

**Object Storage Service**

# **Parallel File System Feature Guide(Paris Region)**

**Issue**                01  
**Date**                 2021-10-30



**Copyright © Huawei Technologies Co., Ltd. 2024. 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.

# Security Declaration

## Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

---

# Contents

---

<b>1 Introduction.....</b>	<b>1</b>
1.1 About Parallel File System.....	1
1.2 Application Scenarios.....	1
1.3 Constraints.....	1
1.4 Using PFS.....	2
<b>2 Managing Parallel File Systems on OBS Console.....</b>	<b>3</b>
2.1 Creating a Parallel File System.....	3
<b>3 Managing Parallel File Systems by APIs.....</b>	<b>5</b>
3.1 Supported APIs.....	5
<b>A Change History.....</b>	<b>12</b>

# 1 Introduction

---

## 1.1 About Parallel File System

Parallel File System (PFS) is a high-performance semantic file system provided by OBS. It features access latency in milliseconds, TB/s-level bandwidth, and millions of IOPS.

You can access data in a PFS via OBS APIs.

## 1.2 Application Scenarios

PFS is highly compatible, scalable, and reliable, and delivers amazing performance.

**It is mainly used in the following scenarios:**

Big data: log analysis, content recommendation, operation reports, user profiling, and interactive analysis

## 1.3 Constraints

### Operations

- An existing OBS bucket cannot be changed to a parallel file system. For details about how to create a parallel file system, see [Creating a Parallel File System](#).
- Parallel file systems do not support quota configuration. By default, there is no quota limit.

### Functions

- Image processing currently cannot be used to process (such as downsize, resize, or watermark) images stored in parallel file systems.
- Server-side encryption is not supported.
- Cross-region replication is not supported.
- Versioning is not supported.

- Static website hosting is not supported.
- Changing file storage class is not supported.
- Configuration of default storage class for a parallel file system is not supported.

## 1.4 Using PFS

You can use the console and REST APIs to process files stored in PFS anytime, anywhere, and quickly obtain the processed files. PFS supports both POSIX and OBS APIs, so you can process files the same way you process objects. There is flexible conversion supported between objects and files.

You can use PFS in the following ways:

 **NOTE**

Access permissions for OBS buckets also apply to parallel file systems. Before using a parallel file system, ensure that you have the required permissions to access OBS buckets.

**Table 1-1** How to use PFS

Method	Function	Reference
Console	On the console, you can create parallel file systems and manage them.	<a href="#">Creating a Parallel File System</a>
OBS API	Use parallel file systems by calling OBS APIs.	<a href="#">Supported APIs</a>

# 2 Managing Parallel File Systems on OBS Console

---

## 2.1 Creating a Parallel File System

You can create a parallel file system on OBS Console.

### Procedure

**Step 1** On the console homepage, click **Service List** in the upper left corner and choose **Storage > Object Storage Service**.

**Step 2** In the navigation pane, select **Parallel File System**.

**Step 3** In the upper right corner of the page, click **Create Parallel File System**.

**Step 4** Select a region and enter a name for the parallel file system.

#### NOTE

- Once a parallel file system is created, its name cannot be changed.
- URLs do not support uppercase letters and cannot distinguish between names containing uppercase or lowercase letters. For example, if you attempt to access the parallel file system **MyFileSystem** using a URL, the file system name will be resolved to **myfilesystem**, causing an access error. For this reason, a parallel file system name can contain only lowercase letters, digits, periods (.), and hyphens (-).

**Step 5** Configure a data redundancy policy. If you select multi-AZ storage, data will be stored in multiple AZs for a higher reliability.

#### NOTE

Once a parallel file system is created, you cannot change the data redundancy policy, so take care when selecting a data redundancy policy.

**Step 6** Configure a policy. You can select **Private**, **Public Read**, or **Public Read and Write** for the parallel file system.

**Step 7** Confirm the settings at the bottom of the page and click **Create Now**.

**Step 8** View the file system you created just now in the parallel file system list.

Then, you can use the parallel file system the same way you use a bucket. For details about how to use PFS, see [Using PFS](#).

----End



# 3 Managing Parallel File Systems by APIs

## 3.1 Supported APIs

This section focuses on the compatibility of parallel file systems and existing OBS APIs, and provides a description of APIs that are partly compatible.

For details about the OBS APIs, see the *Object Storage Service API Reference*.

