Oracle® Advanced Cluster File System

Administrator's Guide

23ai
F46807-02
May 2024

Oracle Advanced Cluster File System Administrator's Guide, 23ai

F46807-02

Primary Author: Eric Belden

Contributing Authors: Jerry Abraham, Don Bolinger, Chan Chan, Xiaobin Chen, Jonathan Creighton, Barb Glover, Fred Glover, Ricardo Gonzalez, Allan Graves, Diane Lebel, Colin McGregor, Valarie Moore, Bob Nelson, Charles Rusman, Joshua Smith, Malai Stalin, Elisabeth Thibault, Samarjeet Tomar, Brian Tsao, Anton Verhulst, Ric Werme

Contents

Glossary

Index

Preface

The *Oracle Advanced Cluster File System Administrator's Guide* describes how to administer Oracle Advanced Cluster File System (Oracle ACFS) for Oracle Database.

This Preface contains the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

The audience for this book includes system administrators, database administrators, and storage administrators. The *Oracle Advanced Cluster File System Administrator's Guide* is intended for database and storage administrators who perform the following tasks:

- Administer and manage Oracle ACFS
- Configure and administer Oracle ACFS

To use this document, you should be familiar with basic Oracle Database and Oracle ASM concepts and administrative procedures. Also, you might want to review the documentation for Oracle Clusterware and Oracle Real Application Clusters (Oracle RAC).

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Database Administrator’s Guide*

- *Oracle Database Concepts*
- *Oracle Database Reference*
- *Oracle Database SQL Language Reference*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- Platform-specific guides, including Oracle Database, Oracle Grid Infrastructure, and Oracle Real Application Clusters installation guides

See Also:

- *Oracle Database Licensing Information User Manual* to determine whether a feature is available on your edition of Oracle Database
- *Oracle Database New Features* for a complete description of the new features in this release
- *Oracle Database Upgrade Guide* for a complete description of the deprecated and desupported features in this release

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Oracle Advanced Cluster File System

Part I discusses Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM), in the following chapters:

1 Introducing Oracle ACFS and Oracle ADVM

Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM) provide key components of storage management.

This chapter describes the components of Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM).

This chapter provides concepts and an overview of Oracle ACFS and Oracle ADVM features with the following topics:

- Changes in Oracle ACFS and Oracle ADVM
- Overview of Oracle ACFS
- Understanding Oracle ACFS Concepts
- Understanding Oracle ACFS Administration
- Overview of Oracle ASM Dynamic Volume Manager

> **See Also:**
>
> - *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle Automatic Storage Management (Oracle ASM)
> - Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group attribute settings
> - Managing Oracle ACFS and Oracle ADVM With ASMCA for information about using Oracle ASM Configuration Assistant to administer Oracle ACFS
> - Managing Oracle ADVM with ASMCMD for information about volume management commands
> - Managing Oracle ACFS with Command-Line Tools for information about Oracle ACFS operating system utilities
> - Using Views to Display Oracle ACFS Information for information about using views to obtain Oracle ACFS information
> - *Oracle Automatic Storage Management Administrator's Guide* for information about the `ALTER DISKGROUP ADD VOLUME` SQL statement to administer volumes

Changes in Oracle ACFS and Oracle ADVM

The following are changes to Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM) in Oracle Database 23ai.

- Oracle ACFS encryption now provides integration with OCI Vaults. OCI Vault Master Encryption Keys can be used to encrypt the Volume Encryption Keys of encrypted ACFS file systems. This introduces an extra level to the ACFS Encryption key hierarchy and introduces key management capabilities within OCI.

- ACFS Encryption will now allow users to migrate from OCR to OKV. ACFS users now have a centralized point for key management using Oracle Key Vault.
- A new command, `acfsutil repl switchover`, provides coordinated failover services.
- A new tool is provided to configure SSH keys management for ACFS Replication.
- ACFS now supports an alternative transport choice for ACFS Replication which eliminates the need to maintain SSH related host and user keys.
- PDB Snapshot Carousel on ACFS will use ACFS file based snapshots to create sparse copies.
- ACFS replication now allows for primary clusters to replicate to standby cluster on a previous or older release.
- The new command `acfsutil plogconfig` offers users a way to manage persistent logging configuration settings.
- ACFS and AFD can now be deployed on ARM64 hardware.

These are deprecated features for Oracle ACFS 23ai.

- The `acfsutil compress` command is deprecated in Oracle Database 23ai.
- The `acfsutil snap remaster` command is deprecated in Oracle Database 23ai.

These are desupported features for Oracle ACFS 23ai.

- The `acfsutil repl reverse` command is desupported in Oracle Database 23ai. Use `repl failover` or `repl switchover` instead.

The following are changes to Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM) in Oracle Database 21c.

- New Features
- Desupported Features

See Also:

- *Oracle Database Licensing Information User Manual* to determine whether a feature is available on your edition of Oracle Database
- *Oracle Database New Features* for a complete description of the new features in this release
- *Oracle Database Upgrade Guide* for a complete description of the deprecated and desupported features in this release

Note:

In Oracle Automatic Storage Management (Oracle ASM) 21c, information about Oracle ASM is documented in the *Oracle Automatic Storage Management Administrator's Guide*.

New Features

These are new features for Oracle ACFS 21c.

- Oracle ACFS File Based Snapshots

  Oracle ACFS file based snapshots provides the ability to create snapshots of individual Oracle ACFS files in a space efficient manner on Linux.

  > **See Also:**
  >
  > – acfsutil fshare create
  >
  > – About Oracle ACFS Snapshots for an overview of Oracle ACFS snapshots

- Oracle ACFS Replication Unplanned Failover

  Oracle ACFS replication failover provides unplanned failover where the standby location assumes the role of the primary in case of failure.

  > **See Also:**
  >
  > – acfsutil repl failover
  >
  > – Oracle ACFS Replication for an overview of Oracle ACFS replication

- Oracle ACFS Automatic Shrinking

  Oracle ACFS automatic shrinking automatically shrinks an Oracle ACFS file system based on policy, providing there is enough free storage available in the volume.

  > **See Also:**
  >
  > – acfsutil size

- Oracle ACFS Mixed Sector Support

  Oracle ACFS mixed sector support enables the Linux primary and accelerator volumes of an Oracle ACFS file system to use a mix of different logical sector sizes, such as 512-bytes and 4096 bytes.

  > **See Also:**
  >
  > – mkfs for information about creating an accelerator volume
  >
  > – Oracle ACFS Accelerator Volume for an overview of accelerator volumes

Desupported Features

These are desupported features for Oracle ACFS 21c.

- Desupport of Oracle ACFS Replication Version 1

  Starting with Oracle ACFS 21c, Oracle ACFS replication protocol version 1 is desupported. Replication protocol version 1 has been replaced with snapshot-based replication version 2, introduced in Oracle ACFS 12c Release 2 (12.2.0.1).

- Desupport of Oracle ACFS Encryption on Solaris and Windows

  Starting with Oracle ACFS 21c, Oracle ACFS encryption is desupported on Solaris and Microsoft Windows operating systems.

- Desupport of Oracle ACFS Security (Vault) and ACFS Auditing

  Starting with Oracle ACFS 21c, Oracle ACFS Security (Vault) and ACFS Auditing are desupported.

- Desupport of Oracle ACFS on Microsoft Windows

  Starting with Oracle ACFS 21c, Oracle ACFS is desupported on Windows.

- Desupport of Oracle ACFS Remote

  Starting with Oracle ACFS 21c, Oracle ACFS on Member Clusters (ACFS Remote) is desupported.

- Desupport of Cluster Domain - Member Clusters

  Starting with Oracle Grid Infrastructure 21c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported.

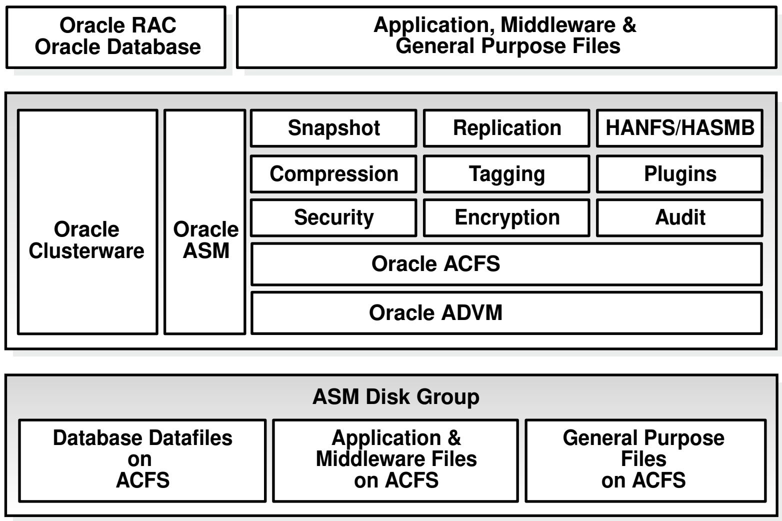
Overview of Oracle ACFS

Oracle Advanced Cluster File System (Oracle ACFS) is a multi-platform, scalable file system, and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support all customer files.

Oracle ACFS supports Oracle Database files and application files, including executables, database data files, database trace files, database alert logs, application reports, BFILEs, and configuration files. Other supported files are video, audio, text, images, engineering drawings, and all other general-purpose application file data. Oracle ACFS conforms to POSIX standards for Linux and UNIX.

An Oracle ACFS file system communicates with Oracle ASM and is configured with Oracle ASM storage, as shown in the following figure.

Figure 1-1 Oracle ACFS Storage Layers



Oracle ACFS leverages Oracle ASM functionality that enables:

- Oracle ACFS dynamic file system resizing
- Maximized performance through direct access to Oracle ASM disk group storage
- Balanced distribution of Oracle ACFS across Oracle ASM disk group storage for increased I/O parallelism
- Data reliability through Oracle ASM mirroring protection mechanisms

Oracle ACFS establishes and maintains communication with the Oracle ASM instance to participate in Oracle ASM state transitions including Oracle ASM instance and disk group status updates and disk group rebalancing. Oracle Automatic Storage Management with Oracle ACFS and Oracle ASM Dynamic Volume Manager (Oracle ADVM) delivers support for all customer data and presents a common set of Oracle storage management tools and services across multiple vendor platforms and operating system environments on both Oracle Restart (standalone) and cluster configurations.

Oracle ACFS is tightly coupled with Oracle Clusterware technology, participating directly in Clusterware cluster membership state transitions and in Oracle Clusterware resource-based high availability (HA) management. In addition, Oracle installation, configuration, verification, and management tools have been updated to support Oracle ACFS.

Oracle ACFS can be accessed and managed using native operating system file system tools and standard application programming interfaces (APIs). Oracle ACFS can also be managed with Oracle ASM Configuration Assistant. Oracle ACFS can be accessed using industry standard Network Attached Storage (NAS) File Access Protocols: Network File System (NFS) and Common Internet File System (CIFS).

In addition to sharing file data, Oracle ACFS provides additional storage management services including support for the Oracle Grid Infrastructure clusterwide mount registry, dynamic online file system resizing, and multiple space efficient snapshots for each file system.

Oracle ACFS contributes to the overall Oracle storage management by providing:

- A general-purpose standalone server and cluster file system solution that is integrated with Oracle ASM and Oracle Clusterware technologies
- A common set of file system features across multiple vendor platforms and operating systems, offering an alternative to native operating system or third-party file system solutions
- Standalone and clusterwide shared Oracle Database homes, all Oracle Database files, and application data
- Uniform, coherent shared file access and clusterwide naming of all customer application files
- Integration with Oracle Clusterware High Availability Resources

Oracle ACFS accommodates large storage capacities and large numbers of cluster nodes. It efficiently manages large numbers of file systems, files, and supports both small and large sized files with exabyte-capable file and file system capacities. Oracle ACFS provides optimized fast directory lookup for large directories with millions of files.

Oracle ACFS provides support for sparse files. Oracle ACFS sparse files greatly benefit NFS client write operations which are commonly received out of order by the NFS server and the associated Oracle ACFS file system. Usually when an application writes beyond the end of file, storage is allocated and zeroes inserted beyond the old end of file and the beginning of the new. With this feature, a hole remains in the file instead of the inserted zeroes. Oracle ACFS then fills these holes with zeroes in memory when the holes are read. The sparse files feature benefits NFS performance and also the performance and disk utilization of other applications that intentionally perform this type of writing. In addition, there are also reduced time and storage benefits for files that are inherently sparse, meaning they have a lot of unused space, such as some image files for virtual machines. For sparse file support, the `COMPATIBLE.ADVM` disk group attribute must be set to `12.2` or greater.

Oracle ACFS file systems are generally mounted on all cluster nodes to deliver a single name space for the cluster so that each node maintains the same view and access capabilities to the mounted file systems. In the event of a member failure, another cluster member quickly recovers any outstanding metadata transactions on behalf of the failed member. Following recovery, access by other active cluster members and any remote client systems can resume.

The following list provides important information about Oracle ACFS:

- It is recommended to change your ADVM compatibility to the running release whenever possible for the latest performance benefits. This is not the default.
- For all applications, Oracle ACFS performance is best with larger write() sizes, such as 8 K or larger.
- For best performance with non-spinning disks, Oracle recommends setting the I/O scheduler to 'none'. For best performance with spinning disks, set the I/O scheduler to to 'mq-deadline' when available, otherwise 'none'.
- Oracle ACFS does not support any files associated with the management of Oracle ASM, such as files in the Oracle Grid Infrastructure home and in the Oracle ASM diagnostic directory.
- Oracle ACFS does not support Oracle Cluster Registry (OCR) and voting files.
- Oracle ACFS functionality requires that the disk group compatibility attributes for `ASM` and `ADVM` be set to `11.2` or higher.

- To use an Oracle ACFS file system for an Oracle Database home, the release level must be Oracle 11*g* Release 2 (11.2) or later.

Note:

The following features are desupported in Oracle ACFS 21c:

- Desupport of Oracle ACFS Replication Version 1

  Starting with Oracle ACFS 21c, Oracle ACFS replication protocol version 1 is desupported. Replication protocol version 1 has been replaced with snapshot-based replication version 2, introduced in Oracle ACFS 12c Release 2 (12.2.0.1).

- Desupport of Oracle ACFS Encryption on Solaris and Windows

  Starting with Oracle ACFS 21c, Oracle ACFS encryption is desupported on Solaris and Microsoft Windows operating systems.

  Oracle ACFS Encryption on Oracle Solaris and Microsoft Windows is based on RSA technology. Retirement of RSA technology has been announced. Oracle ACFS Encryption continues to be supported on Linux, and is unaffected by this deprecation, because Linux uses an alternative technology.

- Desupport of Oracle ACFS Security (Vault) and ACFS Auditing

  Starting with Oracle ACFS 21c, Oracle ACFS Security (Vault) and ACFS Auditing are desupported.

  Desupporting cluster features with limited adoption allows Oracle to focus on improving core scaling, availability, and manageability across all features and functionality. Oracle ACFS Security (Vault) and ACFS Auditing are desupported.

- Desupport of Oracle ACFS on Microsoft Windows

  Starting with Oracle ACFS 21c, Oracle ACFS is desupported on Windows.

  Oracle ACFS is used for two major use cases: Oracle Database Files for Oracle Real Application Clusters (Oracle RAC) and generic files (unstructured data) that need to be shared across multiple hosts. For Oracle Real Application Clusters files, Oracle recommends that you use Oracle ASM. For generic files, depending on your use case, Oracle recommends that you either move files to Oracle Database File System (DBFS), or move files to Microsoft Windows shared files.

- Desupport of Oracle ACFS Remote

  Starting with Oracle ACFS 21c, Oracle ACFS on Member Clusters (ACFS Remote) is desupported.

