



Paragon Technologie GmbH  
Leo-Wohleb-Straße 8 • 79098 Freiburg, Germany  
Tel. +49-761-59018-201 • Fax +49-761-59018-130  
Website: [www.paragon-software.com](http://www.paragon-software.com)  
E-mail: [sales@paragon.software.com](mailto:sales@paragon.software.com)

---

# Paragon APFS for Linux

---

User manual

---

## Abstract

This document covers implementation of APFS file system support in Linux operating systems using Paragon APFS file system driver. Basic installation procedures are described. Detailed mount options description is given. List of supported APFS features is given with limitations imposed by Linux. There is also advanced troubleshooting section.

## Information

### Copyright© 2019 Paragon Technologie GmbH

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assumes no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: March 2019 in Freiburg, Germany.

### Special thanks to:

All the people who contributed to this document, either by writing text, developing solutions to various issues, performing tests, collecting information or by requested support from our team. To our customers who continue to support us and help us to improve the product by constantly demanding more.

### ***We welcome your feedback***

*Please send your feedback to your Paragon contact or to [sales@paragon-software.com](mailto:sales@paragon-software.com).*

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Paragon UFSD technology . . . . .	4
1.2	How UFSD works on Linux . . . . .	4
1.3	Key features . . . . .	4
<b>2</b>	<b>System requirements</b>	<b>6</b>
2.1	Hardware requirements . . . . .	6
2.2	Software requirements . . . . .	6
2.3	Limitations . . . . .	7
<b>3</b>	<b>Installation</b>	<b>8</b>
3.1	Shipment . . . . .	8
3.2	Components . . . . .	8
3.3	Installing driver . . . . .	8
3.4	Uninstalling driver . . . . .	9
<b>4</b>	<b>Using The Driver</b>	<b>11</b>
4.1	Mounting volumes . . . . .	11
4.2	Mount options . . . . .	11
<b>5</b>	<b>Troubleshooting</b>	<b>13</b>
5.1	Loading UAPFS driver module issue . . . . .	13
5.2	UAPFS driver mount volume issue . . . . .	13

# 1 Introduction

## 1.1 Paragon UFSD technology

UFSD (Universal File System Driver) is a unique technology developed by Paragon Software to provide full access (read/write, format, etc.) to volumes of the popular file systems: NTFS, exFAT, HFS+, APFS under various platforms, including Windows, Linux, Mac OS X, etc. in case these file systems are not otherwise supported.

UFSD technology provides access directly to the physical devices that is why it can process partitions regardless of their support by the current Operating System (OS). With UFSD it is possible to mount NTFS, HFS+, APFS and exFAT partitions under Linux, thus getting access to its contents, just the way it is implemented in the NTFS&HFS+ or APFS for Linux driver, and the technology also allows direct access via physical device addressing, the way it is implemented in the driver too.

Paragon UFSDs are designed to be readily integrated into any solution using our UFSD Software Development Kit (UFSD SDK), which includes all of the necessary tools to develop applications with the following main features:

- Access to un-mounted partitions (i.e. drive letter not assigned);
- Access to other file systems that normally would not be supported by the operating system;
- Platform-independent UFSD API.

This software product contains components, which are partly subject to the license terms of the GNU General Public License or GNU Lesser General Public License ("LGPL"). You can request the modified source code of this software via a contact request: <https://www.paragonsoftware.com/contact.html>. The offer is valid for at least 3 years from the date of the publication of the corresponding software product. We deliver the software on CD/DVD or USB stick, whose production costs we claim in return.

*Note:* NTFS, HFS+, APFS and exFAT drivers for Linux as well as utilities were written using UFSD SDK.

## 1.2 How UFSD works on Linux

Modern operating systems are based on the concept of Installable File System drivers (IFS). User simply needs to provide an operating system with the proper file system driver to work with the file system in usual manner. Paragon APFS for Linux includes APFS driver for Linux environment. Once appropriate components of Paragon APFS for Linux are installed, the operating system can mount these file systems and work with directories/files stored on the file systems.

## 1.3 Key features

**Paragon APFS for Linux has the following main features:**

- Transparent read/write access to all subvolumes in APFS container;
- High performance (in some cases even better than HFS+);
- Easy installation and uninstallation (assistant scripts);
- Support for the latest Linux Kernels and distributions;

- Both case-sensitive and case-insensitive APFS volumes are supported;
- Support reading from encrypted volumes;
- Support for Kernel versions from 2.6.36 up to 4.20.x;

## 2 System requirements

This topic highlights requirements to hardware and software that may be used to run Paragon APFS for Linux.

### 2.1 Hardware requirements

#### Minimum hardware requirements:

- Processor: Intel Pentium 300 MHz and higher, or compatible;
- both 32-bit and 64-bit CPUs are supported;
- 120MB of RAM.

RAM consumption depends first of all on whole amount of memory available in the system. If it is low then the driver wouldn't keep a lot of descriptors opened to keep the memory usage at minimum.

### 2.2 Software requirements

#### Supported Linux Kernel versions

- Linux with kernel versions 2.6.36 and newer;
- Linux with kernel versions up to 4.20.x ( APFS driver was tested with Kernels up to 4.20.10).

