

Object Storage Service

Tool Guide (obsfs)(ME-Abu Dhabi Region)

Issue 01
Date 2021-06-30



Copyright © Huawei Technologies Co., Ltd. 2022. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Introduction to obsfs.....	1
2 Environment Preparation.....	3
2.1 Resource Preparation.....	3
2.2 Downloading and Installing obsfs.....	5
2.3 Initializing obsfs.....	7
3 Operation Guide.....	8
3.1 Mounting a Parallel File System.....	8
3.2 Unmounting a Parallel File System.....	13
4 FAQs.....	15
5 Troubleshooting.....	17
5.1 Failure in Mounting a Parallel File System.....	17
5.2 Failure in Unmounting a Parallel File System.....	19
6 Related Operations.....	20
6.1 libfuse Installation Guide.....	20
A Change History.....	22

1 Introduction to obsfs

obsfs is a file system tool based on Filesystem in Userspace (FUSE) and provided by Object Storage Service (OBS), for mounting OBS parallel file systems to Linux operating systems. With obsfs, you can easily access the practically infinite storage of OBS as easily as accessing a local file system.

obsfs is a great option if you are used to storing data locally but your data is now stored in OBS.

Functions

- Lets you mount a parallel file system to your local Linux file system, so that you can manage objects stored in OBS as if they were stored locally.
- Supports synchronous upload. Any file uploaded to the mount directory can be synchronously uploaded to OBS.
- Synchronizes objects from a parallel file system to its mount directory in the local file system, so that you can copy, modify, rename, or truncate objects locally and they will be automatically updated in OBS.

Constraints

- The local directory that the parallel file system is mounted to does not need to be empty, but any files in that directory will be unavailable when the parallel file system is mounted. However, they will still be there when the parallel file system is unmounted later. To avoid any confusion this might cause, mount your parallel file system to an empty directory.
- obsfs supports mounting of parallel file systems but not OBS buckets.
- A parallel file system that is mounted locally cannot provide the same performance and functions as an actual local file system.
 - Files or folders in the mount directory do not support hard links.
 - There can be no more than 45 directory levels.

Applicable Operating Systems

obsfs is compatible with Linux. For details, see [Table 1-1](#).

Table 1-1 Obtaining obsfs

Method	Description	Compatible Distributions	Operation Guide
Download	Download and install the official software package and configure the environment (Linux), and obsfs is ready for use.	Ubuntu 16 and CentOS 7	Download and Install obsfs

2 Environment Preparation

2.1 Resource Preparation

Before using obsfs, you need to prepare the resources listed in [Table 2-1](#).

Table 2-1 Resource preparation

Resource	Description	Operation Guide
IAM user	To ensure account and resource security, it is safer to create an IAM user and assign permissions to that user specifically for accessing OBS resources. Then use the IAM account to access obsfs. NOTE The IAM user who uses obsfs to mount and access a parallel file system must have the same access permissions for OBS buckets. After an OBS role or policy is granted to an IAM user, the role or policy will apply to both OBS buckets and parallel file systems.	Creating an IAM User
Access keys (AK and SK)	Access keys (AK and SK) are used as credentials for your account and for IAM users. obsfs accesses OBS using access keys that are used to encrypt the signature of a request, ensuring the security and integrity of the request and that identities of the request sender and receiver are correct.	Creating Access Keys (AK and SK)
Parallel file system	A parallel file system is a container for storing data.	Creating a Parallel File System

Creating an IAM User

To ensure account and resource security, it is safer to create an IAM user and assign permissions to that user specifically for accessing OBS resources. Then use the IAM account to access obsfs.

Step 1 Log in to the management console using a cloud service account.

Step 2 On the top navigation menu, choose **Service List > Management & Deployment > Identity and Access Management**. The IAM console page is displayed.

Step 3 Create a user group and grant the OBS permissions to the user group.

User groups facilitate centralized user management and streamlined permission management. Users in the same user group have the same permissions. Users created in IAM inherit permissions from the groups to which they belong.