  Oracle ACFS on Member Clusters (ACFS Remote) is desupported. Desupporting certain clustering features with limited adoption allows Oracle to focus on improving core scaling, availability, and manageability across all features and functionality. Deprecating certain clustering features with limited adoption allows Oracle to focus on improving core scaling, availability, and manageability across all features and functionality.

- Desupport of Cluster Domain - Member Clusters

  Starting with Oracle Grid Infrastructure 21c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported.

Desupporting certain clustering features with limited adoption allows Oracle to focus on improving core scaling, availability, and manageability across all features and functionality. Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. While Member Clusters were first introduced to simplify the management of larger cluster estates and minimize outage times for certain failures and configurations, additional enhancements in standalone clusters meanwhile provide the same benefits, making the use of Member Clusters unnecessary. Consequently, customers currently using Member Clusters are best advised to use Standalone Clusters going forward.

See Also:

- Overview of Oracle ASM Dynamic Volume Manager for an overview of Oracle ADVM
- Oracle Clusterware Resources and Oracle ACFS Administration for information Oracle Clusterware High Availability Resources
- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- Managing Oracle ACFS with Command-Line Tools and Using Views to Display Oracle ACFS Information for information about managing and monitoring Oracle ACFS

Understanding Oracle ACFS Concepts

This section describes concepts for the key Oracle ACFS components and contains the following topics:

- About Oracle ACFS
- About the Oracle ACFS Mount Model and Namespace
- About Oracle ACFS and Database Data Files
- About Oracle ACFS and Oracle Database Homes
- About Oracle ASM Dynamic Volume Manager
- About the Oracle ACFS Driver Model
- About the Oracle ACFS Mount Registry
- About Oracle ACFS Snapshots
- About Oracle ACFS and Backup and Restore
- About Oracle ACFS Integration with Oracle ASM

About Oracle ACFS

Oracle ACFS is designed as a general-purpose, standalone server and clusterwide file system that delivers support for all customer files. Users and applications can access and manage Oracle ACFS using native operating system file system application programming interfaces (APIs) and command-line interface (CLI) tools. Users can also manage Oracle ACFS with Oracle ASM Configuration Assistant (ASMCA).

Oracle ACFS supports large files with 64-bit file and file system data structure sizes leading to exabyte capable file and file system capacities on 64 bit platforms. Variable extent-based storage allocation and high-performance directories contribute to fast performance and shared disk configurations that provide direct storage paths to Oracle ACFS file data from each cluster member. File system integrity and fast recovery is achieved with Oracle ACFS metadata checksums and journaling. Oracle ACFS is designed as a multi-node, shared file system model that delivers coherent, cached, direct storage paths to Oracle ACFS file data from each cluster member.

Oracle ACFS files systems are typically configured for clusterwide access. File systems, files, and directories are visible and accessible from all cluster members and can be referenced by users and applications using the same path names from any cluster member. This design enables simplified application deployments across cluster members and facilitates both multiple instance cluster applications and high availability (HA) failover of unmodified standalone server applications.

Oracle ACFS presents single system file access semantics across cluster configurations. Applications and users on all cluster members are always presented with the same view of shared Oracle ACFS file data, supported by the Oracle ACFS clusterwide user and metadata cache coherency mechanism.

About the Oracle ACFS Mount Model and Namespace

Oracle ACFS is designed as a hierarchical file system containing files and subdirectories organized into a tree-structured namespace with files at the leaves. The namespace design is a single-file system naming model for both standalone server and cluster configurations. This design enables each cluster member to present shared files to cluster applications using the same path names, simplifying multi-node application and user access, and overall file system administration. The Oracle ACFS mount model also accommodates node local mounts and cluster node subset mounts in cluster configurations to accommodate additional customer requirements.

It is best practice for Oracle ACFS file systems to be Oracle Clusterware managed with Oracle Clusterware resources to ensure they are properly handled during Oracle Grid Infrastructure startup and shutdown.

You can explicitly use the `mount` command. However, if the resource has been created, then the file system may already be mounted.

About Oracle ACFS and Database Data Files

> **Note:**
>
> The best practice for data files in Oracle ACFS is to use a clusterwide resource for the Oracle ACFS File system that stores the data files.
>
> If the data files are added after the database installation is complete, you must modify the database dependencies to list the new Oracle ACFS file system. Failure to do so results in errors with application reliability. To specify the file systems in the dependency list, use the SRVCTL database object command to modify the Oracle ACFS paths used by the resource.

Oracle ACFS in Oracle Grid Infrastructure 18c cluster and standalone (Oracle Restart) configurations supports all database files for Oracle Database releases 11.2.0.4 or higher. Oracle ACFS can be configured for use with the database particularly to leverage Oracle ACFS snapshots for database testing and development. To support database files in cluster configurations, the `COMPATIBLE.ADVM` attribute must be set to `12.1` or higher for the disk group that contains the Oracle ACFS file system. In an Oracle Restart configuration, the `COMPATIBLE.ADVM` attribute must be set to `12.2.0.1` or higher to support all database files.

Support for database data files on Oracle Exadata (Linux) begins with Oracle Grid 12*c* Release 1 (12.1.0.2). However, Oracle ACFS does not currently have the ability to push database operations directly into storage.

Oracle ACFS on Oracle Grid Infrastructure 12.1.0.2 additionally supports all database files for Oracle Database 10*g* Release 2 (10.2.0.4 and 10.2.0.5) on Oracle Exadata (Linux) storage. For database file support with Oracle Database 10*g* Release 2 (10.2.0.4 and 10.2.0.5) on Oracle Exadata storage, the following conditions must be met:

- When creating an Oracle Database with DBCA, you must set the `REMOTE_LISTENER` initialization parameter to *`your_scan_vip`*`:1521` otherwise DBCA fails during the create process.
- You must modify all the start and stop dependencies of the database instance resources to ensure that the resources start when starting Oracle Clusterware.

The following list provides important information about using Oracle ACFS with database files:

- Oracle ACFS support includes all file types supported by Oracle ASM.
- When storing database data files on Oracle ACFS, you must set the `FILESYSTEMIO_OPTIONS` initialization parameter to `setall`; other settings are not supported. To achieve optimal performance with database data files, set `ASM` and `ADVM` compatibility attributes to `12.1` or higher for the disk group that contains the Oracle ADVM volume intended to hold the data files. For volumes created before 12.1.0.2, set the stripe columns to `1`, or set the stripe columns to `8` and the stripe width to `1 MB`. Volumes created while running 12.1.0.2 or higher already default to the high performance configuration (stripe columns = `8` and stripe width = `1 MB`).
- To obtain optimal database performance with snapshots, the snapshots must be created after the `ADVM` compatibility attribute is set to `12.1` or higher.

- Use a 4 K or larger database block size and tablespace block size with Oracle ACFS for best performance.
- For best performance when using Oracle Exadata, use a 4 K metadata file system.
- Oracle ACFS does not support databases with 2 K block sizes.
- Do not export Oracle ACFS file systems containing database files through NFS to access them from a DNFS client. This configuration is not supported.
- If a data file is configured to automatically extend, then the size of the increments should be large enough to ensure that the extend operation occurs infrequently. Frequent automatic extends have a negative performance impact.
- Running a workload in a snapshot reduces resources for the primary workload running on the base files because the storage is shared between the base file system and the snapshots. To run test scenarios in Oracle ACFS snapshots without impacting the primary workload, copy the file system and then run test workloads on snapshots created in the copied file system.
- Oracle ACFS does not support encryption or replication with Oracle Database data files, tablespace files, control files, redo logs, archive logs, RMAN backups, Data Pump dumpsets or flashback files. To encrypt database data files on Oracle ACFS, Oracle recommends Oracle Advanced Security. Oracle Advanced Security provides Transparent Data Encryption (TDE) to encrypt data files for entire tablespaces. Oracle Data Guard and Oracle GoldenGate provide other replication options for database files on Oracle ACFS.
- Databases that share storage with snapshots or with the base of the file system can become fragmented under active online transaction processing (OLTP) workloads. This fragmentation can cause the location of the data in the volume to be discontiguous for sequential scans. Fragmentation is reported through the `acfsutil defrag dir` and `file` commands, and it can also be viewed with the `acfsutil info file` command. The `acfsutil defrag dir` and `file` commands also enable on-demand defragmentation of a file.
- When a database file is located inside an ACFS snapshot, or when a database file is an fshare, RMAN can be used to back up these database files as a sparse backup. When you choose to perform a sparse backup of such a database file, the operation will copy only the data in the snapshot or fshare which is changed from the corresponding file in the source of the snapshot or fshare. The file in the source of the snapshot or fshare must be unchanged since the snapshot or fshare was created. The file being backed up can be either in an ACFS snapshot or be an fshare, but not both. See the RMAN documentation for more details about sparse backup and restore.

> **Note:**
>
> Oracle ACFS NAS/MAX is supported as a dNFS target for RMAN backup sets. This support begins with Oracle ACFS 19c release. This allows the HA-NFS functionality to be used to provide scaling and highly available dNFS backup targets for RMAN. Other stated restrictions in this guide, such as the lack of support for encryption or replication are still valid.

- When creating a PDB snapshot copy on ACFS with the `clonedb` initialization parameter set to `FALSE` (the default), the source PDB will be cloned using ACFS snapshots. ACFS supports two different types of snapshots - traditional snapshots of the whole file system, and newer per-file snapshots. PDB snapshot copy will use per-file snapshots when possible. In order to use per-file snapshots, the ACFS release must be 21 or higher, the ACFS compatibility must be 21 or higher, and the RDBMS release must be 19.12 or higher. When using ACFS snapshots the database files will be symlinks pointing into the ACFS snapshot directory. When using per-file snapshots, the snapshot files look like regular files to the user. The `acfsutil info file -f` command can be used to identify these files as snapshots. The output will contain the line "`fshare: Yes`".

See Also:

- Creating Oracle ACFS Resources for information about Oracle ACFS resources
- volcreate for information about creating a volume
- mkfs for information about setting the metadata block size with `mkfs -i`
- Oracle ACFS Command-Line Utilities for information about `acfsutil defrag` and `acfsutil info` commands
- *Oracle Database Reference* for information about the `REMOTE_LISTENER` initialization parameter
- *Oracle Automatic Storage Management Administrator's Guide* for a list of file types supported by Oracle ASM
- *Oracle Clusterware Administration and Deployment Guide* for information about resource dependencies
- *Oracle Database Advanced Security Guide* for information about Transparent Data Encryption (TDE)
- *Oracle Data Guard Concepts and Administration* for information about other replication options for database files on Oracle ACFS
- https://www.oracle.com/middleware/technologies/goldengate.html for Oracle GoldenGate documentation about other replication options for database files on Oracle ACFS

About Oracle ACFS and Oracle Database Homes

An Oracle ACFS file system can be configured as an Oracle Database home.

When installing a database in a cluster, a shared Oracle ACFS file system can be used for the database home. You can use an Oracle ACFS file system for an Oracle Database home with Oracle 11*g* Release 2 (11.2) or later.

When installing Oracle Software, there must be a separate Oracle base (`ORACLE_BASE`) associated with each operating system user. For example, there should be a separate Oracle base for a grid user and a database user.

You can locate the Oracle Database base (`ORACLE_BASE` for database) directory and home (`ORACLE_HOME` for database) directory on an Oracle ACFS file system. The Oracle Database base (`ORACLE_BASE` for database) directory should not be the Oracle Grid Infrastructure base (`ORACLE_BASE` for grid) directory or should not be located under the Oracle Grid Infrastructure base directory (`ORACLE_BASE` for grid).

The Oracle Grid Infrastructure base (`ORACLE_BASE` for grid) directory and home (`ORACLE_HOME` for grid) directory cannot be located on the Oracle ACFS file system because the Oracle ACFS file system cannot be created until Oracle Grid Infrastructure is installed.

One or more Oracle Database homes on Oracle ACFS can be created under the same mount point with each home using a separate Oracle ACFS file system.

After the installation of Oracle Grid Infrastructure Software and before the installation of the Oracle Database software with Oracle Universal Installer (OUI), you can create an Oracle ACFS file system to be configured for use as an Oracle Database home.

You can also use the Oracle ASM Configuration Assistant (ASMCA) or Oracle ACFS commands to create the file system.

> **Note:**
>
> When an Oracle ACFS file system contains an Oracle Database home or Oracle Database uses the file system for any file storage, the file system must have an Oracle ACFS file system resource. If you have not used Oracle ASM Configuration Assistant to setup the mount point, then you must use Server Control Utility (SRVCTL) commands to set up Oracle Database dependencies.

In an Oracle Grid Infrastructure clusterware configuration, a clusterwide Oracle ACFS resource is required when using Oracle ACFS for the database home. To enable the database owner to act on the resource, the owner must be specified as a permitted user when creating the resource. You can specify the database owner with the `-u` option of the `srvctl add filesystem` or `acfsutil registry` command. Root privilege is required when adding the resource in Linux or Unix environments.

After the Oracle ACFS file system and resources are created, the Oracle ACFS-based database home mount point location can be selected as the Oracle Database home location by browsing to and then choosing the directory during the Oracle Universal Installer (OUI) Database Software installation.

You can use the `srvctl start filesystem` command to manually mount the Oracle ACFS file system.

> **Note:**
>
> When additional Oracle ACFS file systems are added to an Oracle Database after creation, they must be specified in the Oracle Database Resource Dependency list. Failure to do so results in errors with application reliability. To specify the file systems in the dependency list, use the SRVCTL database object command to modify the Oracle ACFS paths used by the resource.

Oracle ACFS file systems can be also configured for use as a home for applications. However, Oracle ACFS file systems cannot be used for an Oracle base directory or an Oracle Grid Infrastructure home that contains the software for Oracle Clusterware, Oracle ASM, Oracle ACFS, and Oracle ADVM components.

To reduce contention on an Oracle ACFS file system in an Oracle RAC environment where the Oracle Database home is shared on Oracle ACFS, Oracle Database auditing operating system files should be configured as node specific. For a node-specific setup, you must ensure that the `AUDIT_FILE_DEST` initialization parameter in the configuration file of each database instance points to a unique location rather than one location for all the database instances.

For example, if you have a database with the Oracle name set to `TEST` and you want to ensure that the location of `AUDIT_FILE_DEST` initialization parameter for each database instance, such as `TEST1` or `TEST2`, points to a node specific location for that instance, you can run the following SQL statement:

```
SQL> ALTER SYSTEM SET AUDIT_FILE_DEST='$ORACLE_BASE/admin/adump/TEST/@'
     SCOPE=SPFILE SID='*';
```

In the previous example, `@` expands to the `ORACLE_SID` of each instance. If `ORACLE_BASE` has been set to `/acfsmounts` in this example, then that value could have been used in place of the `ORACLE_BASE` variable.

See Also:

- Managing Oracle ACFS with Command-Line Tools for information about using Oracle ACFS commands to create a file system
- Creating an Oracle ACFS File System for Database Use for information about using ASMCA
- Creating Oracle ACFS Resources for additional information Oracle ACFS resources
- *Oracle Clusterware Administration and Deployment Guide* and *Oracle Real Application Clusters Administration and Deployment Guide* for information about SRVCTL commands
- *Oracle Database Installation Guide* for your environment for information about the installation of Oracle Database software and storage options and about Optimal Flexible Architecture (OFA) recommendations for Oracle base and home directories
- *Oracle Grid Infrastructure Installation Guide* for your environment for information about the installation of Oracle Grid Infrastructure software and storage options
- *Oracle Database Administrator's Guide* for information about using Oracle Managed files
- *Oracle Database Reference* for information about the `AUDIT_FILE_DEST` initialization parameter

About Oracle ASM Dynamic Volume Manager

The Oracle ASM Dynamic Volume Manager (Oracle ADVM) provides volume management services and a standard disk device driver interface to clients. File systems and other disk-based applications send I/O requests to Oracle ADVM volume devices as they would to other storage devices on a vendor operating system.