#### Linux distributions the products were tested with:

- Ubuntu 18.04;
- Debian 9.8;
- Fedora 29;
- OpenSuse Leap 15.0;
- CentOS 7.6.

#### Development Environment

A development environment is required to compile Linux driver. Please verify that these tools are all functional. The easiest way is to choose the developer toolkit when installing Linux.

What must be installed:

- DKMS library;

```
#dkms --version
```

- Kernel source code (recommended) or Kernel header files (doesn't always work);

```
#rpm -qa|grep kernel-source (for RPM based kernel-sources)
```

- GNU C compiler;

```
#gcc --version
```

- GNU Make;

```
#make --version
```

- GNU ld (binutils);

```
#ld --version
```

- Modutils (module-init tools).

```
#insmod -V
```

## 2.3 Limitations

- GNU C compiler (gcc) version 4.9 or higher is required;
- The user should login as root to install the driver;
- Limited read-only access to APFS volumes containing local snapshot(s);
- Only 1st subvolume can be mounted on 32-bit platforms;
- No support of moving between subvolumes (via 'mv' command);
- Proper working with volumes more than 16TB is not guaranteed;
- APFS file system utilities (mkapfs and chkapfs) are not provided;
- Correct operation is not guaranteed for customized Linux kernels. Commercial porting service to customized Linux kernels is available from Paragon Software Group — for more information send e-mail to [sales@paragon-software.com](mailto:sales@paragon-software.com)).

## 3 Installation

This section describes workflows related to installing and using Paragon APFS for Linux driver.

### 3.1 Shipment

The setup files for each product of the family are provided as the downloadable TGZ archives, which can be downloaded from the company site.

### 3.2 Components

The package includes the following components:

- Source files for the APFS for Linux driver;
- Assistant script files, which are purposed to simplify the installation and uninstallation routines;
- Source files for DKMS library support.

Paragon APFS Linux driver must be compiled on the end user's system for correct configuration. By installing the software you accept the terms of End User License Agreement listed in License file.

### 3.3 Installing driver

First, APFS driver must be built and installed.

**Steps to install the APFS for Linux driver are as follows:**

1. Log in as root. This step is obligatory;
2. Build and install the Paragon APFS for Linux using `install.sh` script. Alternatively, driver binary module may be built manually using `'configure' 'make driver'` commands.
3. Install the APFS driver (this step will make the modules available for use);
4. Activating (loading) the driver. After building and installing, the APFS driver can be referenced as `(uapfs)` when mounting APFS partitions.

The steps 1-3 should be made only once while step 4 is the standard way of using file system drivers in Linux environment.

APFS for Linux include a set of assistant script files for the simplification of building, installing and uninstalling procedures. Note that these assistant scripts may fail to work in customized Linux configurations or unsupported Linux distributions. Use `install.sh` and `uninstall.sh` script files to install and uninstall (correspondingly) APFS driver . The sections below describe the installation procedure in details.

#### Unpacking Setup Files

The setup files of the Linux-based version of the APFS for Linux are provided in the form of a gzip archive. The archive should be copied to the hard disk and decompressed. For example: For the APFS for Linux driver :



- create a separate folder:

```
$ mkdir /usr/tmp/uapfs
```

- change the current directory to the new one

```
$ cd /usr/tmp/uapfs
```

- use tar utility to unpack the initial archive

```
$ tar -xf /path/to/the/initial/archive/uapfs_*.tar.gz
```

Next step is to build and install the APFS for Linux driver.

### Using the INSTALL.SH Assistant Script

The assistant script "install.sh" provides the extremely easy and flexible way to make the APFS for Linux and install driver module in the system. Additionally, the script can reconfigure OS so that driver module is automatically rebuilt for another supported Kernel version and APFS volumes are automatically mounted with the UFSd driver.

Please note that development tools and kernel sources are required to present on the system and stay in the default locations to build and install the drivers.

### Installation

Just run the `install.sh` script with root privileges:

```
# ./install.sh
```

or

```
$ sudo ./install.sh
```

The assistant script will automatically perform the following actions:

1. Detect the Linux Kernel version;
2. Find kernel header files and libraries needed for building the drivers;
3. Add service for rebuilding driver module for supported Kernels via the DKMS library;
4. Build driver binary modules (`jnl.ko` and `uapfs.ko`);
5. Install the driver;

### INSTALL.SH default mode for the APFS for Linux driver

The assistant script `install.sh` always names the APFS for Linux driver module as `uapfs`. Now you can mount any APFS partition:

```
$ sudo mount -t uapfs <device> <mount_point>
```

## 3.4 Uninstalling driver

To completely remove the drivers from the current Kernel, one should dismount all APFS partitions mounted with the driver, uninstall the drivers and unload binary modules from the Kernel.

APFS for Linux provides tools for the driversuninstall automation.  
The assistant script `uninstall.sh` completely removes the driversfrom the system.

