IBM Cloud Object Storage System™
Version 3.14.7

Native File Interface

Administration Guide

This edition applies to IBM Cloud Object Storage System™ Native File Interface Administration Guide and is valid until replaced by new editions.

Contents

Document information

Intended purpose and audience

This guide explains the capabilities and applications of the IBM Cloud Object Storage System™ Native File Interface.

Chapter 1. Overview

This new feature is designed to provide a Network Attached Storage (NAS) interface to the IBM Cloud Object Storage System™.

Note: This feature is not compatible with Concentrated Dispersal (CD).

The user is able to create multiple NAS volumes (File Systems), each with storage backed by the system. The feature exposes user-defined Network File System server (NFS version 3) exports, which you can mount and interact with as you would expect from a typical NFS server. File metadata is managed and maintained by a clustered database solution, and file data is stored in the system. The system is scalable linearly as nodes are added.

Chapter 2. Benefits

A key benefit of this feature is to provide access to the IBM Cloud Object Storage System™ by using the NFS protocol. Users can access content via NFS or the S3 API, which avoids an immediate need to update existing applications to use Object Storage APIs.

Chapter 3. Terms

A number of terms have special meaning in this document.

seed node
A configured initial contact point.

File Accesser Device
New appliance type where the collection of NAS interface services runs including the NFS server and metadata management services.

File Server Pool
Collection of File Accesser Devices that are configured identically and participate in High Availability failover/failback pairs.

File System
Logical unit that encompasses a collection of File data and metadata. Each File System is mapped to a unique vault where file data is stored.

Share Equivalent to the concept of an NFS export. Connecting to Shares via an NFS client takes the form:

hostname:/fs/*file_system*/*share_name*

Site Manager concept to group devices within a physical location. File Accesser Devices that are within the same site should also share physical deployment locations. The Site designation is used internally by the metadata database. Performance is affected if all File Accesser Devices that are defined in the same Site are not deployed together in a low latency environment. All File Accesser Devices in the same Site participate in a unified device cluster for metadata storage.

Chapter 4. Components

NFSv3 server

Adheres to NFSv3 specification (see the NFSv3 RFC).

The following exceptions are noted:

- TCP IPv4 only
- No hard link support
- No special devices
- Only AUTH_NONE and AUTH_SYS authentication is supported.

Mount 3 protocol:

- Both TCP and UDP support
- No File Mounting support

Filer API

- Splits file metadata from file data.
- File data that is stored directly in IBM Cloud Object Storage System™.
- File metadata stored in scalable clustered database on Filer nodes.

Scalable metadata database (Cassandra)

- Every deployed node participates in clustered metadata database.
- Addition of nodes adds performance and capacity.
- Content is stored redundantly (3x replica).
- Content is mirrored between two sites if deployed in two facilities.
- Management of the metadata database is performed through the Manager Web Interface. Direct management or configuration of Cassandra is not supported.

Chapter 5. Use cases

Large-scale data archival

- Designed to support billions of files and petabytes of data.
- Long-term storage.
- Active reading of any data.
- Infrequent update of written data.

Backing storage for applications better suited for Object Storage

- Written once.
- Read often.
- Updated infrequently.

Application migration to Object Storage

- Supports object API and NFS access. Content that is written over NFS can later be accessed over S3 API through the S3 API endpoint on File Accesser nodes.

Chapter 6. Configuration requirements

Network Connectivity

1. Must be 10 Gbps between File Accesser® and Accesser devices.
2. Should be <1 ms RTT between File Accesser and Accesser devices.

All Devices

1. Must be properly configured to use NTP.

Accesser Devices

1. Must be hardware models A3100, A3105, or A4105 (A3105 or A4105 highly recommended).
2. Must be configured with the following Advanced System Configuration parameters:
   - `segmentation.content-header-cache-enabled = true`
   - `segmentation.content-header-cache-read = true`
3. Should be configured with four or more Accesser devices in the Accesser Pools used with Native File Interface.