For more information about Oracle ADVM, refer to Overview of Oracle ASM Dynamic Volume Manager.

About the Oracle ACFS Driver Model

An Oracle ACFS file system is installed as a dynamically loadable vendor operating system (OS) file system driver and tool set that is developed for each supported operating system platform. The driver is implemented as a Virtual File System (VFS) and processes all file and directory operations directed to a specific file system.

> **Note:**
>
> Errors encountered by the drivers are written to the native operating system console and system event loggers. Refer to Understanding Oracle ACFS I/O Failure Console Messages.

About the Oracle ACFS Mount Registry

The Oracle ACFS mount registry supports Oracle Grid Infrastructure cluster configurations, but does not support Oracle Restart configurations.

File systems that are to be mounted persistently (across restarts) can be registered with the Oracle ACFS mount registry. In cluster configurations, registered Oracle ACFS file systems are automatically mounted by the mount registry, similar to a clusterwide mount table. However, in Oracle Restart configurations the automatic mounting of registered Oracle ACFS file systems is not supported.

By default, an Oracle ACFS file system that is inserted into the Oracle ACFS mount registry is automatically mounted on all cluster members, including cluster members that are added after the registry addition. However, the Oracle ACFS mount registry also accommodates standalone and multi-node (subset of cluster nodes) file system registrations. The mount registry actions for each cluster member mount only registered file systems that have been designated for mounting on that member.

The Oracle ACFS mount registry is implemented using Oracle Clusterware resources, specifically the Oracle ACFS resource. The Oracle ACFS resource actions are designed to automatically mount a file system only one time for each Oracle Grid Infrastructure initialization to avoid potential conflicts with administrative actions to dismount a given file system.

See Also:

- Oracle ACFS and Oracle Restart for more information about Oracle ACFS and Oracle Restart
- acfsutil registry for information about registering an Oracle ACFS file system using the `acfsutil` command
- Oracle Clusterware Resources and Oracle ACFS Administration for more information about the implementation of the mount registry and Oracle ACFS resources
- *Oracle Clusterware Administration and Deployment Guide* and *Oracle Real Application Clusters Administration and Deployment Guide* for information about SRVCTL commands, including the `srvctl add filesystem` command
- *Oracle Clusterware Administration and Deployment Guide* for information about Oracle Clusterware resources

About Oracle ACFS Snapshots

An Oracle ACFS snapshot is an online, read-only or read-write, point in time copy of an Oracle ACFS file system.

The snapshot copy is space-efficient and uses Copy-On-Write functionality. Before an Oracle ACFS file extent is modified or deleted, its current value is copied to the snapshot to maintain the point-in-time view of the file system.

Oracle ACFS snapshots are immediately available for use after they are created. The snapshots are created in the `.ACFS/snaps/` directory of the file system. They are always online while the file system is mounted. Consequently, an Oracle ACFS snapshot can support the online recovery of files inadvertently modified or deleted from a file system. An Oracle ACFS snapshot can also be used as the source of a file system backup, as it can be created on demand to deliver a current, consistent, online view of an active file system.

Up to 1023 read-only, read-write, or combination of read-only and read-write snapshot views are supported for each file system, enabling flexible online file recovery solutions spanning multiple views to be employed. On 64-bit systems, Oracle ACFS supports 1023 snapshots. The total number of snapshots can be any combination of read-only and read-write snapshots. To support the creation of more than 63 snapshots, the disk group compatibility attribute for `ADVM` must be set to `12.1.0.2` or greater. Also, the following conditions must be met to create more than 63 snapshots.

- All snapshots of the file system that were created with `COMPATIBLE.ADVM` set to less than `12.1.0.2` must be deleted.
- All snapshots of the file system that were created after setting `COMPATIBLE.ADVM` to `12.1.0.2` or greater while older snapshots existed must be deleted.

Oracle ACFS read-write snapshots enable fast creation of an snapshot image that can be both read and written without impacting the state of the Oracle ACFS file system hosting the snapshot images. You can use read-write snapshots for:

- Testing of new versions of application software on production file data reflected in the read-write snapshot image without modifying the original production file system

- Running test scenarios on a real data set without modifying the original production file system

To use Oracle ACFS read-write snapshots, the disk group compatibility attribute for `ADVM` must be set to `11.2.0.3.0` or higher. If you create a read-write snapshot on an existing Oracle ACFS file system from a version earlier than `11.2.0.3.0`, then the file system is updated to the `11.2.0.3.0` or higher format. After a file system has been updated to a higher version, an Oracle ACFS file system cannot be reverted to an earlier version, and accordingly cannot be mounted on an earlier Oracle Grid Infrastructure version.

You can create a snapshot from an existing snapshot in the same Oracle ACFS file system. In addition, you can convert a snapshot between read-only and read-write formats. To create from an existing snapshot or convert a snapshot, the disk group compatibility attribute for `ADVM` must be set to `12.1` or higher. In addition, creation from an existing snapshot is not permitted if there are:

- Any snapshots present in the file system that were created with the `ADVM` compatibility set to less than 12.1
- Any snapshots of the file system that were created after `ADVM` compatibility was set to `12.1` but while 11.2 snapshots existed

Oracle ACFS snapshot storage is maintained within the file system, eliminating the management of separate storage pools for file systems and snapshots. Oracle ACFS file systems can be dynamically resized to accommodate additional file and snapshot storage requirements.

You cannot modify encryption metadata in read-write snapshots except for enabling or disabling encryption. If a file is not encrypted in the snapshot, that file cannot be encrypted by encrypting the corresponding file in the active file system.

Files in a read-write snapshot can be encrypted, decrypted, or rekeyed if the operation target is a path specified for a file or directory of the read-write snapshot. However, if an encryption, decryption, or rekey operation is specified at the file system level, then the operation does not process files and directories of snapshots in the `.ACFS/snaps/` directory.

All Oracle ACFS snapshot operations are serialized clusterwide in the kernel. For example, if a snapshot create operation is initiated at the same time as a snapshot delete operation, then both operations would complete, but they would not run in parallel inside of the kernel. One operation would complete before the other was started.

The `acfsutil snap duplicate` commands can be invoked manually to enable a target file system or writable snapshot to track the evolution of a source file system or snapshot with minimal overhead. Oracle ACFS updates the target after each apply operation to indicate its current contents, and checks before starting the next apply operation to ensure that the starting point of the incoming snapshot duplication stream matches the endpoint of the last-applied stream. The target remains writable by any and all applications even when it is in use by `acfsutil snap duplicate apply` . Oracle ACFS determines any updates to the target by a process other than the apply process, and if such an update occurs, invalidates the internal metadata that track the target contents. This prevents further apply operations to the target, and avoids the possibility of a future apply operation yielding unexpected results.

As of the 23ai release, the acfsutil snap duplicate commands may be used to create and restore a backup of a file system plus all of its snapshots, while preserving the sparseness of the snapshots in the backup. In this kind of sparse backup, any blocks shared in the original ACFS volume between snapshots, or between the file system and snapshots, will still be shared in the backup. A sparse backup is always a full backup. There is no provision for performing an incremental backup.

The use of the `acfsutil snap duplicate` commands interacts with Oracle ACFS replication operations. A target file system in use by `acfsutil snap duplicate apply` cannot be specified for use as a file system for replication. And a file system in use by replication cannot be specified as the target for `acfsutil snap duplicate apply`.

To run the `acfsutil snap duplicate` commands, the disk group compatibility attribute for `ADVM` must be set to at least `12.2`.

Additional `acfsutil snap` commands are available to manage duplicate snapshots and remaster an Oracle ADVM volume with an existing snapshot.

Oracle ACFS snapshots are administered with the `acfsutil snap` commands.

> **Note:**
>
> The `link()` and `rename()` system calls fail if an attempt is made to link or rename a file in the Oracle ACFS file system and a file in any associated read-write snapshot, or vice versa. Any tools which use the `link()` and `rename()` system calls, such as `ln` and `mv`, also fail in the same scenario.

> **See Also:**
>
> - acfsutil fshare create for information about creating a shared file (fshare) that functions as a file based snapshot
> - acfsutil snap create for additional information on the target used with the `acfsutil snap duplicate` command
> - Oracle ACFS Replication for information about Oracle ACFS replication
> - Oracle ACFS Encryption for information about Oracle ACFS encryption
> - Oracle ACFS Command-Line Tools for Snapshots for information about the `acfsutil snap` commands
> - acfsutil snap duplicate create for additional information about sparse backups.

About Oracle ACFS and Backup and Restore

Oracle ACFS runs on operating system platforms as a native file system technology supporting native operating system file system application programming interfaces (APIs). Consequently, backup applications that access files using the native operating system file system interfaces are able to access and backup Oracle ACFS file systems and other native operating system file systems. Oracle ACFS snapshots can be dynamically created and used to present a consistent, on-line view of an active file system to a backup application.

Backup applications that use interfaces other than the standard operating system interfaces (read or write) are not supported with Oracle ACFS.

> **Note:**
>
> When Oracle ACFS is used with Oracle Recovery Manager (RMAN) backups or restore, you should tune RMAN to improve performance because the default values may not be ideal for Oracle ACFS file systems. For example, you can adjust RMAN parameters to match the buffer size with the volume stripe width or Oracle ASM AU size. Also, you can adjust the number of buffers to match the number of devices in the Oracle ASM disk group.

> **See Also:**
>
> - Oracle ACFS Tagging for information about using common operating system utilities to preserve Extend Attributes for tagging definitions
> - *Oracle Automatic Storage Management Administrator's Guide* for information about `md_backup` and `md_restore` for backing up and restoring Oracle ACFS metadata
> - My Oracle Support (`https://support.oracle.com`) for articles about tuning RMAN tuning, such as *RMAN Performance Tuning Using Buffer Memory Parameters* (Doc ID 1072545.1) and *RMAN Performance Troubleshooting* (Doc ID 1326686.1)

About Oracle ACFS Integration with Oracle ASM

Oracle ACFS is always configured with Oracle ASM storage and interfaces with Oracle ASM storage through a traditional device file. This device file is presented by Oracle ADVM and is constructed using a dynamic volume file. The Oracle ADVM volume device file is created automatically following the creation of an Oracle ADVM volume. An Oracle ACFS file system is then bound to the Oracle ADVM device file during the file system creation.

After an Oracle ACFS is configured and mounted, the file system inherits the Oracle ASM storage management features associated with an Oracle ADVM volume, including dynamic balanced distribution, mirroring and striping, and dynamic resizing.

The Oracle ACFS driver establishes communication with the Oracle ASM instance to receive Oracle ASM status information including Oracle ASM instance and disk group state transitions. However, I/O does not go through Oracle ASM nor through the Oracle ASM proxy, but rather goes directly to the underlying Oracle ASM storage.

For information about Oracle ACFS and Oracle ASM operations, refer to Oracle ACFS and Dismount or Shutdown Operations.

Understanding Oracle ACFS Administration

This section describes Oracle ACFS administration and contains the following topics:

- Oracle ACFS and File Access and Administration Security
- Oracle ACFS and Grid Infrastructure Installation
- Oracle ACFS Configuration
- Oracle ACFS Features Enabled by Compatibility Attribute Settings
- Oracle Clusterware Resources and Oracle ACFS Administration
- Oracle ACFS and Dismount or Shutdown Operations
- Oracle ACFS Encryption
- Oracle ACFS Compression
- Oracle ACFS Replication
- Oracle ACFS Tagging
- Oracle ACFS Replication with Encryption
- Oracle ACFS Plugins
- Oracle ACFS Accelerator Volume
- Oracle ACFS NAS Maximum Availability eXtensions

Oracle ACFS and File Access and Administration Security

Oracle ACFS supports traditional Unix-style file access control classes (user, group, other) for Linux environments.

Most Oracle ACFS administrative actions are performed by users with either root or Oracle ASM administration privileges for Linux environments. General Oracle ACFS information for file systems can be accessed by any system user.

In support of Oracle ACFS administration, Oracle recommends that the Oracle ASM administrator role is given to a root privileged user, as many common Oracle ACFS file system management tasks including mount, umount, fsck, driver load, and driver unload are root privileged operations. Other privileged Oracle ACFS file system operations that do not require root privileges can be performed by the Oracle ASM administrator. If the Oracle ASM administrator role is not given to a root privileged user, access to Oracle ACFS file systems can be restricted with the `norootsuid` and `nodev` mount options.

> **See Also:**
>
> - Oracle ACFS Encryption for information about Oracle ACFS encryption
> - Managing Oracle ACFS with Command-Line Tools for information about administering Oracle ACFS
> - *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges

Oracle ACFS and Grid Infrastructure Installation

Oracle Grid Infrastructure includes Oracle Clusterware, Oracle ASM, Oracle ACFS, Oracle ADVM, and driver resources software components, which are installed into the Grid Infrastructure home using the Oracle Universal Installation (OUI) tool.

Oracle ACFS Configuration

After a Grid Infrastructure installation and with an operational Oracle Clusterware, you can use Oracle ASM Configuration Assistant (ASMCA) to start the Oracle ASM instance and create Oracle ASM disk groups, Oracle ADVM volumes, and Oracle ACFS file systems. Alternatively, Oracle ASM disk groups and Oracle ADVM volumes can be created using SQL*Plus and ASMCMD command line tools. File systems can be created using operating system command-line tools.

Oracle ACFS file systems are configured with Oracle ADVM based operating system storage devices that are created automatically following the creation of an Oracle ADVM dynamic volume file. After a volume file and its associated volume device file are created, a file system can be created and bound to that operating system storage device. Following creation, an Oracle ACFS file system can be mounted, after which it is accessible to authorized users and applications executing file and file system operations.

See Also:

- Basic Steps to Manage Oracle ACFS Systems for an example of the specific actions required to create a file system
- ASMCA GUI Tool for Managing Oracle ACFS and Oracle ADVM for information about managing Oracle ACFS file systems with ASMCA
- Managing Oracle ACFS with Command-Line Tools for information about using Oracle ACFS commands to create a file system
- Oracle Clusterware Resources and Oracle ACFS Administration for information about configuring Oracle Clusterware resources for use with the database

Oracle ACFS Features Enabled by Compatibility Attribute Settings

This topic describes the Oracle ACFS features enabled by valid combinations of the disk group compatibility attribute settings.

The following list applies to Oracle ACFS features enabled by disk group compatibility attribute settings.

- The value of `COMPATIBLE.ASM` must always be greater than or equal to the value of `COMPATIBLE.RDBMS` and `COMPATIBLE.ADVM`.
- Starting with Oracle Grid Infrastructure 12.2.0.1 software, the minimum setting for `COMPATIBLE.ASM` is `11.2.0.2`.

- A value of not applicable (n/a) means that the setting of the attribute has no effect on the feature.
- Oracle ACFS features not explicitly listed in the following table do not require advancing the disk group compatibility attribute settings.
- Oracle ACFS features explicitly identified by an operating system in the following table are available on that operating system starting with the associated disk group attribute settings.
- If encryption is configured for the first time on Oracle ASM 11*g* Release 2 (11.2.0.3) on Linux or if encryption parameters must be changed or a new volume encryption key must be created following a software upgrade to Oracle ASM 11*g* Release 2 (11.2.0.3) on Linux, then the disk group compatibility attributes for `ASM` and `ADVM` must be set to `11.2.0.3` or higher.
- Using replication or encryption with database files on Oracle ACFS is not supported.
- Oracle ACFS on Oracle Exadata storage is supported starting with Oracle Grid Infrastructure 12.1.0.2 on Linux.

The following table describes the Oracle ACFS features enabled by valid combinations of the disk group compatibility attribute settings.