1. In the navigation pane on the left, click **User Groups**. The **User Groups** page is displayed.
2. Click **Create User Group**.
3. On the **Create User Group** page, enter a name for the user group and click **OK**.

The user group is then displayed in the user group list.

4. Click **Assign Permissions** in the **Operation** column of the row where the created user group resides.
5. In the **Group Permissions** area, locate the row that displays **Global service > OBS**, click **Attach Policy** in the **Operation** column, select the policy name, and click **OK**.

NOTE

In the **Policy Information** area, you can view the details about the policy.

Due to data caching, an RBAC policy and fine-grained policy involving OBS actions will take effect 10 to 15 minutes after it is attached to a user, a user group, or an enterprise project.

Step 4 Create a user.

1. In the navigation pane on the left, click **Users**. The **Users** page is displayed.
2. Click **Create User**.
3. Set the user information and click **Next**.

Table 2-2 User parameters

Parameter	Description
Username	The user name for logging in to the cloud service.

Parameter	Description
Credential Type	A credential refers to the identity credential used for user system authentication. In this example, password is selected. <ul style="list-style-type: none"> - Password: Used for accessing cloud services using the console or development tools. - Access key: Used for logging in to the cloud service using development tools. This credential type is more secure, and is recommended if the user does not need to use the console.
User Groups	The user will inherit the permissions granted to the user group. The default user group is admin , which has the administrator permissions and all of the permissions required to use all cloud resources.
Description	(Optional) brief description of the user

4. Select a type for password generation, set the email address and mobile number, and click **OK**.

Step 5 Use the created IAM user to log in to OBS Console and verify the user permissions.

----End

Creating Access Keys (AK and SK)

Access keys (AK and SK) are used as credentials for your account and for IAM users. obsfs accesses OBS using access keys that are used to encrypt the signature of a request, ensuring the security and integrity of the request and that identities of the request sender and receiver are correct. For details about how to create access keys, see [Access Keys \(AK/SK\)](#). If you already have them, skip this part.

NOTE

Each account or IAM user can create a maximum of two valid access keys.

Creating a Parallel File System

For details about how to create a parallel file system, see the *Parallel File System Feature Guide*.

2.2 Downloading and Installing obsfs

For common Linux distributions, such as CentOS 7 and Ubuntu 16, there are obsfs installation packages available for download. After downloading obsfs, configure the Linux environment and then use the mount command to mount the parallel file system.

Downloading obsfs Software Packages

Table 2-3 obsfs download links

Linux Distribution	Download Link
CentOS 7	https://obs-community.obs.ae-ad-1.g42cloud.com/obsfs/current/obsfs_CentOS7.6_amd64.tar.gz
Ubuntu 16	https://obs-community.obs.ae-ad-1.g42cloud.com/obsfs/current/obsfs_Ubuntu16.04_amd64.tar.gz

Installing obsfs

1. On a PC running Windows, download the obsfs software package based on the version of the Linux distribution.
2. Use a common cross-platform transmission tool (such as WinSCP) to transfer the software package to your Linux OS.
3. Run the following decompression command on the computer running Linux:
`tar -xzf absolute path of the obsfs software package /obsfs_XXX.tar.gz`

NOTE

You can also run the following command to decompress the software package to a specified directory:

```
tar -xzf Absolute path of obsfs software package/obsfs_XXX.tar.gz -C Specified directory
```

4. Go to the directory where the software package is decompressed and run the following command to install obsfs:
`bash install_obsfs.sh`

NOTE

After you run the installation command, crontab will add a scheduled task for compressing and dumping obsfs logs. The dump path is `/var/log/obsfs/`.

An example of such task added by crontab: `*/10 * * * * [-f /opt/dfv/obsfs/obsfs_log_rotate.sh] && bash /opt/dfv/obsfs/obsfs_log_rotate.sh`

Configuring Operating Environment

The operating environment of obsfs depends on the software packages such as openssl-devel, fuse, and fuse-devel. Before running obsfs, you need to run commands to configure the dependent environment.