File Accesser Devices

1. Must be configured with three or more File Accesser devices.
2. Must be able to communicate directly between one another.
3. Must allow access on ports 111, 2048 and 2049 (TCP and UDP) for NFS connectivity from client machines.
4. Must not change the physical IP address after a File Accesser is added to a File Server Pool without removal and reimaging of the device.

Vault or Vault Templates used with Native File Interface

1. Must be configured to use the S3 API.
2. Must not have “Named Index” enabled.
3. Must not be configured in a Vault Mirror configuration.
4. Must not have “Enable Versioning” enabled.
5. Must not have “Delete Restricted” enabled.
6. Should not have “Recovery Listing” enabled.
7. Should be configured with an IDA where Write Threshold = Performance Threshold.
8. Must have “Anonymous Access” enabled IF “Use SSL” is disabled on the file system that is associated with the vault.

Storage Pool used with Native File Interface

1. Must be configured to use Packed Storage.

Chapter 7. Operations

Note: For the purposes of this document and the Manager UI, reference to “File Accesser” and “gateway” devices might be used interchangeably.

Hardware

A list of hardware components.

The AF5100 consists of the following components:

- 1U Rack-mountable unit, deployable in (3)+ unit cluster configuration.
- 250 W avg power consumption.
- Dual SFP+ 10G Interfaces.
- IPMI interface (multicast between all nodes that are needed for HA).
- (8) 960 GB SSD drives configured as single RAID 5 Array.
- (2) E5-2683 CPUs • 256 GB RAM.

Network configuration of File Accesser Devices

Refer to the Hardware configuration guide for Slicestor® Devices, Accesser® Devices and File Accessers® for Network configuration.

Configuring File Accesser Devices

Create a pool of the Native File Accesser Devices and create a new Native file system on the pool.

About this task

File Accesser Device Registrations appear at the bottom of the Manager UI home page. Users are directed to the Bulk Edit Device Site page after approval (per existing behavior). Users are then directed to the Bulk Edit Device Alias page (per existing behavior). Approved File Accesser devices appear in the left navigation pane. File Accesser Device Registrations appear at the bottom of the Manager UI home page.

A file system is the logical unit that encompasses a collection of file data and metadata transferred over the Native File Interfaces (currently NFSv3) by using shares. Each file system is mapped to a unique IBM Cloud Object Storage Vault where file data is stored. The **Settled Writes** function can enable a true **Active Archive** that is not subject to accidental or malicious modifications or deletes.

Procedure

1. Create Access Pool.

   Note: For optimal communication between Accesser and File Accesser devices, contact IBM® Support.
2. Create Storage Pool.

   Note: Work with IBM Support to create the appropriate **Vault Template** for your use case.
   - Using SecureSlice™ impacts performance.
   - Named Index is not used by File Accesser devices and must be disabled.
   - Recovery Listing is not used by File Accesser devices and should be disabled.
   - Create vault templates against Storage Pools that use Packed Storage.

3. Select **File Accesser** devices for this pool.

   Note: All selected File Accesser devices must be located in the same physical location, which are designated by Site. The Site designation is an important concept for File Accesser devices. All File Accesser devices at a particular Site participate in a unified metadata database cluster with each device that shares responsibility for storing portions of the metadata database. Metadata information is stored in a redundant fashion and can be on any of the File Accesser devices at a particular Site, regardless of File Server Pool membership. A minimum of three File Accesser devices must be included in a site.

   IBM Cloud Object Storage supports the use of two Sites along with File Accesser devices. Each site contains a complete set of metadata, replicated in each location defined by the devices in a Site. The amount of metadata storage at each location is identical and it is recommended that the same number of File Accesser devices be deployed at each of the Sites used.
4. Create **Vault Template**.