### API Operations on Buckets

Table 3-1 API operations on buckets

API	PFS Compatible	Differences
Listing buckets	Yes	The <b>x-obs-bucket-type:POSIX</b> header field is required for obtaining the list of parallel file systems.
Creating a bucket	Yes	The <b>x-obs-fs-file-interface:Enabled</b> header field is required for creating a parallel file system.
Listing objects in a bucket	Yes	-
Obtaining bucket metadata	Yes	-
Obtaining bucket region locations	Yes	-
Deleting a bucket	Yes	-

## API Operations on Advanced Bucket Settings

**Table 3-2** API operations on advanced bucket settings

API	PFS Compatible	Differences
Configuring a bucket policy	Yes	-
Obtaining bucket policy information	Yes	-
Deleting a bucket policy	Yes	-
Configuring a bucket ACL	Yes	-
Obtaining bucket ACL information	Yes	-
Configuring logging for a bucket	Yes	-
Obtaining a bucket logging configuration	Yes	-
Configuring bucket lifecycle rules	No	-
Obtaining bucket lifecycle configuration	No	-
Deleting bucket lifecycle rules	No	-
Configuring versioning for a bucket	No	-
Obtaining bucket versioning status	No	-
Configuring event notification for a bucket	Yes	-

API	PFS Compatible	Differences
Obtaining the event notification configuration of a bucket	Yes	-
Configuring storage class for a bucket	No	-
Obtaining bucket storage class information	No	-
Configuring cross-region replication for a bucket	No	-
Obtaining the cross-region replication configuration of a bucket	No	-
Deleting the cross-region replication configuration of a bucket	No	-
Configuring bucket storage quota	Yes	-
Querying bucket storage quota	Yes	-
Querying information about used space in a bucket	Yes	-
Configuring a custom domain name for a bucket	Yes	-
Obtaining the custom domain name of a bucket	Yes	-

API	PFS Compatible	Differences
Deleting a custom domain name of a bucket	Yes	-
Configuring bucket encryption	No	-
Obtaining bucket encryption configuration	No	-
Deleting the encryption configuration of a bucket	No	-

## API Operations for Static Website Hosting

**Table 3-3** API operations for static website hosting

API	PFS Compatible	Differences
Configuring static website hosting for a bucket	No	-
Obtaining the static website hosting configuration of a bucket	No	-
Deleting the static website hosting configuration of a bucket	No	-
Configuring bucket CORS	No	-
Obtaining the CORS configuration of a bucket	No	-

API	PFS Compatible	Differences
Deleting the CORS configuration of a bucket	No	-
OPTIONS buckets	No	-
OPTIONS objects	No	-

## API Operations on Objects

Table 3-4 API operations on objects

API	PFS Compatible	Differences
PUT objects	Yes	<ul style="list-style-type: none"> <li>• Headers not supported: <b>x-obs-storage-class</b>, <b>x-obs-website-redirect-location</b>, <b>x-obs-server-side-encryption</b>, <b>x-obs-server-side-encryption-kms-key-id</b>, <b>x-obs-server-side-encryption-customer-algorithm</b>, <b>x-obs-server-side-encryption-customer-key</b>, <b>x-obs-server-side-encryption-customer-key-MD5</b>, <b>success-action-redirect</b>, and <b>x-obs-expires</b></li> <li>• Objects uploaded using this API cannot be directly stored in the Infrequent Access or Archive storage class and are stored in the Standard storage class by default. You can later change the storage class by using a lifecycle rule or modifying the metadata.</li> </ul>
POST objects	Yes	Headers not supported: <b>x-obs-storage-class</b> , <b>x-obs-website-redirect-location</b> , <b>x-obs-server-side-encryption</b> , <b>x-obs-server-side-encryption-kms-key-id</b> , <b>x-obs-server-side-encryption-customer-algorithm</b> , <b>x-obs-server-side-encryption-customer-key</b> , <b>x-obs-server-side-encryption-customer-key-MD5</b> , <b>success-action-redirect</b> , and <b>x-obs-expires</b>
Copying objects	Yes	Data can only be replicated between a parallel file system and a bucket if the parallel file system and the bucket are in the same cluster.

API	PFS Compatible	Differences
Obtaining object content	Yes	-
Obtaining object metadata	Yes	-
Deleting objects	Yes	-
Batch deleting objects	Yes	-
Restoring Cold objects	No	-
Appending objects	No	-
Configuring object ACL	Yes	-
Obtaining object ACL information	Yes	-
Modifying object metadata	Yes	In a parallel file system, the storage class of a directory cannot be changed. To change the storage class of a file in the directory, modify the metadata of the file or use a lifecycle rule to change the storage class of files in batches.
Modifying an object	Yes	This is a PFS only API, and is not supported by OBS buckets.
Truncating an object	Yes	This is a PFS only API, and is not supported by OBS buckets.
Renaming an object	Yes	This is a PFS only API, and is not supported by OBS buckets.

## API Operations for Multipart Tasks

Table 3-5 API operations for multipart tasks

API	PFS Compatible	Differences
Listing initialized multipart tasks in a bucket	Yes	-

API	PFS Compatible	Differences
Initiating multipart upload tasks	Yes	-
Uploading parts	Yes	-
Copying parts	Yes	Copying parts is not supported for an appended file.
Listing uploaded parts	Yes	-
Merging parts	Yes	-
Canceling multipart tasks	Yes	-

# A Change History

---

Release Date	What's New
2021-10-30	This is the first official release.