NOTE

- obsfs supports libfuse 2.9.7, 2.9.8, or 2.9.9. If your system has a different version of libfuse installed, remove it and install the correct version by referring to [libfuse Installation Guide](#).
- You can view the libfuse version by running the system's version query command or run the following command:
`find / -name libfuse.so*`
- CentOS 7
`yum install -y openssl-devel fuse fuse-devel`

- Ubuntu 16
`apt-get install -y libfuse-dev libcurl4-openssl-dev`

Verification

Go to the directory where obsfs has been installed and run the following command to verify the installation.

```
./obsfs --version
```

If the obsfs version is displayed in the command output, obsfs has been installed, and you can start [initializing](#) and [using](#) obsfs. [libfuse Installation Guide](#)

2.3 Initializing obsfs

Before using obsfs, you need to write the access keys (AK and SK) into the key file, enabling user authentication before using parallel file systems.

This topic describes how to write the access keys to the `/etc/passwd-obsfs` file.

Prerequisites

You have obtained the access keys (AK and SK). For details, see [Resource Preparation](#).

Procedure

Step 1 Open the CLI.

Step 2 Run the following command to write the access keys into the `/etc/passwd-obsfs` file:

```
echo AK:SK > /etc/passwd-obsfs
```

Step 3 Run the following command to check whether the key information is successfully configured:

```
cat /etc/passwd-obsfs
```

Step 4 Run the following command to set the key file to be used only by the current user:

```
chmod 600 /etc/passwd-obsfs
```

----End

3 Operation Guide

3.1 Mounting a Parallel File System

To use obsfs to manage your objects in OBS, you need to mount an OBS parallel file system locally. After the parallel file system is mounted, working on the mount directory is working on the parallel file system. For example, you can upload an object to OBS simply by saving it to the mount directory.

Constraints

- **obsfs supports mounting of parallel file systems but not OBS buckets.**
- The local directory that the parallel file system is mounted to does not need to be empty, but any files in that directory will be unavailable when the parallel file system is mounted. However, they will still be there when the parallel file system is unmounted later. To avoid any confusion this might cause, mount your parallel file system to an empty directory.
- Files or folders in the mount directory do not support hard links.
- There can be no more than 45 directory levels.
- If a parallel file system is manually mounted by running a command, it needs to be mounted again every time the Linux is restarted.
- The actual size and used space of a parallel file system vary with your actual usage.

NOTE

You can configure auto mounting of a parallel file system upon startup. For details, see [Mounting a Parallel File System Automatically upon Startup](#).

- A parallel file system can be mounted to multiple ECSs.

Prerequisites

- obsfs has been obtained. For details, see [Downloading and Installing obsfs](#).
- obsfs has been initialized. For details, see [Initializing obsfs](#).
- There is a mount directory in the local file system.

 NOTE

You can run the **mkdir** command to create a directory for mounting a parallel file system. For example, to create a directory named **mountpoint**, the command is:

```
mkdir mountpoint
```

Mounting a Parallel File System Manually

Step 1 Open the CLI.

Step 2 Run the **cd** command to go to the directory where obsfs is located.

Step 3 Run the mount command in the following format:

```
./obsfs Parallel file system name Local mount directory -o url=Region endpoint address -o passwd_file=Key file path -o big_writes -o max_write=131072 -o use_ino Mount parameters
```

- *Parallel file system name*: name of the parallel file system you want to mount.

 NOTE

A specific directory in a parallel file system can be mounted locally. If you mount just a directory, you can access only this directory in the parallel file system after entering the local mount directory. The mount command is as follows:

```
./obsfs Parallel file system name:/Directory name Local mount directory -o url=Region endpoint address -o passwd_file=Key file path -o big_writes -o max_write=131072 -o use_ino Mount parameters
```

- *Local mount directory*: the absolute or relative path of the directory on which a parallel file system is to be mounted.
- *Region endpoint address*: Obtain the endpoint address according to the region where the mounted parallel file system resides.
- *Mount parameters*:

Mount parameters, excepting **passwd_file** and **use_ino**, are optional. Select proper parameters according to your actual needs. The format is as follows: **-o Parameter=Value**. If you do not need to specify a parameter value, use only **-o Parameter**. [Table 3-1](#) lists OBS mount parameters and their descriptions.