   Note: Associate Vault Template with created Access Pool.
5. Right-click in the left navigation pane and click **Create File Server Pool** to create a pool of File Accesser Devices. A minimum of three File Accesser devices should be deployed in the initial File Server Pool for operation in a production environment. Each device runs metadata services where metadata information is stored with three copies ensure data durability.

   Note: This requirement should be adhered to but is not enforced by the Manager.

   The Configure File Server Pool page (a grouping of file Accesser Devices) is displayed. Users can add File Accesser Devices, create File Systems, and create Shares.

   After File Server Pool creation, on the Monitor File Server Pool page, a message displays "One or more devices in this File Server Pool are still in the joining phase" for few minutes.

   Note: Contact customer support if the message does not disappear (how long it takes depends on the number of devices in the file server pool).

   The Create file system page is displayed. Recall that Vault Templates must exist.
6. Enter the **Name** (50 characters max) for the file system.

   The name cannot be blank and must be unique regarding other File Systems. Users should name the file system and select the wanted associated Vault Template. A Vault is created automatically by using the selected Vault Template.

   Note: The first character must not be a number, the entire name must be ASCII printable (character codes 32-127), and the name must be unique regarding other file systems in the Cloud Object Storage system.
7. Select whether **SSL** is to be used to communicate with the Vault. [Default = Enabled].

   By default **Use SSL** is selected. This option uses system-generated certificates over SSL for communication between the File Accesser and Accesser devices. This option provides the most secure method of communication between devices, but impacts communication performance. Clear this option for maximum performance.

   Note: If this option is cleared, **Anonymous Access** must be enabled on the vault that was created for the file system. Modify this setting under Vault configuration of the vault with the same name as the file system.
8. Select the **Vault Template** to be used for this file system.

   Note:

   The vault uses the name of the template. Only a single template can be selected. When you create a Vault template:

- **Packed Storage** vaults should be enabled.
- **Named Index** and **Recovery Listing** are not used by File Accesser devices and should be disabled.
- SecureSlice might slightly degrade performance, depending on Accesser hardware and work load.

9. Enable the **Settle Time**. [Default = Disabled] A system administrator can opt to define a **Settle Time**. The file/directory will become read only after the settle time passes.

   A user that matches an override UID or an override GID is able to edit the file/directory. If a settled file/directory is edited, the settle time is reset for the entity and its parent, and normal POSIX permissions apply until the settle time expires again.

   Important: The settle time cannot be changed or disabled after a file system is created, and cannot be enabled on existing file systems. If no settle time is provided, the function is disabled for the file system.

10. Enter the **UID Override** [Optional].

    Valid values are any positive integer and zero. A **Not Defined** value means that users in the GID Override group can bypass the settled writes function. The **UID Override** is a UID mapped to the user that can bypass the settled writes function.

    If the user is the user with the defined UID, then they are able to write/delete files, including settled files. A blank value means that any user with a matching override GID is able to bypass settled files.

11. Enter the **GID Override** [Optional].

    Valid values are any positive integer and zero. A **Not Defined** value means a user that matches the UID Override can bypass the settled writes function. The **GID Override** is a GID mapped to the set of users that can bypass the settled writes function.

    If the user is in the group with the defined GID, then they are able to write/delete files, including settled files. A blank value for a user that matches the override UID is able to bypass settled files.

12. Create a Share. The file system is created. Users must now enable the file system by creating a Share.
13. Click **Create Share** to display the Create Share page. Creating a “Share” adds an NFSv3 export to an existing file system.
14. Name the Share and select the **File System** in the drop-down menu that you want to associate with the Share. If you want the share to be “Read Only”, select this check box. If you want the Share to be read/write, leave it cleared.

    Note: Use Root is selected by default. This option creates a Share at the root of the file system. For a newly created file system, no other option is available. Shares that are created against existing File Systems with content can be created against any directory.

15. Locate directory for Share. If you clear the **Use Root** option, a directory browser is displayed. Use this browser to locate the directory in which to create the wanted Share. This option applies to Shares being created on file systems with existing directories.