Table 1-1 Oracle ACFS features enabled by disk group compatibility attribute settings

Disk Group Features Enabled	COMPATIBLE.ASM	COMPATIBLE.RDBMS	COMPATIBLE.ADVM
Volumes in disk groups	>= 11.2	n/a	>= 11.2
Encryption, replication, security, tagging (Linux systems)	>= 11.2.0.2	n/a	>= 11.2.0.2
Read-only snapshots	>= 11.2.0.2	n/a	>= 11.2.0.2
Read-write snapshots	>= 11.2.0.3	n/a	>= 11.2.0.3
Unlimited file system expansion	>= 11.2.0.4	n/a	>= 11.2.0.4
Performance and scalability improvements in `ls` and `find`	>= 11.2.0.4	n/a	>= 11.2.0.4
Storing database files in Oracle ACFS for Oracle RAC configurations.	>= 12.1	n/a	>= 12.1
Encryption, replication, security, tagging (Solaris systems)	>= 12.1	n/a	>= 12.1
Replication and tagging (AIX systems)	>= 12.1	n/a	>= 12.1
Creation from an existing snapshot and conversion of a snapshot	>= 12.1	n/a	>= 12.1
Support for 1023 snapshots	>= 12.1	n/a	>= 12.1.0.2
Storing database files in Oracle ACFS for Oracle Restart configurations	>= 12.2	n/a	>= 12.2
Accelerator volume for Oracle ACFS file system	>= 12.2	n/a	>= 12.2
Metadata storage on accelerator volume for Oracle ACFS file system	>=12.2, or >=12.1.0.2 on Oracle Data Appliance (ODA)	n/a	>=12.2, or >=12.1.0.2 on Oracle Data Appliance (ODA)
Logical sector size of the Oracle ADVM volume	>= 12.2	n/a	>= 12.2
Oracle ACFS support for 4 K sectors	>=12.2	n/a	>=12.2

Table 1-1 (Cont.) Oracle ACFS features enabled by disk group compatibility attribute settings

Disk Group Features Enabled	COMPATIBLE.ASM	COMPATIBLE.RDBMS	COMPATIBLE.ADVM
Oracle ACFS automatic resize	>=12.2, or >=12.1.0.2 on Oracle Data Appliance (ODA)	n/a	>=12.2, or >=12.1.0.2 on Oracle Data Appliance (ODA)
Oracle ACFS sparse files	>=12.2, or >=12.1.0.2 on Oracle Data Appliance (ODA)	n/a	>=12.2, or >=12.1.0.2 on Oracle Data Appliance (ODA)
Oracle ACFS compression	>=12.2	n/a	>=12.2
Oracle ACFS snapshot quotas	>= 12.2	n/a	>= 12.2
Oracle ACFS snapshot duplication	>= 12.2	n/a	>= 12.2
Oracle ACFS snapshot remastering	>= 12.2	n/a	>= 12.2
Space usage information by individual Oracle ACFS snapshots	>= 12.2	n/a	>= 12.2
Oracle ACFS snaphot-based replication	>=12.2	n/a	>=12.2
Oracle ACFS Defragger	>=12.2	n/a	>=12.2
Oracle ADVM volume on a flex disk group	>=12.2	>=12.2	>=12.2
Oracle ACFS replication role reversal or unplanned failover	>=18.0	n/a	>=18.0
Reducing the size of an Oracle ACFS file system	>=18.0	n/a	>=18.0
Oracle ACFS snapshot links	>=18.0	n/a	>=21.1
File Level Snapshots (Fshares)	>=21.1	n/a	>=21.1

See Also:

- Overview of Oracle ACFS for information, including any limitations or restrictions, about storing data files in an Oracle ACFS file system
- *Oracle Automatic Storage Management Administrator's Guide* for more information about disk group compatibility attributes
- *Oracle Automatic Storage Management Administrator's Guide* for information about the rebalancing process and the `ASM_POWER_LIMIT` initialization parameter

Oracle Clusterware Resources and Oracle ACFS Administration

Oracle Clusterware resources support all aspects of Oracle ACFS. The resources are responsible for enabling and disabling volumes, loading drivers and mounting and unmounting file systems.

This section discusses the following topics:

- Summary of Oracle ACFS Resource-based Management

- High Availability Actions
- Creating Oracle ACFS Resources
- Node-Local or Clusterwide File Systems
- Monitoring Oracle ACFS resources
- Stopping Oracle ACFS resources
- Oracle ACFS resource Limitations

Summary of Oracle ACFS Resource-based Management

The following list provides a summary of Oracle ACFS resource-based management.

- The Oracle ACFS, Oracle Kernel Services (OKS), and Oracle ADVM drivers are dynamically loaded when the Oracle ASM instance is started.
  - Oracle ACFS

    This driver processes all Oracle ACFS file and directory operations.
  - Oracle ADVM

    This driver provides block device services for Oracle ADVM volume files that are used by file systems for creating file systems.
  - Oracle Kernel Services Driver (OKS)

    This driver provides portable driver services for memory allocation, synchronization primitives, and distributed locking services to Oracle ACFS and Oracle ADVM.

  The drivers are managed as a single resource set. For additional information, see "Oracle ACFS Drivers Resource Management" and "Oracle ACFS Driver Commands".
- When a volume is created, Oracle ADVM creates a resource with the name of `ora.`*`DISKGROUP`*`.`*`VOLUME`*`.advm`. This resource is usually managed through transparent high availability calls from Oracle ASM and requires no user interaction. However, the user may choose to use the SRVCTL command interface to start and stop volumes as well as control the default state of the volume after an Oracle ASM restart. This is especially beneficial in a large cluster or an Oracle Flex ASM cluster, as volumes on other nodes may be operated upon.

  In addition, these Oracle ADVM resources can be used by other resources in the Oracle Clusterware stack to maintain dependency chains. Dependency chains ensure that the resources a program requires to run are available. For instance, if a resource was monitoring a backup application that was backing up to Oracle ADVM volume, the backup application would want to ensure that it specified the Oracle ADVM volume resource in it's `START` and `STOP` dependency list. Because the Oracle ADVM volume resource will enable the volume, this ensures that the volume is available before the backup begins.
- Oracle ACFS file systems are either manually mounted or dismounted using an Oracle ACFS or Oracle Clusterware command-line tool, or automatically mounted or dismounted based on an Oracle Clusterware resource action.

  For example, a file system hosting an Oracle Database home is named in the dependency list of the associated Oracle Database resource such that issuing a start on the database resource results in mounting the dependent Oracle ACFS hosted database home file system.

  Oracle ACFS file system resources provide the following actions:

  - MOUNT

    During the `START` operation the resource mounts the file system on the path configured in the resource. The Oracle ACFS file system resource requires all components of the Oracle ASM stack to be active (volume device, ASM) and ensures that they are active before attempting the mount.

  - UNMOUNT

    During the `STOP` operation, the resource attempts to unmount a file system.

- Oracle provides two resource types for Oracle Highly Available NFS. For more information, refer to Oracle ACFS NAS Maximum Availability eXtensions.

As with all Oracle Clusterware resources, these resources provide for high availability by monitoring the underlying device, file system, or driver to ensure that the object remains available. In the event that the underlying object becomes unavailable, each resource attempts to make the underlying object available again.

High Availability Actions

The following are the actions of the High Availability resources:

- Oracle ACFS resource

  This resource attempts to unmount the file system. After the unmount has succeeded, the resource remounts the file system, making the file system available again. If processes are active on the file system during unmount, the resource identifies and terminates those processes.

- Oracle ADVM resource

  This resource attempts to disable any volume device, and then reenable the volume device. At that point, any configured Oracle ACFS resource can remount the file system. If processes are active on the volume during this period, the resource identifies and terminates the processes.

Creating Oracle ACFS Resources

Oracle ACFS resources can be created with the following methods:

- Oracle ASM Configuration Assistant (ASMCA) provides a GUI that exposes the most common functionality. In all cases, creating a file system resource does not format the underlying file system. Attempts to start the resource require the user to format the file system either manually or with ASMCA.
- SRVCTL provides a highly flexible command line utility for creating Oracle ACFS file system resources through the *`filesystem`* object. Oracle ACFS resources created through this mechanism have access to the full feature set, including server pools.
- `acfsutil` commands provide an alternative method to create Oracle ACFS file system resources using the *`registry`* object. Oracle ACFS resources created through this methodology have access to a limited set of options.

The differences between SRVCTL and `acfsutil` commands are:

- Oracle ACFS resources created through SRVCTL and specifying a server pool or list of nodes are only mounted on one of those nodes. (node-local)

- Oracle ACFS resources created through SRVCTL can take advantage of Oracle Server Pools.
- Oracle ACFS resources created through `acfsutil` commands and specifying a list of nodes are mounted on all listed nodes. (node-local)
- Oracle ACFS resources created through `acfsutil` commands are created with `AUTOSTART` set to `ALWAYS`.
- Oracle ACFS resources created through SRVCTL allow for advanced Application ID functionality. Using this functionality enables the resource type to be set by the administrator. After the type is set, other resources can depend on this type, allowing different node-local file systems to be used to fulfill dependencies on each node. In a simplified example, this would allow the administrator to have a different device mounted on the `/log` directory on each node of the cluster, and be able to run an Apache resource. The Apache resource would specify the new type in its resource dependency structure, rather than specifying an individual resource.
- Oracle ACFS resources created through SRVCTL can specify additional `AUTOSTART` parameters. These parameters can be used to prevent the resource from starting on stack startup, to always force the resource to start, or to only start the resource if it was previously running.
- Oracle ACFS resources created through SRVCTL have access to functionality such as accelerator volumes.

The common elements of both SRVCTL and `acfsutil` commands are:

- User

  This is an additional user that can act upon the resource. By default, you must be the `root` user to start and stop an Oracle ACFS resource.
- Options

  These are mount options that should be used to mount the file system when the resource is starting.

Node-Local or Clusterwide File Systems

When creating Oracle ACFS file system resources, you can create a node-local file system or to create a clusterwide file system.

- Node-local

  This file system type is limited to the number of nodes it can mount on. Depending on if it is created with SRVCTL or `acfsutil` commands, it may only mount on one node, a subset of nodes, or all the configured nodes. In some cases, this could look the same as a full cluster configuration, but if new nodes are added to the cluster, the file system is not automatically mounted on them without modifying the list of allowable nodes.
- Clusterwide

  This type of file system mounts on all nodes of the cluster, with no exceptions. When new members are added to the cluster, the file system is automatically available on them. This type of resource is required for certain configurations, such as Oracle Database or Oracle HANFS.

Monitoring Oracle ACFS resources

Similar to all Oracle Clusterware resources, Oracle ACFS resources enables you to monitor the state of the system. You can do this monitoring with the following commands:

- Using SRVCTL commands

  When the command `srvctl status filesystem` or `srvctl status volume` is run, the output of the command reports if the file system is mounted or the volume is enabled, and which nodes this is true on.

- Using CRSCTL commands

  When the `crsctl status resource` command is run, a state of `ONLINE` is reported for each resource that is available, whether through a mounted file system or an enabled volume. A state of `OFFLINE` is reported for each resource that is not available, whether through an unmounted file system or a disabled volume. Additional status may be presented in the `STATUS` field of this output.

Stopping Oracle ACFS resources

You can be stop Oracle ACFS file system resources with the following methods:

- You can stop the entire Oracle Clusterware stack. When the Oracle Clusterware stack is stopped, all Oracle ACFS resources are automatically stopped.
- To stop individual resources, you can use SRVCTL management commands with the Oracle ACFS files system or volume object. If there are other resources that are depending on the resource that you are attempting to stop, then the command may require the `-force` option.
- You may engage a manual action, such as running `unmount` on a file system or by manually stopping a volume using ASMCMD or SQL*Plus commands. In this case, the Oracle ACFS resource transitions to the `OFFLINE` state automatically.

Non-Oracle Grid Infrastructure usage of mount points can prevent unmounting and disabling of volumes in the kernel for some situations. For example:

- Network File System (NFS)
- Samba/Common Internet File System (CIFS)

If either of the previous examples reflects your situation, then ensure that you discontinue the use of the functionality before initiating a stack shutdown, file system unmount, or volume disable.

Additionally, some user space processes and system processes may use the file system or volume device in a manner that prevents the Oracle Grid Infrastructure stack from shutting down during a patch or upgrade. If this problem occurs, then use the `lsof` and `fuser` commands (Linux and UNIX) to identify processes which are active on the Oracle ACFS file systems and Oracle ADVM volumes. To ensure that these processes are no longer active, dismount all Oracle ACFS file systems or Oracle ADVM volumes and issue an Oracle Clusterware shutdown. Otherwise, errors may be raised during an Oracle Clusterware shutdown relating to activity on Oracle ACFS file systems or Oracle ADVM volumes, preventing the successful shutdown of Oracle Clusterware.

Oracle ACFS resource Limitations

Oracle ACFS has the following resource limitations:

- All Oracle ACFS resources require `root` privileges to create.
- All Oracle ACFS resources require `root` privileges to remove.
- All Oracle ACFS file system resources require `root` privileges to act upon, such as starting and stopping the resources, but can be configured to allow another user, such as a database user, to do so. In this case, the `root` user must be used to configure the resource.
- All Oracle ADVM volume resources allow the `ASMADMIN` user to act upon them.
- All Oracle ACFS resources are only available in Oracle RAC mode. Oracle ACFS resources are not supported in Oracle Restart configurations. For more information about Oracle ACFS and Oracle Restart, refer to Oracle ACFS and Oracle Restart.

Oracle ACFS and Dismount or Shutdown Operations

It is important to dismount any active file system configured with an Oracle ADVM volume device file before an Oracle ASM instance is shutdown or a disk group is dismounted. After the file systems are dismounted, all open references to Oracle ASM files are removed and associated disk groups can be dismounted or the instance shut down.

If the Oracle ASM instance or disk group is forcibly shut down or fails while an associated Oracle ACFS is active, the file system is placed into an offline error state. If any file systems are currently mounted on Oracle ADVM volume files, the `SHUTDOWN ABORT` command should not be used to terminate the Oracle ASM instance without first dismounting those file systems. Otherwise, applications encounter IO errors and Oracle ACFS user data and metadata being written at the time of the termination may not be flushed to storage before the Oracle ASM storage is fenced. If it is not possible to dismount the file system, then you should run two `sync` (1) commands to flush cached file system data and metadata to persistent storage before issuing the `SHUTDOWN ABORT` operation.

Any subsequent attempt to access an offline file system returns an error. Recovering a file system from that state requires dismounting and remounting the Oracle ACFS file system. Dismounting an active file system, even one that is offline, requires stopping all applications using the file system, including any shell references. For example, a previous change directory (`cd`) into a file system directory. The Linux `fuser` or `lsof` commands list information about processes and open files.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about shutting down an Oracle ASM instance
- *Oracle Automatic Storage Management Administrator's Guide* for information about dismounting a disk group

Oracle ACFS Encryption

Oracle ACFS encryption enables you to encrypt data stored on disk (data-at-rest).

The encryption feature protects data in an Oracle ACFS file system in encrypted format to prevent unauthorized use of data in the case of data loss or theft. Data is never written to secondary storage in plaintext. Even if physical storage is stolen, the data stored cannot be accessed without encryption keys. Encryption keys are never stored in plaintext.Both encrypted and non-encrypted files can exist in the same Oracle ACFS file system.

Some encryption functionality requires system administrator privileges. This functionality incudes the commands for initiating, setting, and reconfiguring encryption.

System administrators and Oracle ACFS security administrators can initiate encryption operations. Also, unprivileged users can initiate encryption for files they own.

Oracle ACFS encryption provides two type of encryption keys:

- File Encryption Key (FEK)

  This is a key for a file and is used to encrypt the data in the file.
- Volume Encryption Key (VEK)

  This is a key for a file system and is used to encrypt the file encryption keys.

You must first create the encryption key store, then specify file system-level encryption parameters and identify the directories. No extra steps are required for a user to read encrypted files if the user has the appropriate privileges for accessing the file data.