 NOTE

For details about obsfs mount parameter configurations, run the **./obsfs --help** command to learn more.

Table 3-1 obsfs mount parameters

Parameter	Value	Description
passwd_file	Type: String Value: Set this parameter to the path of the access key file configured in Initializing obsfs .	The access key file contains the AK and SK information required for access obsfs.

Parameter	Value	Description
use_ino	Required	If this parameter is specified, obsfs allocates the inode number.
big_writes	Not required	After the configuration, the maximum size of the cache can be modified.
max_write	Type: Integer Recommended value: 128 KB	This parameter is valid only when big_writes is configured. The default write size is 4 KB. The recommended value is 128 KB.
nonempty	Not required	This parameter allows the mount point to be a non-empty directory.
allow_other	Not required	This parameter allows other users to access the parallel file system.
max_background	Type: Integer Recommended value: 100	You can use this parameter to set the maximum number of waiting requests in the background.
use_path_request_style	Not required	Indicates the path-based access (legacy API). If this parameter is configured, you need to use the specified endpoint address to access OBS. NOTE If the value of the URL in the command is an IP address, this parameter must be carried in the mount command. Otherwise, a domain name resolution error is reported.
umask	Not required	This parameter masks the file permissions. The umask command is executed upon each access to the file system, and automatically configures the mask for new permissions.

Parameter	Value	Description
obsfslog	Not required	<p>This parameter enables the loading of log configuration parameters from the configuration file, including the log mode and level. You can change the log level without restarting the process when this parameter is enabled.</p> <p>NOTE The path of the configuration file is: /etc/obsfsconfig. If you want to use this parameter, you need to manually create the configuration file path on the local host and write the content by referring to the following example and comments.</p> <p>An example of the configuration file content: //Log mode. The value 0 indicates that the log mode is displayed on the client. If the value is 1, the log file is generated in the following path: /var/log/obsfs, which is named after the process ID. If the value is 2, the log is written into the system log. dbglogmode=1 //Log level. dbg, info, warn, err, crit. dbglevel=info</p>

Step 4 Run the following command to view the result:

```
df -h
```

If the information similar to the following is displayed, the parallel file system is successfully mounted:

Filesystem	Size	Used	Avail	Use%	Mounted on
obsfs	256T	0	256T	0%	/path/to/mountpoint

 **NOTE**

The actual size and used space of a parallel file system vary with your actual usage.

If the preceding information is not displayed, the parallel file system fails to be mounted. In this case, you can add the following parameters to the command, so that the mount process and detailed debug logs are displayed in the command output:

```
-d -d -f -o f2 -o curldb
```

Then you can locate the fault based on the error information. For details about troubleshooting of common faults, see [Failure in Mounting a Parallel File System](#).

----End

Command Example

In this example, the **filesystem001** parallel file system and the access key file path **/etc/passwd-obsfs** are used as the prerequisites for mounting an OBS parallel file system.

- Mount the parallel file system **filesystem001** to the mount point **/mnt/mount_path/** and allow the directory to be non-empty.

```
./obsfs filesystem001 /mnt/mount_path/ -o url=obs.ae-ad-1.g42cloud.com -o passwd_file=/etc/passwd-obsfs -o big_writes -o max_write=131072 -o nonempty -o use_ino
```
- Mount the parallel file system **filesystem001** to the **/mnt/mount_path/** directory and set the write cache size.

```
./obsfs filesystem001 /mnt/mount_path/ -o url=obs.ae-ad-1.g42cloud.com -o passwd_file=/etc/passwd-obsfs -o big_writes -o max_write=131072 -o max_background=100 -o use_ino
```
- Mount the **filesystem001** parallel file system to the mount point **/mnt/mount_path/** and allow other users to access the directory.

```
./obsfs filesystem001 /mnt/mount_path/ -o url=obs.ae-ad-1.g42cloud.com -o passwd_file=/etc/passwd-obsfs -o big_writes -o max_write=131072 -o max_background=100 -o use_ino
```

NOTE

If the value of this parameter is an IP address, add the **-o use_path_request_style** parameter to the mount command. Otherwise, a domain name resolution error is reported.