    Note: If a user deletes a directory that was selected as a Share from a higher-level NFS mount, the share becomes invalid. In this circumstance, the Share must be removed from the Manager by an administrator.

16. Click **Save**. The share is now created and a user can now mount by using the IP of g1, g2, or g3 (File Accesser 1, 2, or 3).
17. Create shares of subdirectories.

What to do next

Users can now monitor the current state, view the configuration, and edit the configuration of File Accesser Devices.

Optional configuration

Explanation of some configuration options.

High Availability

The following steps are optional and create High Availability (HA) access to a Share via a virtual IP. High Availability configuration supports continued operation during the outage of one File Accesser device when more than three File Accesser devices are configured.

1. Under **HA Configuration**, enter a Virtual IP address. Multiple addresses are allowed in a comma-separated list.
2. Click **Update**. The added virtual IP is now listed on the File Accesser Configure Device page.
3. The Monitor File Server Pool page shows the status of the defined virtual IP.
4. Defined virtual IP addresses can now be used.

Note:

When new File Accesser devices are added to an existing File Server Pool cluster (with more than five devices), the HA inter-device communication system might timeout before status is properly reported for the entire cluster. If it occurs, you must restart the HA service on each device in the File Server Pool by using the following command.

```
$ service corosync stop; service corosync start; service pacemaker start
```

This procedure is only needed after the addition of a new node where the cluster status appears to not come back online after the new device addition.

Two sites for Disaster Recovery

Two sites can be configured for File Accesser Devices. The second site contains a mirror of all metadata stored within the primary site. The second site should have the same number of File Accesser Devices as the first site. The second site is not supported as an active use location and should be used for Disaster Recovery purposes during the technology preview.

When File Server Pools are created, the “Site” designation identifies all of the File Accesser devices that should participate in a single metadata cluster. A “Site” can have one or multiple File Server Pools that participates in the metadata cluster.

To create a second site for Disaster Recovery, include the same number of File Accesser devices as in the first site (some of all devices in each File Server Pool). All that is needed to enable the Disaster Recovery site is to create a File Server Pool (or multiple File Server Pools) with a second “Site” designation. The metadata Cluster software automatically handles the mirrored replication between the two sites.

Access Control

System administrators can provide a list of authorized systems into the Authorized Systems text area.

A blank list implies all systems can access the share. A non-empty list implies only the systems that are listed in the text box have access to the share.

The Manager application validates each authorized system in the list. An authorized system must be one of the following.

- An IPv4 address.
- A Classless Inter-Domain Routing string. For example, **192.168.56.101/24**.

- An IPV4 address with a subnet address. For example, **192.168.56.101/255.255.255.0**.
- A Fully Qualified Domain Name. For example, **usil.ibm.com**
- A string equal to "*****". A '*****' implies any system can access the share.
- A '*****' in domain names. A name with a '*****' is matched as a Posix basic regular expression. For example, ***.ibm.com**.

  Note: *.ibm.com matches **usil.ibm.com** or **host1.usil.com**.
- One or more '**?**' in the domain name. A domain name with one or more '**?**' is also matched as a Posix basic regular expression. For example,**usil?.ibm.com** matches **usi.ibm.com** or **usil.ibm.com**
- Character Classes ([..]) in domain names. For example, **[abc]-host.ibm.com**. It matches **a-host.ibm.com** or **b-host.ibm.com** or **c-host.ibm.com**
- '*****' or '**?**' or '**[]**' can be present in the same domain name multiple times.

Note: If an entry in the list contains '**?**', '*****', or '**[**', then the Manager application verifies that the authorized system string is a valid regular expression. It is so that entries such as ***.ibm.com**, **host?.ibm.com**, and **host[1-9].ibm.com** can be entered.

Note: Wildcards are not supported in IP/subnet addresses.

Note: Each entry must be 1 - 253 characters.

Note: Entries are case-insensitive.