Oracle ACFS encryption supports both Oracle Cluster Registry (OCR) and Oracle Key Vault as a key store. Both OCR and Oracle Key Vault can be used in the same cluster. However, a single file system uses either OCR or Oracle Key Vault as a key store for VEKs. Oracle Key Vault is currently only available with file systems on Linux.

If you are using OCR as a key store for VEKs, you can store VEKs in either password-protected PKCS wallets or passwordless SSO wallets. You should back up the OCR after creating or updating a volume encryption key to ensure there is an OCR backup that contains all of the volume encryption keys (VEKs) for the file system.

If you are using Oracle Key Vault as a key store for VEKs, note that Oracle Key Vault endpoints must be created with an endpoint type of “Oracle ACFS”. Additionally, all Oracle Key Vault endpoints within the same cluster must share the same endpoint password.

Within an OCI environment, Oracle ACFS encryption also provides integration with OCI Vaults. Master Encryption Keys can be generated within OCI Vaults, and these Master Encryption Keys can be used to encrypt the VEKs for ACFS file systems. This adds an extra level to the Oracle ACFS encryption key hierarchy and adds key management capabilities within OCI. Note that when VEKs are encrypted by OCI Vault Master Encryption Keys, they must be stored in the OCR using passwordless SSO wallets.

Auditing and diagnostic data are logged for Oracle ACFS encryption. The log files include information such as `acfsutil` commands that have been run, the use of security or system administrator privileges, and run-time failures. Logs are written to the following files:

- *mount_point*/.Security/encryption/logs/encr-*hostname_fsid*.log

  The directory is created with acfsutil encr set command.

- *GRID_HOME*/log/*hostname*/acfs/security/acfssec.log

  The messages that are logged to this file are for commands that are not associated with a specific file system, such as acfsutil encr init. The directory is created during installation and is owned by the root user.

When an active log file grows to a pre-defined maximum size (10 MB), the file is automatically moved to *log_file_name*.bak, the administrator is notified, and logging continues to the regular log file name. When the administrator is notified, the administrator must archive and remove the *log_file_name*.bak file. If an active log file grows to the maximum size and the *log_file_name*.bak file exists, logging stops until the backup file is removed. After the backup log file is removed, logging restarts automatically.

Note the following when working with Oracle ACFS encryption:

- A copy of an encrypted file is not encrypted unless the copy of the file is made in an encrypted directory.

  Some applications, such as the vi editor, re-create a file when the file is modified. The modified file is saved as a temporary file, the original file is removed, and temporary file is copied with the original file name as the destination name. This process creates a new file. The new file is not encrypted unless it is created in an encrypted directory. If you are planning to copy an encrypted file, you should ensure that the parent directory is also encrypted.

- Using encryption with database files on Oracle ACFS is not supported.
- Oracle ACFS encryption cannot be used with password-protected (PKCS) wallets if any of the file systems using encryption are configured to be mounted with the Oracle ACFS mount registry.
- ACFS encryption is not FIPS-140 compliant.

To use Oracle ACFS encryption functionality on Linux, the disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.2 or higher. The disk group compatibility attributes for ASM and ADVM must be set to 11.2.0.3 or higher on Linux for the following cases:

- If encryption is configured for the first time on Oracle ASM 11*g* Release 2 (11.2.0.3).
- If encryption parameters must be changed or a new volume encryption key must be created following a software upgrade to Oracle ASM 11*g* Release 2 (11.2.0.3). .

Encryption information for Oracle ACFS file systems is displayed in the V$ASM_ACFS_ENCRYPTION_INFO view. To configure encryption and manage encrypted Oracle ACFS file systems, you can use the acfsutil encr command-line functions and Oracle ASM Configuration Assistant.

Note:

Starting with Oracle ACFS 21c, Oracle ACFS encryption is desupported on Solaris and Microsoft Windows operating systems. Oracle ACFS Encryption on Oracle Solaris and Microsoft Windows is based on RSA technology. Retirement of RSA technology has been announced. Oracle ACFS Encryption continues to be supported on Linux, and is unaffected by this deprecation, because Linux uses an alternative technology.

See Also:

- acfsutil encr set and acfsutil encr rekey for information about changing or creating a volume encryption key
- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- About Oracle ACFS Snapshots for information about Oracle ACFS encryption and snapshots
- Using Views to Display Oracle ACFS Information for information about `V$ASM_ACFS` views
- Encrypting Oracle ACFS File Systems using OCR as Encryption Key Store and Oracle ACFS Command-Line Tools for Encryption for information about managing encryption with Oracle ACFS command-line tools
- Managing Security and Encryption for Oracle ACFS with ASMCA for information about using Oracle ASMCA with encryption features
- *Oracle Key Vault Administrator's Guide* for information about Oracle Key Vault

Oracle ACFS Compression

Oracle ACFS compression is enabled on a specified Oracle ACFS file system for general purpose files. Oracle ACFS compression is not supported for Oracle Database files.

Cached IO compression is performed asynchronously, after the application has written to the file. When enabling compression on a file system, existing files are not compressed, only newly-created files are compressed. When compression is disabled, compressed files are not uncompressed. Compressed files are associated with a compression unit and the compression algorithm operates on this unit. The default unit size is currently 32 K. `lzo` is the default compression algorithm and the only compression algorithm currently supported.

The `acfsutil compress` command sets and resets the compression state of a file system with `acfsutil compress on` and `acfsutil compress off`. To display the compression state and effectiveness of the compression operation, use the `acfsutil` `compress info` command. The `acfsutil info fs` and `acfsutil info file` commands have been enhanced to report on Oracle ACFS compression status.

Compressed files consume less disk space than non-compressed files. However, for applications using the file, the size reported is equal to the uncompressed file size, not the smaller compressed size. Some utilities, such as `ls -l`, report the uncompressed size of the file. Utilities such as `du`, `acfsutil compress info`, and `acfsutil info file`, report the actual disk allocation of the compressed file.

Note the following about Oracle ACFS compression.

- Oracle ACFS compression is not supported for Oracle ACFS file systems which are intended to hold database files. Instead, use Oracle Advanced Compression.
- Loopback mounts are not supported with compressed files. This includes files intended for use by Oracle ACFS remote service. If a loopback device is associated with a compressed file, read and write operations to the loopback device fail.
- A loopback device can be associated with an uncompressed file on an Oracle ACFS file system that has been enabled for compression.
- For Oracle Grid Infrastructure 12*c* release 2 (12.2.0.1), Oracle ACFS compression is supported on Linux and AIX.
- Oracle ACFS compression is only supported with Oracle ACFS snapshot-based replication that is available starting with Oracle Grid Infrastructure 12c release 2 (12.2.0.1).
- ADVM disk group compatibility must be set to `12.2` or higher.

See Also:

- Oracle ACFS Command-Line Tools for Compression for information about the Oracle ACFS compression commands
- Oracle ACFS Command-Line Utilities for information about the `acfsutil info` commands

Oracle ACFS Replication

Oracle ACFS snapshot-based replication enables replication of Oracle ACFS file systems across a network to a remote site, providing disaster recovery capability for the file system.

Oracle ACFS replication enables either a mounted file system or a snapshot of a mounted file system to be designated as a replication *storage location*. The source Oracle ACFS location of an Oracle ACFS replication relationship is referred to as a primary location. The target Oracle ACFS location of an Oracle ACFS replication relationship is referred to as a standby location.

> **Note:**
>
> - Oracle ACFS replication functionality supports only one standby location for each primary location.
> - The standby location is read-only for as long as replication is active on it. Read-write snapshots may be created of the standby.
> - A primary site running Linux, Solaris or AIX can replicate to a standby site running any of those operating systems where Oracle ACFS is supported.
>
>   Note that application data is not modified in any cross-platform use of Oracle ACFS replication. Oracle ACFS replication only ensures the validity of its metadata when transferring between different operating systems.
> - The primary and standby sites should be running the same version of the Oracle Grid Infrastructure software. However, replication will work correctly when the primary and standby sites are running different versions, so long as both versions are 19c or later. So long as 19c or later is in use, the two sites may be upgraded in either order.
> - Oracle ACFS replication is not supported with Oracle Restart.
> - An Oracle Key Vault keystore is not supported on a standby file system containing replication locations.
> - Oracle ACFS encryption cannot be undone on a primary file system containing replication locations.
>
>   You cannot undo encryption on a file system having active snapshots. If you want to undo encryption on primary file system containing active replication locations, then first terminate replication. After replication has stopped, then undo encryption and start replication again.

A site can host both primary and standby locations. For example, if there are cluster sites `A` and `B`, a primary file system hosted at site `A` can be replicated to a standby snapshot at site `B`. At the same time, a primary snapshot hosted at site `B` can be replicated to a standby file system at site `A`. However, an Oracle ACFS file system or snapshot cannot be used simultaneously as a primary and a standby location.

Oracle ACFS snapshot-based replication operates by recording snapshots of the primary location. After the initial snapshot is transferred to the standby location, replication continues by transferring the changes between successive snapshots of the primary to the standby location. These replication operations can occur either in *constant mode* (enabling a new operation to start as soon as the previous one completes), or can be scheduled to occur at fixed intervals. This replication solution is by nature asynchronous.

Oracle ACFS replication uses snapshot functionality on the primary site initially to externalize both the contents of the initial snapshot, and later the differences between two specified snapshots. The result is called a snapshot duplication stream. The replication process then uses snapshot functionality on the standby site to apply this stream to the standby location, creating a duplicate of the primary location.

On the primary, because replication works by comparing successive snapshots, it is critical that there is enough disk space available on the site hosting the primary to contain the version of the primary recorded in each snapshot, as well as the current primary contents. In addition, there must always be enough space to create the snapshots required. Each replication snapshot is deleted when no longer needed.

On the standby, a backup snapshot is created at the end of each replication operation. This snapshot records the latest consistent contents of the standby, and can be used to recover those contents if a permanent outage occurs during the current replication operation. Each backup snapshot is deleted when the following replication operation is complete, so it must always be possible to create a backup snapshot. In addition, enough space must exist for the version of the standby captured in the snapshot and the current standby contents.

You should ensure that the primary and standby file systems do not run out of disk space. If either file system runs out of available storage, you should either expand the file system or free up space by removing files from the file system or any read-write snapshots present. You can also configure automatic resize to avoid running out of space.

If the primary file system runs out of space and you decide to free up space by removing files, then you should only remove files that are not being replicated. Replicated files have been stored in a snapshot pending transfer to the standby file system and are not deleted. You can delete any Oracle ACFS snapshots not created by replication.

Replication Configuration Information

As of the 23ai release, on the OL8 / X64 platform, Oracle ACFS replication can use either SSL-enabled sockets or the ssh utility as the transport between the primary and standby clusters. For other platforms, ssh remains the only transport. For either transport, replication requires a specific user to be designated as the replication user, or *repluser*.

If SSL is used as the transport, configuration consists of the simple sharing of credentials between the two clusters.

For more information, refer to Configuring SSL-Based Oracle ACFS Replication.

If `ssh` is used as the transport, configuration must enable the automated use of `ssh`, by setting up two kinds of keys for repluser. These keys must be available on each node where replication is enabled to run.

- On each node in each cluster, the *repluser* must have a *host key* stored for each node in the other cluster.
- On each node in each cluster, a public key for *repluser*, as defined on the other cluster, must be stored that is authorized to log in as *repluser* on that node..

These *keys* are required to ensure that replication running on a primary host is able to authenticate the standby host to which it is sending data using the host keys. Also, replication running on a primary host must be able to log in as the *apply user* on a standby host with the user keys to update the standby location. In addition, replication has the ability to reverse the roles being played by primary and standby. For this switchover operation to be successful, primary and standby hosts require the same types of `ssh` keys to be configured.

To ease the use of ssh as its transport, in the 23ai release replication provides a key setup assistant, acfsreplssh, that partially automates configuration.

For more information, refer to Configuring `ssh` for Use With Oracle ACFS Replication.

Before using replication on a file system or snapshot, ensure that you have checked the following:

- There is sufficient network bandwidth to support replication between the primary and standby locations.
- The configuration of the sites hosting the primary and standby locations enable the standby file system to keep up with the rate of change on the primary location.
- If SSL is used as the transport, that credentials have been shared between the two sites.
- If ssh is used as the transport, that host keys and user keys for `ssh` have been configured properly.

Directories and files in an Oracle ACFS file system or snapshot can be tagged to select specific objects that you want to replicate in a file system.

Before replicating a given location, a replication configuration must be created to identify any necessary information, such as the site hosting the primary location, the site hosting the standby location, the file system to be replicated, the mount point of the file system for the location, and a list of tags.

The primary and standby sites must share the same user and group configurations, including all uids and gids in use in the two locations. The apply user *repluser* described previously must be configured on each node where replication is enabled. This user should be a member of the Oracle ASM administration group and must have Oracle ASM administrator user privileges.

> **Note:**
>
> The mappings between user names and numeric `uids`, and between group names and numeric `gids`, must be identical on both the primary cluster and the standby cluster. This is required to ensure that the numeric values are used in the same manner on both clusters because replication transfers only the numeric values from the primary to standby.

Replication Switchover

Oracle ACFS replication provides role reversal or switchover functionality, which enables the original primary and standby locations to reverse roles. Using the `acfsutil repl switchover` command, you can change the original primary to the new standby, and the original standby to the new primary. The switchover functionality enhances replication to provide additional disaster recovery capabilities.

Replication Unplanned Failover

Oracle ACFS replication provides for unplanned failover functionality with the `acfsutil repl failover` command. The command can be invoked on the standby in both the planned failover case (primary up) and the unplanned failover case (primary down). To distinguish the two cases, `acfsutil repl failover` attempts for a specific amount of time to contact the primary. There is an option available to control how long the standby waits to hear from the primary before deciding that the primary is down.

When `acfsutil repl failover` attempts to contact the primary:

- If the primary responds, then the standby collaborates with the primary to do a planned failover with no data loss.

- If the primary does not respond, then the standby assumes that the primary is down, and initiates an unplanned failover with no involvement of the primary. There is a likely chance of incurring data loss.

Additional Information

To use Oracle ACFS replication functionality, the disk group compatibility attributes for `ASM` and `ADVM` must be set to `12.2` or higher for the disk groups that contain the file systems for the primary and standby locations. To use Oracle ACFS role reversal or unplanned failover replication functionality, the disk group compatibility attributes for `ASM` and `ADVM` must be set to `18.0` or higher for the disk groups that contain the file systems for the primary and standby locations.

To use Oracle ACFS replication on Solaris Sparc hardware, the system must be running Solaris 10 update 8 or later.

To configure replication and manage replicated Oracle ACFS locations, use the `acfsutil repl` command-line functions.

> **Note:**
>
> Starting with Oracle ACFS 21c, Oracle ACFS replication protocol version 1 is desupported. Replication protocol version 1 has been replaced with snapshot-based replication version 2, introduced in Oracle ACFS 12c Release 2 (12.2.0.1).

See Also:

- Oracle ACFS Command-Line Tools for Replication for information about Oracle ACFS `acfsutil` replication commands
- Oracle ACFS Replication with Encryption for information about using Oracle ACFS replication with other Oracle ACFS features
- acfsutil size for information about resizing an Oracle ACFS file system
- Oracle ACFS Tagging for information on tagging
- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- Replicating Oracle ACFS File Systems for information about using the `acfsutil repl` command-line functions to configure replication and manage replicated Oracle ACFS file systems
- Configuring Oracle ACFS Snapshot-Based Replication for information about converting an existing Oracle ACFS replication environment to the snapshot-based replication introduced in release 12.2.0.1
- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges
- Information about tuning your network on the Oracle Maximum Availability Architecture (MAA) website at `https://www.oracle.com/database/technologies/high-availability/maa.html`
- *Data Guard Redo Transport & Network Configuration* document for relevant information about tuning your network
- Additional information on the Oracle ACFS website at `https://www.oracle.com/database/technologies/rac/acfs.html`

Oracle ACFS Tagging

Oracle ACFS tagging assigns a common naming attribute to a group of files.