Mounting a Parallel File System Automatically upon Startup

The following uses CentOS 7 as an example to describe how to configure the parallel file system to automount at startup.

Step 1 Log in to the Linux server as user **root**.

Step 2 Run the following command to create a **/home/startobsfs.sh** script:

```
vi /home/startobsfs.sh
```

Step 3 Press **i** to enter the editing mode, and write the following comments and mount commands to the **/home/startobsfs.sh** script.

```
#!/bin/bash  
cd Absolute path of obsfs  
./obsfs Parallel file system name Local mount directory -o url=Region endpoint address -o passwd_file=Key file path -o big_writes -o max_write=131072 -o use_ino Mount parameters
```

NOTE

For details about the mount command parameters, see [Step 3](#) in section "Mounting a Parallel File System Manually."

Step 4 Press **Esc** and enter **:wq** to save and exit the script.

Step 5 Run the following command to grant the **root** user the permission to execute the script:

```
chmod +x /home/startobsfs.sh
```

Step 6 Run the following command to grant the **root** user the permission to execute the **rc.local** file:

```
chmod +x /etc/rc.d/rc.local
```

NOTE

The location of the **rc.local** file varies depending on the OS. In CentOS 7, the file is stored in **/etc/rc.d/rc.local**.

Step 7 Add the following command to the end of `/etc/rc.d/rc.local`, save the file, and exit:

```
bash /home/startobsfs.sh
```

 **NOTE**

If the `/etc/rc.d/rc.local` file contains an exit command, for example, `exit 0`, add the command prior to the exit command.

Step 8 Restart the server and verify the mounting.

After the server is restarted, run the following command to check whether the parallel file system is successfully mounted:

```
df -h
```

 **NOTE**

If you no longer want the parallel file system to automount at startup, delete the script from `rc.local`.

----End

Follow-up Procedure

Before adding, modifying, or deleting any parameters for a mounted parallel file system, you need to unmount the parallel file system and then mount it again. For details about how to unmount a parallel file system, see [Unmounting a Parallel File System](#).

3.2 Unmounting a Parallel File System

If you do not need to access a mounted parallel file system or you need to add, modify, or delete parameters it has configured, you can unmount it.

 **NOTE**

Unmounting a parallel file system does not affect data stored in the parallel file system.

Prerequisites

The parallel file system is currently mounted.

Procedure

Step 1 Open the CLI.

Step 2 Check whether the current directory is the mount point.

- If it is, navigate to a different directory. Then go to [Step 3](#).
- If it is not, go to [Step 3](#).

Step 3 Run the following command to unmount a parallel file system:

```
umount Local mount point
```

Step 4 Run the following command to confirm that the parallel file system has been unmounted:


```
df -h
```

```
----End
```

4 FAQs

Can I Mount a Parallel File System to Multiple Clients at the Same Time?

Yes.

Can Multiple Parallel File Systems Be Mounted to the Same Client?

Yes. Each mounting process is independent.

What Can I Do If Error Message "error: conflicting types for 'int64_t'" Is Displayed When Compiling libfuse.2.9.7 on an Arm Platform?

Modify the content in lines 92 and 93 of the `include/fuse_kernel.h` to the following:

```
typedef unsigned long long __u64;  
typedef long long __s64;
```

[Figure 4-1](#) and [Figure 4-2](#) are examples of the files before and after the modification.