Note: Each share can have 1000 allowed machine names.

Cloud Storage Object Access

Users can read their NFS written files with the Cloud Storage Object API. To set either anonymous or authenticated Cloud Storage Object Access, use the Cloud Storage Object Access section of the Configure Share page.

Remove a configuration

How to remove a configuration.

Removing Shares

If a Share is no longer needed, remove it by clicking **Configure** > **File Server Pool** for the wanted File Server Pool and clicking the Share name that you want to remove. You then can click **Delete Share**. When a share is removed, only the NFS export that is associated with the Share is deleted. No content on the file system is removed.

Removing File Systems

To remove a file system, follow theses steps.

1. Delete all Shares that are associated with the file system.
2. On the File Server Pool configuration screen, click the name of the file system you want to remove.
3. On the file system configuration screen, click **Delete File System**.

Note: If **Delete File System** is disabled, Shares are still associated with the file system that must first be removed.

Note: Deleting File Systems is a permanent action that cannot be recovered. Be sure that you no longer want to use the file system before you remove it.

Removing a File Accesser device from a File Server Pool

Contact IBM Support to remove a File Accesser device from a File Server pool. IBM Support can assist with migrating metadata from a functional File Accesser device before removal with minimal system disruption. Abrupt removal of a File Accesser device from a File Accesser Pool without the assistance of IBM Support might require a lengthy metadata redistribution and validation process.

Removal of multiple File Accesser devices requires working with IBM Support to ensure metadata integrity.

After a File Accesser device is removed from a File Server Pool, the device must be reimaged before reuse.

Removing a File Server Pool

It is imperative that the "nodetool decommission" command is completed on all devices in the pool before it is deleted. Deleting a file server pool is a permanent action. If File Systems exist where data is to be preserved, IBM Support must be involved in the File Server Pool removal process.

To delete a File Server Pool, follow these steps.

1. Remove all Shares within the File Server Pool.
2. On the File Server Pool page and click **Configure** > **Delete File Server Pool**.

This process removes the File Server Pool from the Manager configuration. All file systems must be deleted before you can delete the last File Server Pool.

Note: All Vaults that are associated with the File Server Pool (named by the file system names that are created within the File Server Pool) must be manually deleted to remove any remaining content. File Accesser devices that were a part of the File Server Pool must be reimaged before they are used in a new File Server Pool. If the file server pool has an associated seed node, and if another seed node is not defined elsewhere (in another file server pool), the file server pool cannot be deleted.

Use the Manager API to configure

API methods allow configuration of the File Accesser options.

Edit Device

Full path

`https://manager_host_or_ip/manager/api/format/1.0/editDevice.adm`

Valid formats

JSON, XML.

Capable roles

Super User, System Administrator.

Table 1. editDevice parameters

Parameter	Value	Description	Data type
id	required	ID of the device.	Long
alias		Alias to assign to the device.	String
description		Description to assign to the device.	String
siteId		ID of site on which to place the device.	Long
accessPoolId		Access Pool ID that the device should point to.	Long

Table 1. editDevice parameters (continued)

Parameter	Value	Description	Data type
driveFailureWarningCount		Quantity of drives that must be down for the device to report a warning level.	Integer
driveFailureErrorCount		Quantity of drives that must be down for the device to report an error level.	Integer
managementVaultId		Management Vault ID that the device should point to.	Long
advancedConfiguration		key=value pairs for various advanced configuration options to set for the device.	String
virtualIps		Comma or white space separated list of virtual IPs for the device, for use in HA configuration.	String
seedNode		Set to true if this device should be a seed node for other devices.	Boolean
tagIds		Tags that are currently associated with the device.	List
tagIdMap		Map of tag IDIDs to action, to add or remove the tag from the vault.	Map

Add or Remove device from File Server Pool

Full path

`https://`*`manager_host_or_ip`*`/manager/api/`*`format`*`/1.0/addOrRemoveDeviceFromFileServerPool.adm`

Valid formats

JSON, XML.