Oracle ACFS Replication can use this tag to select files with a unique tag name for replication to a different remote cluster site. The tagging option avoids having to replicate an entire Oracle ACFS file system.

Oracle ACFS implements tagging with Extended Attributes. Some editing tools and backup utilities do not retain the Extended Attributes of the original file by default; you must set a specific switch. The following list describes the necessary requirements and switch settings for some common utilities to ensure Oracle ACFS tag names are preserved on the original file.

- The `cp` command requires flags to preserve tag names.

  Install the `coreutils` library (version `coreutils-5.97-23.el5_4.1.src.rpm` or `coreutils-5.97-23.el5_4.2.x86_64.rpm` or later) on Linux to install versions of the `cp` command that supports Extended Attribute preservation with the `--preserve=xattr` switch and the `mv` command that supports Extended Attribute preservation without any switches.

`cp` does not preserve tag names assigned to symbolic link files.

The `cp` switches required to preserve tag names on files and directories are:

  - Linux: `--preserve=xattr`
  - Solaris: `-@`
  - AIX: `-U`

- The `cpio` file transfer utility requires flags to preserve tag names.

  The `cpio` switches required to preserve tag names on files and directories are:

  - Linux: `cpio` does not preserve tag names
  - Solaris: `-@` is required to preserve or restore tag names for files and directories, but does not preserve tag names for symbolic link files
  - AIX: `-U` is required to preserve or restore tag names for files and directories, but does not preserve tag names for symbolic link files

- `emacs` requires that the `backup-by-copying` option is set to a non-nil value to preserve tag names on the original file name rather than a backup copy. This option must be added to the `.emacs` file.
- The `pax` file transfer utility requires flags to preserve tag names.

  The `pax` switches required to preserve tag names on files and directories are:

  - Linux: `pax` does not preserve tag names
  - Solaris: `-@` is required to preserve or restore tag names for files and directories, but does not preserve tag names for symbolic link files
  - AIX: `-U` is required to preserve or restore tag names for files and directories, but does not preserve tag names for symbolic link files

- The `rsync` file transfer utility requires flags to preserve tag names.

  The `rsync` switches required to preserve tag names on files and directories are:

  - Linux: `-X -l` are required to preserve tag names for files and directories, but these switches do not preserve tag names for symbolic link files
  - Solaris: `rsync` does not preserve tag names
  - AIX: not available

- The `tar` backup utility can have flags set on the command line to preserve tag names on a file. However, `tar` does not retain the tag names assigned to symbolic link files.

  The `tar` switches required to preserve tag names on files and directories are:

  - Linux: `--xattrs`
  - Solaris: `-@`
  - AIX: `-U`

- The `vim` or `vi` editors require the `set bkc=yes` option in the `.vimrc` (Linux) file to make a backup copy of a file and overwrite the original. This preserves tag names on the original file.

To use Oracle ACFS tagging functionality on Linux, the disk group compatibility attributes for `ASM` and `ADVM` must be set to `11.2.0.2` or higher. To use Oracle ACFS tagging functionality on

Solaris or AIX, the disk group compatibility attributes for `ASM` and `ADVM` must be set to `12.1` or higher.

> **See Also:**
>
> - Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
> - Tagging Oracle ACFS File Systems and Oracle ACFS Command-Line Tools for Tagging for information about using the `acfsutil tag` command-line functions to configure tagging and manage tagged Oracle ACFS file systems
> - Oracle ACFS Tagging Generic Application Programming Interface for information about Oracle ACFS tagging application programming interfaces (APIs)

Oracle ACFS Replication with Encryption

> **Note:**
>
> Starting with Oracle ACFS 21c, Oracle ACFS Security (Vault) and ACFS Auditing are desupported. Desupporting cluster features with limited adoption allows Oracle to focus on improving core scaling, availability, and manageability across all features and functionality. Oracle ACFS Security (Vault) and ACFS Auditing are desupported.

Encryption can be enabled on an Oracle ACFS file system on which replication has been configured. The replicated standby file system is secured with the same encryption policies as the primary file system. For this replicated environment, the primary and standby file systems must both be 12.1 or higher installations. For more information about Oracle ACFS replication, refer to "Oracle ACFS Replication".

To ensure successful replication, the standby file system must be a generic file system without encryption metadata on it. Oracle ACFS does not support using a standby file system that once had encryption and then had encryption removed. Additional conditions that must be met for Oracle ACFS encryption are listed in this section.

Note the following about Oracle ACFS encrypted file systems:

- Encrypted files on the primary file system remain encrypted on the standby file system with the same key and encryption parameters (algorithm and key length).
- Encryption operations done on the primary file system are replayed on the standby file system - on, off, and rekey.
- Encryption may be enabled before or after a file system is replicated. In either case, an encryption wallet is transparently created on the standby file system if one does not exist because `acfsutil encr init` has not been run on the standby file system.

- A password-protected wallet is not supported on the standby file system. If a PKCS wallet already exists on a site that is to be used as a standby file system, the administrator must use the `acfsutil keystore migrate` command to transfer all keys to an SSO wallet.

Oracle ACFS Plugins

The Oracle ACFS plugin functionality enables a user space application to collect *just-in-time* Oracle ACFS file and Oracle ADVM volume metrics from the operating system environment.

Applications can use the Oracle ACFS plug-in infrastructure to create customized solutions that extend the general application file metric interfaces to include detailed Oracle ACFS file system and volume data.

The Oracle ACFS plug-in functionality can be enabled on separate Oracle ACFS file systems mounted on a standalone host or on one or more nodes of an Oracle Grid cluster where the Oracle ACFS file system is mounted. This functionality enables message communication between a node-local plugin enabled Oracle ACFS file system and an associated user space application module using Oracle ACFS plug-in application programming interfaces (APIs).

The plugin message APIs support both polling and posting message delivery models and multiple message payload types.

See Also:

- Oracle ACFS Command-Line Utilities for information about Oracle ACFS plugin commands
- Oracle ACFS Plug-in Generic Application Programming Interface for information about the Oracle ACFS plug-in application programming interface

Oracle ACFS Accelerator Volume

Using an accelerator volume can improve performance by reducing the time to access and update Oracle ACFS metadata. You should create the accelerator volume on a disk group with storage that is significantly faster than the storage of the primary volume. For example, Solid State Disk (SSD) storage could be used. Oracle ADVM volumes are created with the ASMCMD `volcreate` command. For information about the `volcreate` command, refer to volcreate.

The recommended size of the accelerator volume depends on the workload. It is especially helpful for files with many extents, especially if that extent metadata is updated frequently. You can use the `acfsutil info file` command to view a report on a file's extents. Database files generally have many extents and when Oracle ACFS snapshots are in use, the extent metadata is updated frequently. A workload that greatly benefits from an accelerator is a compressed file system.

If Oracle ACFS cannot allocate space on the accelerator for critical metadata, then that metadata is stored on the primary volume instead. Depending on the frequency of metadata updates, it can have a disproportionate impact on performance. If the slow metadata is written in the same transaction as the fast metadata, then the slow metadata brings the performance of the entire operation down.

The recommended starting accelerator size is minimally 0.6% of the size of the file system. If many snapshots are in use representing several points in time for a database workload, the recommendation is an additional 0.4% per snapshot. For example, a file system with 5 snapshots may need an accelerator whose size is 2.6% of the size of the primary volume. `acfsutil size` can be configured to automatically grow the accelerator as needed along with the primary volume. The accelerator increases in units of 64 mega bytes. The minimum size of the accelerator volume is 256 M. `mkfs` requires that the initial accelerator size be at least 0.4% of the size of the primary volume

The accelerator volume is linked to the primary volume specified with the `mkfs` command. When mounting a file system, only the primary volume is specified. If the accelerator volume becomes inaccessible for any reason after a file system with the volume is mounted, then the file system is taken offline. Only one storage accelerator volume can be associated with an Oracle ACFS file system. After an accelerator volume is associated with a file system, the volume cannot be disassociated from the file system.

The accelerator volume can be created on Linux environments with the `-a` option of the `mkfs` command. To create an accelerator volume on Linux, the value of `COMPATIBLE.ADVM` must be at least `12.2`. For information about the commands used to manage accelerator volumes, refer to:

- mkfs for information about creating an accelerator volume
- acfsutil accel replace for information about replacing an existing accelerator volume

Oracle ACFS NAS Maximum Availability eXtensions

Oracle ACFS NAS Maximum Availability eXtensions (Oracle ACFS NAS/MAX) is a set of extensions that provide High Availability Extensions for Common NAS Protocols, such as NFS and SMB.

When using these extensions, the protocol in question is running in high availability mode, enabling the protocol to move between nodes in an Oracle RAC cluster. This functionality provides a way to address a single point of failure for a given protocol, so that if at least one node of the cluster is available, then the protocol is available. In addition to providing for high availability, the extensions provide for integration with common NAS protocols and the Oracle ACFS stack, enabling administrators to easily utilize these protocols without creating additional infrastructure. Note that the Oracle ACFS NAS Maximum Availability eXtensions functionality adds value to existing OS NAS protocol implementations, but does not replace them.

Oracle ACFS NAS/MAX is supported as a dNFS target for RMAN backup sets. This support begins with Oracle ACFS 19c release. This allows the HA-NFS functionality to be used to provide scaling and highly available dNFS backup targets for RMAN. Other stated restrictions in this guide, such as the lack of support for encryption or replication are still valid.

Oracle ACFS High Availability Network File System

High Availability Network File System (HANFS) for Oracle Grid Infrastructure provides uninterrupted service of NFS v2, v3, or v4 exported paths by exposing NFS exports on Highly Available Virtual IPs (HAVIP) and using Oracle Clusterware agents to ensure that the VIPs and NFS exports are always online. While base NFS supports file locking, HANFS does not support NFS file locking, except with NFS v4. Refer to Oracle ACFS HANFS with NFS Locks.

> **Note:**
>
> - This functionality relies on a working NFS server configuration available on the host computer. You must configure the NFS server before attempting to use the Oracle ACFS NFS export functionality.
> - This functionality is not supported in Oracle Restart configurations.
> - The HAVIP cannot be started until at least one file system export resource has been created for it.

To set up High Availability NFS for Oracle Grid Infrastructure, perform the following steps:

1. Add and register a new HAVIP resource.

   For example:

   ```
   # srvctl add havip -id hrexports -address my_havip_name
   ```

   In the example, *`my_havip_name`* is mapped in the domain name server (DNS) to the VIP address and is used by the client systems when mounting the file system.

   The initial processing of `srvctl add havip` ensures that:

   - The address being used is static, not dynamic
   - Any DNS names resolve to only one host, not round-robin multiple DNS resolutions
   - The network resource and provided IP address and resolved name are in the same subnet
   - The name is not in use

   SRVCTL creates the appropriate HAVIP name using the *`id`*, ensuring it is unique. As a final validation step, SRVCTL ensures that the *`network`* resource (if provided) of `ora.net#.network` exists. After this step, SRVCTL adds a new `havip` of type `ora.havip.type` with the name of `ora.`*`id`*`.havip`. In this example, the name is `ora.`*`hrexports`*`.havip`.

   Next SRVCTL modifies HAVIP start dependencies, such as `active dispersion`; sets the stop dependencies; and ensures the *`description`* attribute (if provided) is appropriately set.

2. Create a shared Oracle ACFS file system.

   High Availability NFS for Oracle Grid Infrastructure operates only with Oracle ACFS file systems configured for clusterwide accessibility and does not support Oracle ACFS file systems configured for access on particular subsets of cluster nodes. High Availability NFS is not supported with non-Oracle ACFS file systems.

3. Register the Oracle ACFS file system.

   For example:

   ```
   $ srvctl add filesystem -device /dev/asm/d1volume1-295 -volume VOLUME1 \
     -diskgroup HR_DATA -mountpath /oracle/cluster1/acfs1
   ```

4. Create an Oracle ACFS file system export resource.

   For example:

   ```
   # srvctl add exportfs -id hrexports -path /oracle/cluster1/acfs1 -name 
   hrexport1
   ```

   After the file system export resource has been created, then you can start the HAVIP created in step 1 to export the file system using the `srvctl start havip` command.

   The NFS mount option FSID is added to any export options, utilizing the FSID of the underlying Oracle ACFS file system plus a unique identifier. This FSID option provides for reliable fail over between nodes and allows the usage of snapshot mounting.

   The default mount and export options for configured exports are the defaults for the NFS server.

   Relative paths that are fully-qualified are converted to absolute paths. Relative paths that are not fully-qualified are not accepted as an export path.

   VIPs attempts to find the best server to run on based on available file systems and other running VIPs, but this dispersion only occurs during CSS membership change events, such as a node joining or leaving the cluster.

   > **Note:**
   >
   > It is not recommended to start and stop exports individually; this functionality should be provided through the start and stop operations of HAVIP.
   >
   > When HAVIP is not running, exports can exist on different nodes. After the associated HAVIP is started, the exports gather on a single node.
   >
   > Clients that are using an export that is stopped while HAVIP is running raise the NFS error `estale`, and must dismount and remount the file system.
   >
   > When mounting an HANFS exported file system on a client, the following `CLIENT` mount options are recommended:
   >
   > ```
   > hard,intr,retrans=10000
   > ```

Oracle ACFS HANFS with NFS Locks

Oracle ACFS HANFS now supports HANFS NFS v4 with NFS Locks. This functionality is only available on specific operating system (OS) platforms. To activate this functionality, additional steps must be performed after the Oracle Grid Infrastructure software is installed. Note that after these steps are completed, the OS NFS server functionality of the cluster is managed by the Oracle Clusterware stack. In addition, the location of certain OS NFS configuration files will be moved from their default location to a designated Oracle ACFS file system.

Some common tasks are:

- Activate: `acfshanfs addnode`
- Uninstall: `acfshanfs uninstall`

- Check the installation status: `acfshanfs installed`
- Check if this platform is supported: `acfshanfs supported`

When activating the HANFS v4 lock functionality, the following command must be run on each node:

```
# grid_home/bin/acfshanfs addnode -nfsv4lock -volume volume_device
```

The volume is formatted with an Oracle ACFS file system and mounted on a designated Oracle ACFS clusterware mount point For example on Linux:

```
/dev/asm/nfs-81 on /var/lib/nfs type acfs (rw)
```

Restrictions on the Oracle ADVM volume include:

- No previously existing Oracle ACFS resource should exist for this new Oracle ADVM volume.
- No Oracle ACFS file system should exist on this Oracle ADVM volume.
- This Oracle ADVM volume should not be in use anywhere in the cluster.

When Oracle HANFS v4 lock functionality is activated, there are differences from normal HANFS operations. The differences are noted in the following list:

- The OS NFS server is under Oracle Clusterware control through the `ora.netstorageservice` resource. When starting and stopping the Oracle Clusterware stack, the OS NFS server is also started and stopped.
- This resource has a dependency on an Oracle ACFS file system: `ora.data_`*`hostname`*`.nfs.acfs`

  The *hostname* is the hostname of the first node on which the setup for Oracle HANFS locking has been run.
- Only Oracle HANFS should be used to export NFS file systems from the Oracle RAC cluster. The NFS server is configured and moved around the Oracle RAC cluster; only file systems exported by Oracle HANFS are accessible when the NFS server has migrated to an alternate cluster node.
- When locking is initialized, Oracle HANFS exports are run from only a single node, unlike non-locking mode, where Oracle HANFS exports are distributed throughout the cluster.
- On client nodes, mount the file system specifying NFS v4 as the NFS version. This prevents the server from defaulting to NFS v3, and enables support for the NFS v4 locking functionality.