Figure 4-1 File content before the modification

```
87  
88 #ifndef _LINUX_FUSE_H  
89 #define _LINUX_FUSE_H  
90  
91 #include <sys/types.h>  
92 #define __u64 uint64_t  
93 #define __s64 int64_t  
94 #define __u32 uint32_t  
95 #define __s32 int32_t  
96 #define __u16 uint16_t  
97
```

Figure 4-2 File content after the modification

```
87
88 #ifndef _LINUX_FUSE_H
89 #define _LINUX_FUSE_H
90
91 #include <sys/types.h>
92 typedef unsigned long long __u64;
93 typedef long long __s64;
94 #define __u32 uint32_t
95 #define __s32 int32_t
96 #define __u16 uint16_t
97
```

How Can Other Users Access a Mounted Parallel File System?

Generally, only the user who mounted the parallel file system has access.

To authorize other users the access to the mount directory, you can specify the **allow_other** parameter in the mount command, for example:

```
./obsfs filesystem001 /mnt/mount_path/ -o url=obs.ae-ad-1.g42cloud.com -o passwd_file=/etc/passwd-obsfs -o big_writes -o max_write=131072 -o allow_other -o use_ino
```

5 Troubleshooting

5.1 Failure in Mounting a Parallel File System

This section describes some typical reasons a parallel file system may not mount and what actions you can take to resolve the issue.

Symptom 1: "The specified bucket does not exist" or Similar Errors Occur When Mounting a Parallel File System

Fault Locating

The parallel file system to be mounted does not exist.

Solution

Create a parallel file system and mount it again. For details about how to create a parallel file system, see [Creating a Parallel File System](#).

Symptom 2: "file system not support this request: this bucket not support filesystem" or Similar Errors Occur When Mounting a Parallel File System

Fault Locating

The mounted bucket is not a parallel file system.

Solution

Create a parallel file system and mount it again. For details about how to create a parallel file system, see [Creating a Parallel File System](#).

Symptom 3: "The request signature we calculated does not match the signature you provided. Check your key and signing method." or Similar Errors Occur When Mounting a Parallel File System

Fault Locating

Authentication failed. The access keys (AK and SK) are incorrect.

Solution

Check whether the AK and SK information in the key file is correctly configured. For details about the check method, see [Step 3](#) in the section "Initializing obsfs."

Symptom 4: "specified passwd_file is not readable" or Similar Errors Occur When Mounting a Parallel File System

Fault Locating

The key file path configured in the **passwd_file** file is incorrect.

Solution

Check whether the key file path configured in the **passwd_file** file is correct. For details about the check method, see [Step 3](#) in the section "Initializing obsfs."

Symptom 5: "Access Denied" or Similar Errors Occur When Mounting a Parallel File System

Fault Locating

Authentication failed. The IAM user does not have OBS operation permissions.

Solution

Authorize the user with required OBS operation permissions. For details, see [Resource Preparation](#).

Symptom 6: "unable to access MOUNTPOINT /obsfs: Transport endpoint is not connected" or Similar Errors Occur When Mounting a Parallel File System

Fault Locating

The mount point is occupied by another obsfs mount process.

Solution

Run the **df** command to query the partitions and points that have been mounted to. Then select an unoccupied mount point and mount the bucket again.

Symptom 7: "unknown option" or Similar Errors Occur When Mounting a Parallel File System

Fault Locating

One or more parameters in the mount command are incorrectly set and therefore failed to be identified by obsfs.

Solution

Check the mount parameters by referring to [Table 3-1](#) in the section "Mounting the Parallel File System".

Symptom 8: "unable to access MOUNTPOINT /mnt/obsfs: No such file or directory" or Similar Errors Occur When Mounting a Parallel File System

Fault Locating

The mount directory does not exist.

Solution

Use the **mkdir** command to create a directory or mount the bucket to an existing directory.

5.2 Failure in Unmounting a Parallel File System

This topic describes the possible causes of and solutions to common failures in unmounting parallel file systems.

Symptom 1: "target is busy" or Similar Errors Occur When Unmounting a Parallel File System

Fault locating:

During unmounting, the CLI terminal stays in the mount directory or its subdirectory.

Solution:

Exit the mount directory.

6 Related Operations

6.1 libfuse Installation Guide

This section uses libfuse 2.9.7 as an example target version to describe how to update libfuse on CentOS 7, SUSE 12, and Ubuntu 16.