Capable roles

Super User, System Administrator

Table 2. addOrRemoveDeviceFromFileServerPool parameters

Parameter	Value	Description	Data type
id	required	The ID of the file server pool	Long
deviceId	required	The ID of the File Accesser device	Long
action		"add" to add a device and "remove" to remove a device from the file server pool.	String

Create File Server Pool

Full path

`https://`*`manager_host_or_ip`*`/manager/api/`*`format`*`/1.0/createFileServerPool.adm`

Valid formats

JSON, XML.

Capable roles

Super User, System Administrator

Table 3. createFileServerPool parameters

Parameter	Value	Description	Data type
name	required	Unique name for the file server pool.	String
deviceIds		A list of File Accesser devices to assign to the file server pool.	List

Create File System

Full path

`https://manager_host_or_ip/manager/api/format/1.0/createFileSystem.adm`

Valid formats

JSON, XML.

Implementation notes

On success, returns the Manager-generated ID of the created file system.

Capable roles

Super User, System Administrator

Table 4. createFileSystem parameters

Parameter	Value	Description	Data type
name	required	The name of the file system.	String
templateId	required	The ID of the vault or mirror template to use to back up the file system.	Long
templateType	vaultTemplate	The type of the template for which the ID is being provided.	String
updateAccessTimes	true (default)	Set to true if inodes in this file system should get their access times updated on read operations.	Boolean
maxBytes		The maximum size that is allowed for this file system.	Long
redundancyType	loadBalance (default)	The type of redundancy action for a file system when more than one Accesser device is specified.	String
storageType	vault (default)	The type of stable storage for the file system	String
sslEnabled	false (default)	If set to true, File Accesser devices use SSL to communicate with vaults.	Boolean
settleTime	Optional, default is empty	The defined settle time in milliseconds. If the value is empty, the settled writes functionality is disabled.	Long
uidOverride	default is empty	A UID mapped to the user who can bypass the settled writes functionality. Empty value prevents any user to bypass settled writes.	Long

Table 4. createFileSystem parameters (continued)

Parameter	Value	Description	Data type
gidOverride	default is empty	A GID mapped to the user's GID who can bypass the settled writes functionality. Empty value prevents all groups to bypass settled writes.	Long

Create Share

Full path

`https://manager_host_or_ip/manager/api/format/1.0/createShare.adm`

Valid formats

JSON, XML.

Implementation notes

On success, returns the Manager generated ID of the created share.

Capable roles

Super User, System Administrator

Table 5. createShare parameters

Parameter	Value	Description	Data type
fileServerPoolId	required	The ID of the file server pool where the share is created.	Long
name	required	The name of the share.	String
fileSystemId	required	The ID of the file system the created share exposes	Long
iNodeId		The iNode of the share.	Long
useRootINode	true (false)	If set to true, inodeId param is ignored and root inode of file system is used.	Boolean
requireSecurePort	false (default)	The accessing client must use a secure port (a port less than 1024).	Boolean
readOnly	false (default)	Set to true if this share should be read only.	Boolean
userMapping	noRootSquash (default)	The type of user mapping.	String
anonUid	0	The anonymous uid.	Long
anonGid	0	The anonymous gid.	Long

Edit Share Access Control

Full path

`https://manager_host_or_ip/manager/api/format/1.0/editShareAccessControl.adm`

Valid formats

JSON, XML.

Capable roles

Super User, System Administrator

Table 6. editShareAccessControl parameters

Parameter	Value	Description	Data type
id	required	A share ID.	Long
allowedMachineNames		A comma-separated list of ALL authorized systems that should have access to the share.	String

Edit Share Authorization

Full path

`https://manager_host_or_ip/manager/api/format/1.0/editShareAuthorization.adm`

Valid formats

JSON, XML.