### Using the UNINSTALL.SH Assistant Script

The assistant script `uninstall.sh` provides the extremely easy and flexible way to deactivate and remove the driversfrom the system. The script performs correct deactivation, uninstallation and the complete removal of driver's.

### Uninstalling

Unmount all currently mounted APFS partitions and then run the '`uninstall.sh`' script:

```
$ sudo ./uninstall.sh
```

The assistant script will automatically perform the following actions:

1. Deactivate the driver modules. If the driver is still in use, the further script execution is aborted;
2. Uninstall the drivers;
3. Remove all binary and source files of the driver

## 4 Using The Driver

After building and installing Paragon APFS for Linux driver, it can be automatically loaded at the system startup. The driver allows to mount supported partitions and provides access to their whole contents.

### 4.1 Mounting volumes

APFS partition(s) should be mounted before testing driver operation on the corresponding system by executing the following command:

```
# mount -t uapfs [-o options] device mount_point
```

Please execute the 'mount | grep uapfs' command to make sure the device is mounted using the **uapfs** driver.

```
$ sudo mount -t uapfs /dev/sdf1 /mnt/uapfs
$ mount | grep uapfs
/dev/sdf1 on /mnt/uapfs type uapfs (rw,relatime,nls=utf8)
```

### 4.2 Mount options

Example on mount options usage:

```
# mount -t uapfs -o nls=utf8,mask=000,dmask=000 /device /mount_point
```

Below you can see the list of available options.

Mount option	Description
nls=<code_page>	This option informs the driver how to interpret path strings and translate them to Unicode and back. Up to 8 different code pages can be specified. The driver tries to use the codepages from specified list in order until it manages to translate all the characters in the string. If none of the specified codepages allows to translate all the characters, Kernel's default codepages is used. E.g. nls=utf8
nocase	If this option is used all file and directory operations (open, find, rename...) are case insensitive. Casing is preserved in the names of existing files and directories. <b>This mount option is enabled by default.</b>
ro	Mounts APFS volume in read-only mode.
rw	Mounts APFS volume with write access.
dmask=<value>	Changes the permissions for directories that already exist on a mounted volume.
fmask=<value>	Changes the permissions for files that already exist on a mounted volume.
gid=<groupid>	By specifying the 'gid' parameter you can set an owner group of the files. The groupid can be any name from /etc/group, or any number representing a group id. By default all existed files on a mounted APFS volume are owned by group root.

Mount option	Description
uid=<userid>	By specifying the 'uid' parameter you can set an owner of files. The userid can be any name from /etc/passwd, or any number representing a user id. By default all existed files on a mounted APFS volume are owned by root.
pass<num_of_subvolume>=<password>	Specifying passwords for encrypted subvolumes. When the password is specified this way, it cannot contain commas.  Example: -o pass1='this_is_password'
passfile= <path_to_file>	Specifying the file(s) containing a password for encrypted subvolumes.  Example: -o passfile=/tmp/passfile.txt Example of file contents: pass1=this_is_password pass2=this_is_second_password
subvolumes	Mount all APFS subvolumes. First subvolume is still mounted into mount point root, and all other subvolumes are mounted into 'Ufsd_Volumes' directory. This option does not work on 32-bit platforms.

## 5 Troubleshooting

### 1. Consult Documentation

Please consult documentation to make sure that the encountered behaviour is not by design, with special attention given to the sections related to the Troubleshooting.

### 2. Make sure the issue is not related to Linux itself

Make sure that the root cause of the issue is not related to Linux itself. For example, if an issue is discovered while performing a certain filesdump system-related operation on a volume mounted with Paragon's 'uapfs' driver, please verify that the same issue is not observed when the same operation is performed on a 'native' file system like Ext2fs, Ext3fs or FAT (except, of course, for operations specific to APFS file systems or to Paragon's driver itself).

### 3. Prepare to report the issue

After performing previous steps and making sure that the issue is related to Paragon's uapfs driver, prepare to report the issue to Paragon, based on the issue type:

- [Loading UAPFS driver module issue](#);
- [UAPFS driver mount volume issue](#).

### 4. Collect all information on the issue

The most important point in issue resolution process is quickly obtaining all the information related to the issue. Quick collection of required information is the key to resolving an issue faster. Please assist Paragon engineers, try to provide as detailed information on the issue, as possible.

## 5.1 Loading UAPFS driver module issue

Verify that another uapfs driver module is not already loaded into the platform. If the issue still exists, please collect and send the following information to Paragon:

- Linux Kernel version;
- Console output of the 'uapfs' driver loading;
- Dmesg output after `uapfs.ko` module wasn't loaded;
- `uapfs.ko` module used on your side.

## 5.2 UAPFS driver mount volume issue

Please verify that the test volume is formatted into the APFS file system and it is the first volume in the container. Check that the volume is mountable by MAC OS.

Please make sure that your use case is subject to [Limitations](#).

If the issue still exists, please collect and send the following information to Paragon:

- Linux Kernel version;
- Storage device information: device type, partitioning type, number of partitions, file system for the test volume, size of the test volume, etc;

- Mount command used on your side;
- dmesg output after 'Paragon APFS for Linux' was used to mount test volume.