Prerequisites

The source code of libfuse has been downloaded and then compiled to generate libfuse.so.2.9.7.

Download address of libfuse: <https://github.com/libfuse/libfuse/releases/tag/fuse-2.9.7>

```
tar -zxvf fuse-2.9.7.tar.gz
cd fuse-2.9.7/ && ./configure && make && make install
echo -e "\n/usr/local/lib" >> /etc/ld.so.conf
ldconfig
```

Installing libfuse on CentOS 7

libfuse.so.2.9.2 is installed on CentOS 7 by default.

Ensure that you have downloaded the libfuse source code and generated libfuse.so.2.9.7 after compilation as described in [Prerequisites](#). Then perform the following steps.

Step 1 Run the following command to locate the links of the libfuse.so.2.9.2 library:

```
find / -name libfuse.so*
```

Step 2 Run the following command to copy libfuse.so.2.9.7 to the directory where the libfuse.so.2.9.2 library resides.

```
cp /usr/local/lib/libfuse.so.2.9.7 /usr/lib64/
```

Step 3 Run the following command to remove all links of the libfuse.so library of the earlier version:

```
rm -f /usr/lib64/libfuse.so
rm -f /usr/lib64/libfuse.so.2
```

Step 4 Run the following command to create links of libfuse.so.2.9.7 library similar to the deleted ones of the libfuse.so library of the earlier version:

```
ln -s /usr/lib64/libfuse.so.2.9.7 /usr/lib64/libfuse.so  
ln -s /usr/lib64/libfuse.so.2.9.7 /usr/lib64/libfuse.so.2
```

----End

Installing libfuse on SUSE 12

libfuse.so.2.9.3 is installed on SUSE 12 by default.

Ensure that you have downloaded the libfuse source code and generated libfuse.so.2.9.7 after compilation as described in [Prerequisites](#). Then perform the following steps.

Step 1 Run the following command to locate the links of the libfuse.so.2.9.3 library:

```
find / -name libfuse.so*
```

Step 2 Run the following command to copy libfuse.so.2.9.7 to the directory where the libfuse.so.2.9.3 library resides.

```
cp /usr/local/lib/libfuse.so.2.9.7 /usr/lib64/
```

Step 3 Run the following command to remove all links of the libfuse.so library of the earlier version:

```
rm -f /usr/lib64/libfuse.so.2  
rm -f /lib64/libfuse.so.2.9.3
```

Step 4 Run the following command to create links of libfuse.so.2.9.7 library similar to the deleted ones of the libfuse.so library of the earlier version:

```
ln -s /usr/lib64/libfuse.so.2.9.7 /usr/lib64/libfuse.so.2  
ln -s /usr/lib64/libfuse.so.2.9.7 /lib64/libfuse.so.2.9.7
```

----End

Installing libfuse on Ubuntu 16

libfuse.so.2.9.4 is installed on Ubuntu 16 by default.

Ensure that you have downloaded the libfuse source code and generated libfuse.so.2.9.7 after compilation as described in [Prerequisites](#). Then perform the following steps.

Step 1 Run the following command to locate the links of the libfuse.so.2.9.4 library:

```
find / -name libfuse.so*
```

Step 2 Run the following command to copy libfuse.so.2.9.7 to the directory where the libfuse.so.2.9.4 library resides.

```
cp /usr/local/lib/libfuse.so.2.9.7 /lib/x86_64-linux-gnu/
```

Step 3 Run the following command to remove all links of the libfuse.so library of the earlier version:

```
rm -f /lib/x86_64-linux-gnu/libfuse.so.2
```

Step 4 Run the following command to create links of libfuse.so.2.9.7 library similar to the deleted ones of the libfuse.so library of the earlier version:

```
ln -s /lib/x86_64-linux-gnu/libfuse.so.2.9.7 /lib/x86_64-linux-gnu/libfuse.so.2
```

----End

A Change History

Release Date	What's New
2021-06-30	This is the first official release.