Capable roles

Super User, Security Officer

Table 7. editShareAuthorization parameters

Parameter	Value	Description	Data type
id	required	A share ID.	Long
shareAnonymousPermission	required	The share permission to assign to anonymous users.	String
userPermissions	required	A map of account IDs to share permissions.	Long

Delete File Server Pool

Full path

`https://manager_host_or_ip/manager/api/format/1.0/deleteFileServerPool.adm`

Valid formats

JSON, XML.

Implementation notes

It is imperative that the 'nodetool decommission' command is completed on all devices in the pool before it is deleted. Deleting a file server pool is a permanent action.

Capable roles

Super User, System Administrator

Table 8. deleteFileServerPool parameters

Parameter	Value	Description	Data type
id	required	Manager generated ID of the object to delete.	Long
password	required	Password of the user that is making the request.	String

Delete File System

Full path

`https://manager_host_or_ip/manager/api/format/1.0/deleteFileSystem.adm`

Valid formats

JSON, XML.

Implementation notes

All data that is associated with the file system is lost.

Capable roles

Super User, System Administrator

Table 9. deleteFileSystem parameters

Parameter	Value	Description	Data type
id	required	Manager generated ID of the object to delete.	Long
password	required	Password of the user that is making the request.	String

Delete Share

Full path

`https://manager_host_or_ip/manager/api/format/1.0/deleteShare.adm`

Valid formats

JSON, XML.

Implementation notes

All data that is associated with the share is lost.

Capable roles

Super User, System Administrator

Table 10. deleteShare parameters

Parameter	Value	Description	Data type
id	required	Manager generated ID of the object to delete.	Long
password	required	Password of the user that is making the request.	String

Edit File Server Pool

Full path

`https://manager_host_or_ip/manager/api/format/1.0/editFileServerPool.adm`

Valid formats

JSON, XML.

Capable roles

Super User, System Administrator

Table 11. editFileServerPool parameters

Parameter	Value	Description	Data type
id	required	The ID of the file server pool.	Long
name		Unique name for the file server pool.	String
mcastAddr		mcastAddr of each device in the file server pool.	String
accessDeviceIps		This set of access device IPs overrides the known set of access devices.	String

Edit File System

Full path

`https://manager_host_or_ip/manager/api/format/1.0/editFileSystem.adm`

Valid formats

JSON, XML.

Implementation notes

Update the values that are associated with a file system.

Capable roles

Super User, System Administrator

Table 12. editFileSystem parameters

Parameter	Value	Description	Data type
id	required	The ID of the file system to be modified.	Long
name		The name of the file system	String
updateAccessTimes		Set to true if inodes in this file system should get their access times updated on read operations.	Boolean
maxBytes		The maximum size that is allowed for this file system.	Long
redundancyType		The type of redundancy action for a file system when more than one Accesser device is specified.	String
storageType		The type of stable storage for the file system	String
sslEnabled		If set to true, File Accesser devices use SSL to communicate with vaults.	Boolean
uidOverride	default is empty	A UID mapped to the user who can bypass the settled writes functionality. Empty value prevents any user to bypass settled writes.	Long
gidOverride	default is empty	A GID mapped to the user's GID who can bypass the settled writes functionality. Empty value prevents all groups to bypass settled writes.	Long

Edit Share

Full path

`https://manager_host_or_ip/manager/api/format/1.0/editShare.adm`

Valid formats

JSON, XML.

Capable roles

Super User, System Administrator

Table 13. editShare parameters

Parameter	Value	Description	Data type
id	required	The ID of the share.	Long
name		The name of the share.	String
readOnly		Set to true if this share should be read only.	Boolean
requireSecurePort		The accessing client must use a secure port (a port less than 1024).	Boolean
userMapping		The type of user mapping.	String
anonUid		The anonymous uid.	Long
anonGid		The anonymous gid.	Long

File System Directory List

Full path

`https://`*`manager_host_or_ip`*`/manager/api/`*`format`*`/1.0/fileSystemDirectoryView.adm`

Valid formats

JSON, XML.