After High Availability Locking is activated, control of HANFS with locking is the same as described previously in this section.

Oracle ACFS HANFS with High Availability SMB

Oracle ACFS supports High Availability Samba (SMB), also known as CIFS (Common Internet File System) in previous Microsoft implementations. This protocol is commonly used to interface with Microsoft servers and Active Directory Domains and is supported by various operating system (OS) implementations. However, Oracle ACFS High Availability SMB requires the Microsoft SMB implementation or Samba.

Note the following:

- Samba is available from www.samba.org

- Ensure that Samba or SMB is correctly configured on your host OS before attempting to utilize High Availability SMB.
- High Availability SMB is not supported in Oracle Restart mode.
- After adding an HAVIP resource, an SMB Export resource must also be added; otherwise, the HAVIP resource does not start.
- For highest performance and best results, ensure that both server and client are using SMB3. Note the following:
  - Use the newest version of Samba, v4 or later.
  - Use the latest Microsoft OS version (2012 or later). To check the SMB version, use the Powershell `cmdlet Get-SmbConnection` command.
  - Previous versions of SMB require that the client must remount the SMB export after a storage failure.
- Similar to HANFS, options may be specified on the command line and are passed to the host operating system. Appropriate error messages are passed back. If no options are provided to the SRVCTL command, the following default options apply:
  - Linux, Solaris, and AIX: Read Only, Browsable = True
- Supported Option Sets:
  - Linux, Solaris, or AIX: Any options supported by the Samba configuration stanza.

To set up High Availability SMB for Oracle Grid Infrastructure, perform the following steps:

1. Add and register a new HAVIP resource.

   For example:

   ```
   # srvctl add havip -id hrexports -address my_havip_name
   ```

   In the example, *`my_havip_name`* is mapped in the domain name server (DNS) to the VIP address and is used by the client systems when mounting the file system.

   The initial processing of `srvctl add havip` ensures that:

   - The address being used is static, not dynamic
   - Any DNS names resolve to only one host, not round-robin multiple DNS resolutions
   - The network resource and provided IP address and resolved name are in the same subnet
   - The name is not in use

   SRVCTL creates the appropriate HAVIP name using the *`id`*, ensuring it is unique. As a final validation step, SRVCTL ensures that the *`network`* resource (if provided) of `ora.net#.network` exists. After this step, SRVCTL adds a new `havip` of type `ora.havip.type` with the name of `ora.`*`id`*`.havip`. In this example, the name is `ora.`*`hrexports`*`.havip`.

   Next SRVCTL modifies HAVIP start dependencies, such as `active dispersion`; sets the stop dependencies; and ensures the *`description`* attribute (if provided) is appropriately set.

2. Create a shared Oracle ACFS file system.

High Availability SMB for Oracle Grid Infrastructure operates only with Oracle ACFS file systems configured for clusterwide accessibility and does not support Oracle ACFS file systems configured for access on particular subsets of cluster nodes. High Availability NFS is not supported with non-Oracle ACFS file systems.

3. Register the Oracle ACFS file system.

   For example:

   ```
   $ srvctl add filesystem -device /dev/asm/d1volume1-295 -volume VOLUME1 \
     -diskgroup HR_DATA -mountpath /oracle/cluster1/acfs1
   ```

4. Create an Oracle ACFS file system export resource.

   For example:

   ```
   # srvctl add exportfs -id hrexports -path /oracle/cluster1/acfs1 -name hrexport1 -
   type SMB
   ```

   After the file system export resource has been created, then you can start the HAVIP created in step 1 to export the file system using the `srvctl start havip` command.

   During the start of the resource, the Oracle ACFS Export resource creates a Samba configuration file (Linux, Solaris, or AIX) or runs the `net.exe` binary to export the file system.

   VIPs attempts to find the best server to run on based on available file systems and other running VIPs, but this operation only occurs during CSS membership change events, such as a node joining or leaving the cluster.

   Note:

   - It is not recommended to start and stop exports individually; this functionality should be provided through the start and stop operations of HAVIP.
   - When HAVIP is not running, exports can exist on different nodes. After the associated HAVIP is started, the exports gather on a single node.

See Also:

- Creating an Oracle ACFS File System for information about creating an Oracle ACFS file system
- *Oracle Clusterware Administration and Deployment Guide* for information about the `srvctl add filesystem` command
- *Oracle Clusterware Administration and Deployment Guide* for information about Oracle Clusterware resources
- *Oracle Clusterware Administration and Deployment Guide* and *Oracle Real Application Clusters Administration and Deployment Guide* for information about SRVCTL commands

Overview of Oracle ASM Dynamic Volume Manager

Oracle ASM Dynamic Volume Manager (Oracle ADVM) provides volume management services and a standard disk device driver interface to clients.

File systems and other disk-based applications send I/O requests to Oracle ADVM volume devices as they would to other storage devices on a vendor operating system.

An Oracle ADVM volume device is constructed from an Oracle ASM dynamic volume. One or more Oracle ADVM volume devices may be configured within each Oracle ASM disk group. The Oracle ADVM Driver maps I/O requests against an Oracle ADVM volume device to blocks in a corresponding Oracle ASM dynamic volume and disk set located within an Oracle ASM disk group. An Oracle ADVM volume device exports Oracle ASM volume manager features and ensures that volume mirrors remain consistent in the face of irregular system shutdowns, Oracle ASM instance failures, or system failures.

Oracle ADVM extends Oracle ASM by providing a disk driver interface to Oracle ASM storage allocated as Oracle ADVM volume files. You can use Oracle ADVM to create virtual disks that contain file systems. These file systems contained on Oracle ADVM volumes are able to support files beyond Oracle Database files, such as executable files, report files, trace files, alert logs, and other application data files. Because Oracle ADVM volumes are actually Oracle ASM files, they require the same administrative privileges as the Oracle ASM files.

Oracle Advanced Cluster File System (Oracle ACFS) communicates with Oracle ASM through the Oracle ADVM interface. With the addition of the Oracle ADVM, Oracle ASM becomes a complete storage solution of user data for both database and non-database file needs.

To add a volume to an Oracle ASM disk group, disk group attributes `COMPATIBLE.ASM` and `COMPATIBLE.ADVM` must be set to `'11.2'`.

> **Note:**
>
> Dynamic volumes supersede traditional device partitioning. Each volume is individually named and may be configured for a single file system. Oracle ADVM volumes may be created on demand from Oracle ASM disk group storage and dynamically resized as required. These attributes make Oracle ADVM volumes far more flexible than physical devices and associated partitioning schemes.

See Also:

- Managing Oracle ADVM with ASMCMD for information about ASMCMD command-line tools for managing Oracle ADVM volumes
- Using Views to Display Oracle ACFS Information for information about `V$ASM` views to display information about Oracle ADVM volumes
- Managing Oracle ACFS with Command-Line Tools for information about Oracle ACFS operating system utilities for managing file systems and volume device files
- *Oracle Automatic Storage Management Administrator's Guide* for information about the `ALTER DISKGROUP ADD VOLUME` SQL statement to administer volumes
- *Oracle Database SQL Language Reference* for information about the `ALTER DISKGROUP` SQL statement

2 Using Views to Display Oracle ACFS Information

Dynamic views display important information about Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM).

This chapter contains information about using dynamic views to display Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM) information.

- Views Containing Oracle ACFS Information
- Oracle ACFS Support for Oracle Database File Mapping Views

> **See Also:**
>
> - *Oracle Database Reference* for information about all of the `V$ASM_ACFS*` dynamic performance views
> - Oracle ACFS Command-Line Utilities for `acfsutil info` commands that display Oracle ACFS information

Views Containing Oracle ACFS Information

You can use the views in Table 2-1 to obtain information about Oracle Advanced Cluster File System (Oracle ACFS). These views are accessible from the Oracle ASM instance.

> **Note:**
>
> To display information about Oracle ACFS file systems or volumes that are located on nodes in an Oracle Flex ASM configuration, you must connect to the Oracle ASM proxy instance instead of the local Oracle ASM instance.

Table 2-1 Oracle ASM dynamic views for Oracle ACFS information

View	Description
`V$ASM_ACFS_ENCRYPTION_INFO`	Contains encryption information for each Oracle ACFS file system.
`V$ASM_ACFS_SECURITY_INFO`	Contains security realm information for each Oracle ACFS file system.
`V$ASM_ACFS_SEC_ADMIN`	Contains information about Oracle ACFS security administrator in the cluster.
`V$ASM_ACFS_SEC_CMDRULE`	Contains information about Oracle ACFS security command rules for each Oracle ACFS file system.

Table 2-1 (Cont.) Oracle ASM dynamic views for Oracle ACFS information

View	Description
V$ASM_ACFS_SEC_REALM	Contains information about every realm in the Oracle ACFS security realm for each Oracle ACFS file system.
V$ASM_ACFS_SEC_REALM_FILTER	Contains information about every filter in the Oracle ACFS security realm for each Oracle ACFS file system. A filter is a defined as `commandrule`/`ruleset` pair in a realm.
V$ASM_ACFS_SEC_REALM_GROUP	Contains information about every group in the Oracle ACFS security realm for each Oracle ACFS file system.
V$ASM_ACFS_SEC_REALM_USER	Contains information about every user in the Oracle ACFS security realm for each Oracle ACFS file system.
V$ASM_ACFS_SEC_RULE	Contains information about every Oracle ACFS security rule for each Oracle ACFS file system.
V$ASM_ACFS_SEC_RULESET	Contains information about every Oracle ACFS security ruleset for each Oracle ACFS file system.
V$ASM_ACFS_SEC_RULESET_RULE	Contains information about every rule in Oracle ACFS security ruleset for each Oracle ACFS file system.
V$ASM_ACFSREPL	Contains information for Oracle ACFS file systems that are initialized for replication. For Oracle ASM releases 12.2 or higher, this view contains records just for storage locations that are file systems (and not for ones that are snapshots)
V$ASM_ACFSREPLTAG	Contains replicated tag information for Oracle ACFS file systems that are initialized for replication. This view only contains records for Oracle ASM releases prior to 12.2. To display Oracle ACFS replication information for Oracle ASM releases 12.2 or higher, use the `acfsutil repl info` command.
V$ASM_ACFSSNAPSHOTS	Contains snapshot information about every mounted Oracle ACFS file system.
V$ASM_ACFSTAG	Contains all tag name information about files on all mounted Oracle ACFS file systems
V$ASM_ACFSVOLUMES	Contains information about mounted Oracle ADVM volumes, correlated with `V$ASM_FILESYSTEM`.
V$ASM_FILESYSTEM	Contains information about every mounted Oracle ACFS file system.
V$ASM_VOLUME	Contains information about each Oracle ADVM volume that is a member of an Oracle ASM instance.
V$ASM_VOLUME_STAT	Contains information about statistics for each Oracle ADVM volume.

Example 2-1 Viewing encryption information in V$ASM_ACFS_ENCRYPTION_INFO

This example shows information displayed from the `V$ASM_ACFS_ENCRYPTION_INFO` view. The `FS_NAME` column contains the mount point. The `VOL_DEVICE` contains the name of the Oracle ADVM device.

```
SELECT SUBSTR(fs_name,1,24) FILESYSTEM, SUBSTR(vol_device,1,20) DEVICE, set_status,
   enabled_status, algorithm, key_length, last_rekey_time FROM V$ASM_ACFS_ENCRYPTION_INFO;

FILESYSTEM               DEVICE                 SET_STA ENABLED_ ALGORIT KEY_LEN LAST_REKE
------------ ----------- ---------------------- ------- -------- ------- ------- ---------
/acfsmounts/acfs1        /dev/asm/volume1-228   YES     ENABLED  AES     192
```

Example 2-2 Viewing security information in V$ASM_ACFS_SECURITY_INFO

This example shows information displayed from the `V$ASM_ACFS_SECURITY_INFO` view. The `FS_NAME` column contains the mount point. The `VOL_DEVICE` contains the name of the Oracle ADVM device.

```
SELECT SUBSTR(fs_name,1,24) FILESYSTEM, SUBSTR(vol_device,1,20) DEVICE, prepared_status,
     enabled_status FROM V$ASM_ACFS_SECURITY_INFO;

FILESYSTEM                DEVICE                PREPARE ENABLED_
------------------------- --------------------- ------- --------
/acfsmounts/acfs1        /dev/asm/volume1-228    YES     ENABLED
```

Example 2-3 Viewing security rules information in V$ASM_ACFS_SEC_RULE

This example shows information displayed from the `V$ASM_ACFS_SEC_RULE` view. The `FS_NAME` column contains the mount point.

```
SELECT SUBSTR(rule_name,1,24) rule, SUBSTR(rule_type,1,12) type,
    SUBSTR(rule_value,1,16) value, SUBSTR(fs_name,1,24) filesystem
    FROM V$ASM_ACFS_SEC_RULE;

RULE                     TYPE         VALUE            FILESYSTEM
------------------------ ------------ ---------------- ------------------------
medHistRule1a            TIME         22:00:00         /acfsmounts/acfs1
medHistRule1c            TIME         08:00:00         /acfsmounts/acfs1
medHistRule1b            USERNAME     medMaintenance   /acfsmounts/acfs1
medHistRule1d            USERNAME     medBrowse        /acfsmounts/acfs1
SYSTEM_RULE_Auditor      GROUPNAME    myaudit_mgr_grp  /acfsmounts/acfs1
SYSTEM_RULE_AuditManager GROUPNAME    myauditor_grp    /acfsmounts/acfs1
SYSTEM_RULE_Always       TIME         00:00:00         /acfsmounts/acfs1
```

Example 2-4 Viewing security ruleset information in V$ASM_ACFS_SEC_RULESET

This example shows information displayed from the `V$ASM_ACFS_SEC_RULESET` view. The `FS_NAME` column contains the mount point.

```
SELECT SUBSTR(ruleset_name,1,36) ruleset, ruleset_option r_option, SUBSTR(fs_name,1,24) filesystem
    FROM V$ASM_ACFS_SEC_RULESET;

RULESET                              R_OPTION FILESYSTEM
------------------------------------ -------- -----------------
medRuleSet1                          ALL_TRUE /acfsmounts/acfs1
medRuleSet2                          ALL_TRUE /acfsmounts/acfs1
SYSTEM_RULESET_Auditor               ALL_TRUE /acfsmounts/acfs1
SYSTEM_RULESET_AuditManager          ALL_TRUE /acfsmounts/acfs1
SYSTEM_RULESET_AuditMgr_Auditor      ANY_TRUE /acfsmounts/acfs1
SYSTEM_RULESET_AlwaysDeny            ANY_TRUE /acfsmounts/acfs1
```

Example 2-5 Viewing security ruleset information in V$ASM_ACFS_SEC_RULESET_RULE

This example shows information displayed from the `V$ASM_ACFS_SEC_RULESET_RULE` view.

```
SELECT SUBSTR(ruleset_name,1,36) ruleset, substr(rule_name,1,24) rule,
    SUBSTR(fs_name,1,36) filesystem FROM V$ASM_ACFS_SEC_RULESET_RULE;

RULESET                              RULE                     FILESYSTEM
------------------------------------ ------------------------ ------------------
medRuleSet1                          medHistRule1a            /acfsmounts/acfs1
medRuleSet1                          medHistRule1b            /acfsmounts/acfs1
medRuleSet2                          medHistRule1c            /acfsmounts/acfs1
```

```
medRuleSet2                         medHistRule1d            /acfsmounts/acfs1
SYSTEM_RULESET_Auditor              SYSTEM_RULE_Auditor      /acfsmounts/acfs1
SYSTEM_RULESET_AuditManager         SYSTEM_RULE_AuditManager /acfsmounts/acfs1
SYSTEM_RULESET_AuditMgr_Auditor     SYSTEM_RULE_Auditor      /acfsmounts/acfs1
SYSTEM_RULESET_AuditMgr_Auditor     SYSTEM_RULE_AuditManager /acfsmounts/acfs1
SYSTEM_RULESET_AlwaysDeny           SYSTEM_RULE_Always       /acfsmounts/acfs1
```

