + 0x4fcd6 = 0x14fcd6)
String "myimportantfile.doc" found at offset 1048576 + 494446 = 1543022 (0x100000 + 0x78b6e = 0x178b6e)
```

16. Files & Logs of hfsprescue

hfsprescue stores it's files in the directory './hfsprescue-data/' or the directory you sett with '-d'.

filesfound.db	Detail infos about recoverable files, duplicate entries not removed.
fileinfo.db	Detail infos about recoverable files, duplicate entries removed.
fileinfo.sha	Helper file to cleanup the file database.
foldertable.db	Detail infos about recoverable directory structure.
s1.log	Log file of Step 1.
s2.log	Log file of Step 2.
s3.log	Log file of Step 3.
s4.log	Log file of Step 4.
s5.log	Log file of Step 5.
onefile.log	Log file when recovering one file '--one-file'.
find.log	Log file of byte and/or Unicode search.
find-eof.log	Log file of Extents Overflow Files search '--find-eof'.
extract-eof.log	Log file of Extents Overflow Files extract to file '--extract-eof'.
find-vh.log	Log file of Volume Header / partition search '--find-vh'.
find-avh.log	Log file of Alternate Volume Header search '--find-avh'.

Note: Logs are appended when you restart an action. Its uppon the user to truncate/remove the log files.

17. The 'restored' directory

The 'restored' directory will be created in the directory where you start hfsprescue or you set with the '-d' parameter. Your files and directories will be recoverd to this directory.

When you use '--one-file', then the restored file will be written directly to the 'recovered' directory without any path.

After step 6 has been completed, there are also a few additional files and directories, created by hfsprescue.

Directory: restored/newroot/recovered/

You find your recovered files here.

Directory: restored/newroot/x_directory_problem/

It was not possible to link the files and directories to their parent directory. You will find here files too.

Directory: restored/newroot/x_unknown/

No valid parent directory has been found. Maybe you will find here already deleted files. The files in this directories can be invalid.

Files in restored/newroot/

INFO.TXT

Just an info file for you.

recovered-files.txt, x_directory_problem-files.txt, x_unknown-files.txt

The contents of those files is just a list of the files in the corresponding directory. Useful when you need an overview of the restored files.

18. Problems / FAQ

The following articles are about why files are defect recovered and what you can do against it. Mostly its a problem because of a wrong access to the partition data. You will also see how you can find the start of a partition.

Overview:

- Restored files are invalid/defect! Why?
- Partition table damaged, lost or not usable! How do I find the correct partition offset?
- How do I find the correct partition offset with a reference file.
- Unusual block size.
- Extents Overflow File problems.
- Permission denied.
- Precompiled - FATAL: kernel too old.
- sudo: hfsprescue: command not found.
- Problem with file names that have accents.

19. Restored files are invalid/defect! Why?

Reasons for invalid/defect files can be

- you are using a wrong partition start (because of a corrupt partition table or you working with a complete hard disk image, and so on).

- the file has been deleted and overwritten on the damaged file system.

- you are using a wrong block size.

- it was not possible to export the Extents Overflow File.

- the Extents Overflow File is damaged.

The most cases are a wrong partition start. See here how to get the correct partition start offset.

20. Partition table damaged, lost or not usable! How do I find the correct partition offset?

HFS+ Rescue can help you to find the correct partition offset when

- you are working with a real drive and

  - the partition table has been damaged.
  - the partition table has been overwritten with new partition data.
  - the partition table doesn't exists anymore.

OR

- you are working with a complete hard disk image and
  - you are unable to use the program kpartx.
  - the partition table has been damaged.
  - the partition table has been overwritten with new partition data.
  - the partition table doesn't exists anymore.

When do I need the partition start offset?

When you are unable to access the partition then hfsprescue cannot read the block size and the start block of the Extents Overflow File. Also it's not possible to successful restore the files, because they are stored relative to the partition start.

There are 2 ways to find the partition start offset with hfsprescue.

1. When your HFS+ Volume Header is ok, then see Find the HFS+ Volume Header and the partition start offset.

2. When you have a reference file, then see How do I find the correct partition offset with a reference file.

21. How do I find the correct partition offset with a reference file

Before you can start, it's required that you completed the file restore Steps 1 & 2. When you did that some time ago then continue.

You should know the block size of the damaged file system. Usually it's 4096. When the partition is bigger than 2 TB, then the usual block size is 8192.

You need a file that was on the drive. It can be any type of file. Maybe the executable file of a program (get it from another computer) or a picture file (maybe from an old backup). I prefer a picture file. Choose a file that was nearly only once on the drive to reduce false hits.

1) At first check if the file has been found during -s1 and -s2. For this example I am using a picture file called 'PerfectPicture.jpg'.

hfsprescue --list | grep PerfectPicture.jpg

Sample output:

```
77: PerfectPicture.jpg, 892187 bytes, Sun May 10 18:21:18 2015, Start block 131074
```

Great, the 'PerfectPicture.jpg' file has been found. And we have luck, it was only once on the drive ;). When you have a file that has been found more than once, then either choose another file or you have to test various offset values at the end of the process until you have the right one.

2) The next step is to do a byte search for the file on the drive.

hfsprescue --find /dev/sdb -ff 4096 PerfectPicture.jpg

Sample output:

```
File bytes found at offset 746586112 + 28672 = 746614784 (0x2c800000 + 0x7000 = 0x2c807000)
```

Great, the bytes of our file have been found and what a surprise, the bytes have been found also only once. When the bytes have been found more than once, then you have to calculate offset values with the other found offsets too.

3) The formula.

offset = byte_search_result - list_start_block * block_size

With our values it looks like

offset = 746614784 - 131074 * 4096

When you doesn't have a calculator then use the shell for the calculation.

echo $((746614784 - 131074 * 4096))

The result is 209735680

A simple check if it could be a correct result is to modulo the value with 512. echo $((209735680 % 512)) the result must be 0.

4) Check if the calculated offset is correct.

Restore the file with '--one-file' and the file number from '--list' and the calculated offset value.

hfsprescue --one-file /dev/sdb 77 -b 4096 -o 209735680

This command restored the file to restored/PerfectPicture.jpg.

Now compare the reference file with the restored file. diff PerfectPicture.jpg restored/PerfectPicture.jpg

When 'diff' doesn't report a difference, then the offset value for the partition is correct.

Remove the old 'restored/' directory and restart with Step 3 '-s3' and use the calculated offset value with '-o'.

Example: hfsprescue -s3 /dev/sdb -b 4096 -o 209735680

22. Unusual block size

The correct block size is required to restore files. The block size is stored in the HFS+ Volume Header. When this header is corrupt or not available, then its not possible for hfsprescue to determine the correct block size. There can be many reasons for a corrupt header. Either it is really overwritten by some data or you do not access the partition from it's beginning (Common mistake when you work with a complete hard disk image).

Solutions:

> On overwritten/corrupt Volume Header: Use a common block size value and set it with '-b'.
> Example: -b 4096
>
> When you work with a hard disk image: Use kpartx or set the partition start offset with '-o'.
>
> On lost partition table: Set the partition start offset with '-o'.

Usually, the value for the block size is '4096'. When the partition is bigger than 2 TB, then the usual block size is '8192'.

23. Extents Overflow File problems

When you have strong fragmented files, then its required to access the Extents Overflow File which is an area on the HFS+ partition. When this area has been overwritten, then there is no chance to restore files that have information stored in the affected area.

With '--find-eof' you can search for possible start blocks of the Extents Overflow File.

With '--extract-eof' its posbbile to save the Extents Overflow File into a file. This file can be used for the restore process.

24. Permission denied

When you try to access a device, then you need root permissions. Just use 'sudo'.

Example: sudo hfsprescue -s1 /dev/sdb2

or change to the root user with 'su'.

It's also possible that the device is mounted. Unmount the device before you run hfsprescue. For Mac OS X see here.

25. sudo: hfsprescue: command not found

When you start hfsprescue with 'sudo' and the program is not in your PATH, then you have to set a path.

Example: hfsprescue is in the current directory: sudo ./hfsprescue

26. Precompiled - FATAL: kernel too old

The x86 Linux versions are compiled against LibC 2.11. When your Linux distribution is using an older version then you will see the error 'FATAL: kernel too old'. You have to compile the source code on your computer. You need the GCC (GNU Compiler Collection) installed to compile hfsprescue.

27. File name accent troubles

This is related **for other operating systems than Mac OS X**.

The unicode to UTF-8 file name conversion is not fully supported. There are problems with accents on non Mac OS X systems.

A fix: The accents should be corrected when you copy the restored files to a HFS+ file system.

This bug will be fixed maybe in a further version.

28. Asian file names

When you have files with asian file names, then you have to set '--utf8len 2' on step 2.

Example: hfsprescue -s2 --utf8len 2

When you don't set the utf8len, then all files with asian chars will be seen as false file name and filtered.

Note: You don't have to run again -s1 to change the utf8len. Just run again -s2 with the required setting.

29. Mac OS X notes

When you try to restore data from a drive that is automatically mounted when you connect it, then unmount it with 'diskutil'. This situation can happen when you connect a drive with a valid file system. Maybe after an unwanted fresh formatting.

Just figure out the device name with 'mount'.

Sample output:

```
/dev/disk0s2 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)
/dev/disk6s2 on /Volumes/Untitled (hfs, local, nodev, nosuid, journaled, noowners)
```

The affected partition is '/dev/disk6s2 (/Volumes/Untitled)'.

Unmount the partition with 'sudo diskutil umount /dev/disk6s2'.

Now you are able to access the partition with hfsprescue.

Example: sudo hfsprescue -s1 /dev/disk6s2

30. Personal notes

I wrote the first version of this tool for my neighbour. He was unable to mount the HFS+ partition. He got the error 'hfs: failed to load catalog file' and some other errors about the B-Tree. I was able to restore the most files.

Meanwhile, my program became very sophisticated with many features. When there are still problems then just contact me, but don't forget a friendly "hello".