Implementation notes

Returns the directories within a directory on a file system.

Capable roles

Super User, System Administrator

Table 14. fileSystemDirectoryView parameters

Parameter	Value	Description	Data type
iNodeId		The iNode of the directory whose subdirectories are listed. Leave blank for root iNode.	Long
fileSystemId	required	The ID of the file system to look in.	Long

IPMI Reboot Command Controller

Full path

`https://`*`manager_host_or_ip`*`/manager/api/`*`format`*`/1.0/ipmiReboot.adm`

Valid formats

JSON, XML.

Capable roles

Super User, System Administrator

Table 15. ipmiReboot parameters

Parameter	Value	Description	Data type
fileServerPoolId	required	The ID of a file server pool.	Long

Native File Interface log collection

Native File Interface related logs can be collected from a File Accesser device. The Native File Interface related log types are **cassandra**, **nfsfiler**, and **corosync**. For more information on log collection, see the log collection section of the *Manager Administration Guide*.

Chapter 8. Additional information

Limitations

A number of limitations apply to this procedure.

- Maximum individual file size limit is 1 TB.
- A maximum of 100 shares per file system is supported.
- A maximum of 1000 File Systems is permitted.
- A file system cannot expose two shares of the same name.
- The system does not support simultaneous writing to the same file from multiple clients. This action can cause data corruption in those files.
- The system supports File Accesser devices that are defined at two sites. This restriction is not enforced.
- Native File Interface does not support netgroups or integration with LDAP/NIS+ servers.
- Changing the IP of seed nodes, removing seed nodes, reimaging seed nodes, and turning off seed nodes results in undesired behavior.
- Host names and IPs of devices must not be changed.
- File Accesser devices should never be left offline for periods of time over 4 hours.
- HA recovery is triggered by outages at 10 seconds.
- A user cannot write an object via S3 and read by using NFS.
- The HA Virtual IP table on the Monitor File Server Pool page is only as recent as existing polling cycles allow.
- Native File Interface supports the failure of only a single device.
- All devices in a File Server Pool must be located in the same site.
- A device that is in a File Server Pool cannot have its site changed unless it is removed from the File Server Pool.
- The Cloud Object Storage Manager Database should be backed up after every Native File Interface-related configuration change. It includes but is not limited to file system or share creation and modification.

File Accesser Upgrades

File Accesser devices may be upgraded through the Manager as other IBM COS device types. It is strongly recommended that upgrades are performed during a maintenance window without active client traffic. Upgrading with active client traffic may result in I/O errors and unexpected application outages during the process. Upgrade procedure takes approximately 10 minutes per device.

File Accesser Failure Recovery

There are three failure scenarios that File Accesser instances cannot automatically recover from and require manual action:

- File Accesser experiences an unexpected hardware failure
- Services cannot start on a File Accesser after a reboot
- A File Accesser appeared unreachable for more than 3 hours

Attention: Contact IBM Customer Support execute the repair procedure on the File Accesser Devices after any of these events.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Accesser®, Cleversafe®, ClevOS™, Dispersed Storage®, dsNet®, IBM Cloud Object Storage Accesser®, IBM Cloud Object Storage Dedicated™, IBM Cloud Object Storage Insight™, IBM Cloud Object Storage Manager™, IBM Cloud Object Storage Slicestor®, IBM Cloud Object Storage Standard™, IBM Cloud Object Storage System™, IBM Cloud Object Storage Vault™, SecureSlice™, and Slicestor® are trademarks or registered trademarks of Cleversafe, an IBM Company and/or International Business Machines Corp.

Other product and service names might be trademarks of IBM or other companies.

Homologation statement

This product may not be certified in your country for connection by any means whatsoever to interfaces of public telecommunications networks. Further certification may be required by law prior to making any such connection. Contact an IBM representative or reseller for any questions.

Printed in USA