Example 2-6 Viewing snapshot information in V$ASM_ACFSSNAPSHOTS

This examle shows information displayed from the `V$ASM_ACFSSNAPSHOTS` view. The `FS_NAME` column contains the mount point. The `VOL_DEVICE` contains the name of the Oracle ADVM device.

```
SELECT SUBSTR(FS_NAME,1,24) FILESYSTEM, SUBSTR(VOL_DEVICE,1,22) DEVICE,
  SUBSTR(SNAP_NAME,1,12) SNAPSHOT, CREATE_TIME TIME, SUBSTR(PARENT,1,10) PARENT,
  SUBSTR(TYPE,1,4) TYPE FROM V$ASM_ACFSSNAPSHOTS;

FILESYSTEM                DEVICE                 SNAPSHOT     TIME      PARENT     TY
------------------------ ---------------------- ------------ --------- ---------- --
/acfsmounts/acfs1         /dev/asm/volume1-229   snaprw       13-MAR-12 NULL       RW
/acfsmounts/acfs1         /dev/asm/volume1-229   snaprw_child 13-MAR-12 snaprw     RW
/acfsmounts/acfs2         /dev/asm/volume2-321   snapro       13-MAR-12 NULL       RO
```

Example 2-7 Viewing volume information with V$ASM_ACFSVOLUMES

This example shows information displayed from the `V$ASM_ACFSVOLUMES` view. The `PRIMARY_VOL` column contains `TRUE` if the volume is the primary volume for the file system.

```
SELECT fs_name, vol_device, primary_vol, total_mb, free_mb FROM V$ASM_ACFSVOLUMES;

FS_NAME             VOL_DEVICE             PRIMARY_VOL   TOTAL_MB  FREE_MB
------------------ ---------------------- -----------   --------- ----------
/acfsmounts/acfs1  /dev/asm/volume1-228   TRUE            1024000 578626.522
/acfsmounts/acfs2  /dev/asm/volume2-375   TRUE            1024000 685761.463
...
```

Example 2-8 Viewing volume information with V$ASM_FILESYSTEM

This example shows information displayed from the `V$ASM_FILESYSTEM` view.

The `STATE` column contains the status of the file system, either `AVAILABLE` or `OFFLINE`. An offline file system can only be dismounted; other attempts at access result in errors. Offline means that either the Oracle ASM instance is down, the disk group has been forced dismounted, or less commonly, a metadata I/O failure occurred or serious metadata corruption was detected. With a metadata I/O failure, the file system is also marked as corrupt.

The `CORRUPT` column indicates whether the file system needs the `fsck` command run on it.

```
SELECT fs_name, available_time, block_size, state, corrupt FROM V$ASM_FILESYSTEM;

FS_NAME                     AVAILABLE BLOCK_SIZE STATE         CORRUPT
-------------------------- --------- ---------- ------------- -------
/acfsmounts/acfs1           19-JUL-09          4 AVAILABLE     FALSE
/acfsmounts/acfs2           19-JUL-09          4 AVAILABLE     FALSE
```

Example 2-9 Viewing volume information with V$ASM_VOLUME

This example shows information displayed from the `V$ASM_VOLUME` view for volumes contained in the `DATA` disk group.

```
SELECT dg.name AS diskgroup, v.volume_name, v.volume_device, v.mountpath
    FROM V$ASM_DISKGROUP dg, V$ASM_VOLUME v
    WHERE dg.group_number = v.group_number and dg.name = 'DATA';

DISKGROUP        VOLUME_NAME       VOLUME_DEVICE           MOUNTPATH
---------------- ----------------- ----------------------- ------------------
DATA             VOLUME1           /dev/asm/volume1-228    /acfsmounts/acfs1
DATA             VOLUME2           /dev/asm/volume2-375    /acfsmounts/acfs2
```

Example 2-10 Viewing volume information with V$ASM_VOLUME_STAT

This example shows information displayed from the `V$ASM_VOLUME_STAT` view for volumes contained in the `DATA` disk group. The `BYTES_READ` column contains the total number of bytes read for the volume. The `BYTES_WRITTEN` column contains the total number of bytes written for the volume.

```
SELECT dg.name AS diskgroup, v.volume_name, v.bytes_read, v.bytes_written
    FROM V$ASM_DISKGROUP dg, V$ASM_VOLUME_STAT v
    WHERE dg.group_number = v.group_number and dg.name = 'DATA';

DISKGROUP                      VOLUME_NAME                    BYTES_READ  BYTES_WRITTEN
------------------------------ ------------------------------ ----------- -------------
DATA                           VOLUME1                        12370105856      43510272
DATA                           VOLUME2                            2685728      32201504
```

Example 2-11 Viewing tag name information with V$ASM_ACFSTAG

This example shows tag names for the `/acfsmounts/acfs1` file system displayed from the `V$ASM_ACFSTAG` view.

```
SELECT SUBSTR(TAG_NAME,1,8) TAG_NAME, SUBSTR(FS_NAME,1,20) FS_NAME,
     SUBSTR(PATH_NAME,1,42) PATH_NAME FROM V$ASM_ACFSTAG WHERE TAG_NAME='tag5';

TAG_NAME FS_NAME              PATH_NAME
-------- --------------------- --------------------------------------
tag5     /acfsmounts/acfs1     /acfsmounts/acfs1/d1/d2/d3/d4/d5/f6
tag5     /acfsmounts/acfs1     /acfsmounts/acfs1/d1/d2/d3/d4/d5
tag5     /acfsmounts/acfs1     /acfsmounts/acfs1/d1/d2/d3/d4
tag5     /acfsmounts/acfs1     /acfsmounts/acfs1/d1/d2/d3
tag5     /acfsmounts/acfs1     /acfsmounts/acfs1/d1/d2
tag5     /acfsmounts/acfs1     /acfsmounts/acfs1/d1
tag5     /acfsmounts/acfs1     /acfsmounts/acfs1/f1
tag5     /acfsmounts/acfs1     /acfsmounts/acfs1
```

See Also:

Oracle Database Reference for information about the Oracle ACFS dynamic views

Oracle ACFS Support for Oracle Database File Mapping Views

Oracle ACFS supports Oracle Database file mapping views to the Oracle ASM device level.

Note:

This feature is available starting with Oracle Database 12c Release 1 (12.1.0.2).

The following database mapping views are supported by Oracle ACFS:

- `V$MAP_FILE`
- `V$MAP_FILE_EXTENT`
- `V$MAP_ELEMENT`
- `V$MAP_FILE_IO_STACK`

These `V$MAP` views are only refreshed by executing the procedure `DBMS_STORAGE_MAP.MAP_ALL`. The Oracle ACFS file mapping interface does not utilize the external `fmputl` process or its supporting libraries.

Note:

Oracle ACFS does not provide support for the `V$MAP_SUBELEMENT` view.

Before running any queries on the `V$MAP` views, ensure that the `FILE_MAPPING` initialization is set to `TRUE`, then run the `DBMS_STORAGE_MAP.MAP_ALL` procedure to build the mapping information for the entire I/O subsystem associated with the database. For example, connect as `SYSDBA` to the database instance and run the following:

```
SQL> ALTER SYSTEM SET file_mapping=true;

SQL> EXEC DBMS_STORAGE_MAP.MAP_ALL(10000);
```

The SQL statements in Example 2-12 to Example 2-15 are run from the Oracle Database instance.

Example 2-12 Viewing Oracle ASM information with V$MAP_ELEMENT

This example displays information from the `V$MAP_ELEMENT` view.

```
SQL> SELECT ELEM_NAME, ELEM_IDX, ELEM_TYPE, ELEM_SIZE, ELEM_DESCR 
       FROM V$MAP_ELEMENT;

ELEM_NAME      ELEM_IDX ELEM_TYPE   ELEM_SIZE ELEM_DESCR
------------ ---------- ---------- ---------- -----------------
+/dev/xvdd1           0 ASMDISK     117184512 TEST_0001
+/dev/xvdc1           1 ASMDISK     117184512 TEST_0000
```

Example 2-13 Viewing Oracle ACFS Data File Information with V$MAP_FILE

This example displays information from the `V$MAP_FILE` view.

```
SQL> SELECT FILE_NAME, FILE_MAP_IDX, FILE_TYPE, FILE_STRUCTURE, FILE_SIZE,
      FILE_NEXTS FROM V$MAP_FILE WHERE REGEXP_LIKE(FILE_NAME, '*users01.dbf');

FILE_NAME                 FILE_MAP_IDX FILE_TYPE FILE_STRU FILE_SIZE FILE_NEXTS
```

```
------------------------ ------------ --------- --------- --------- ----------
/dbdata1/orcl/users01.dbf            4 DATAFILE  FILE          10256         41
```

Example 2-14 Viewing Element and File Offset Information with V$MAP_FILE_EXTENT

This example displays the element offset versus file offset information for each extent with `V$MAP_FILE_EXTENT`, specifying `FILE_MAP_IDX` equal to `4`, which is the file map index of the `/dbdata/orcl/users01.dbf` file.

```
SQL> SELECT FILE_MAP_IDX, EXT_NUM, EXT_ELEM_OFF, EXT_SIZE, EXT_FILE_OFF,
       EXT_TYPE, ELEM_IDX FROM V$MAP_FILE_EXTENT WHERE FILE_MAP_IDX=4;

FILE_MAP_IDX    EXT_NUM EXT_ELEM_OFF   EXT_SIZE EXT_FILE_OFF EXT_TY   ELEM_IDX
------------ ---------- ------------ ---------- ------------ ------ ----------
           4          0     58105664        192            0 DATA            0
           4          1     58154752        256          192 DATA            1
           4          2     58089472        256          448 DATA            0
...
           4         39     58140928        256         9920 DATA            1
           4         40     58108160         88        10176 DATA            0
41 rows selected.
```

Example 2-15 Viewing Extent Information With V$MAP_FILE_IO_STACK

This example displays information from `V$MAP_FILE_IO_STACK` specifying `FILE_MAP_IDX` equal to `4`. The `V$MAP_FILE_IO_STACK` view is similar to `V$MAP_FILE_EXTENT`, but the display groups contiguous extents which are on the same device or element and of the same size.

```
SQL> SELECT FILE_MAP_IDX, ELEM_IDX, CU_SIZE,STRIDE, NUM_CU,ELEM_OFFSET,
       FILE_OFFSET FROM V$MAP_FILE_IO_STACK WHERE FILE_MAP_IDX=4;

FILE_MAP_IDX   ELEM_IDX    CU_SIZE     STRIDE     NUM_CU ELEM_OFFSET FILE_OFFSET
------------ ---------- ---------- ---------- ---------- ----------- -----------
           4          0        256       1024         10    58089472 448
           4          0        192          0          1    58105664 0
           4          0        256       1024          9    58105856 960
           4          0         88          0          1    58108160 10176
           4          1        256       1024         10    58138624 704
           4          1        256       1024         10    58154752 192

6 rows selected.
```

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about file mapping to Oracle ASM files
- *Oracle Database Administrator’s Guide* for information about the Oracle Database File Mapping Interface
- *Oracle Database Reference* for details about the `V$MAP` views

3

Administering Oracle ACFS with Oracle Enterprise Manager

Oracle Enterprise Manager Cloud Control provides tools for administering Oracle Advanced Cluster File System (Oracle ACFS).

This chapter describes how to administer Oracle Advanced Cluster File System (Oracle ACFS) with Oracle Enterprise Manager Cloud Control.

All Oracle ASM administration tasks begin with the Oracle Automatic Storage Management home page in Oracle Enterprise Manager Cloud Control.

> **Note:**
>
> To manage or monitor Oracle ACFS file systems or volumes that are located on nodes in an Oracle Flex ASM configuration, you must connect to the Oracle ASM proxy instance instead of the local Oracle ASM instance.

This chapter contains the following topics:

- Accessing the Oracle ASM and Oracle ACFS Home Page
- Managing Oracle ACFS File Systems and Volumes with Oracle Enterprise Manager
- Managing Oracle ACFS Snapshots with Oracle Enterprise Manager
- Managing Encryption Features with Oracle Enterprise Manager
- Managing Tagging Features with Oracle Enterprise Manager

> **See Also:**
>
> - Managing Oracle ACFS with Command-Line Tools for information about using command-line tools to administer Oracle ACFS file systems
> - Basic Steps to Manage Oracle ACFS Systems for information about the basic steps for creating Oracle ACFS file systems
> - *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle Flex ASM

Accessing the Oracle ASM and Oracle ACFS Home Page

All Oracle ASM administration tasks begin with the Oracle Automatic Storage Management home page in Oracle Enterprise Manager Cloud Control. The Oracle Automatic Storage Management home page displays:

- The status of the Oracle ASM instance.
- A chart that shows the used and free space of each disk group and disk group internal usage.
- A list of databases that are serviced by the Oracle ASM instance.
- A list of Oracle Advanced Cluster File System (Oracle ACFS) file systems that are serviced by the Oracle ASM instance.
- A list of other non-Oracle ACFS volumes.
- A list of alerts for the Oracle ASM instance and the host computer.
- Links to the Oracle ASM Performance, Disk Groups, Configuration, Users, and Oracle ACFS pages.

To access the Oracle Automatic Storage Management home page on a single-instance system:

1. Log in to Oracle Enterprise Manager Cloud Control.
2. Under the **Targets** drop down menu at the top of the page, select **All Targets**.
3. In the lists of targets on the **All Targets** page, click the target for the Oracle ASM instance.
4. If prompted for Oracle ASM login credentials, then enter the user `SYS`, provide the `SYS` password that was set for the Oracle ASM instance during installation, and connect as `SYSASM`. The Oracle Automatic Storage Management home page displays.

> **See Also:**
>
> - Managing Oracle ACFS with Command-Line Tools for information about using command-line tools to administer Oracle ACFS file systems
> - Basic Steps to Manage Oracle ACFS Systems for information about the basic steps for creating Oracle ACFS file systems
> - *Oracle Automatic Storage Management Administrator's Guide* for additional information about the Oracle ASM home page

Managing Oracle ACFS File Systems and Volumes with Oracle Enterprise Manager

This section discusses how to manage Oracle ACFS file systems and volumes systems with Oracle Enterprise Manager Cloud Control.

This section contains the following topics:

- Creating Oracle ACFS File Systems and Volumes
- Viewing and Modifying Oracle ACFS Volumes and File Systems

For more information about Oracle ACFS file systems and volumes, refer to "About Oracle ACFS".

> **See Also:**
>
> *Oracle Enterprise Manager Cloud Control Administrator's Guide*

Creating Oracle ACFS File Systems and Volumes

To create an Oracle ACFS volume, perform the following steps.

1. Access the **Disk Groups** page from the Oracle ASM home page.
2. Click the **Volumes** link in the **General** tab of the **Disk Group** page.
3. Click **Create** in the **Volumes** tab of the **Disk Group** page.

   The **Create ASM Volume** page displays.

   Enter the volume name, disk group name that contains the volume, and the initial size of the volume.

   You can also select the redundancy setting for the volume and the region settings for primary and mirror extents.

To create an Oracle ACFS file system on a volume in a disk group, perform the following steps.

1. Access the **Volumes** tab of the **Disk Group** page.
2. Select an existing volume in the disk group and click **Create ASM Cluster File System**.

   The compatibility parameters `COMPATIBLE.ASM` and `COMPATIBLE.ADVM` must be set to `11.2` or higher for the disk group..
3. On the **Create ASM Cluster File System** page, enter the information to create a file system. You can optionally register and mount the file system.

   You must enter the volume name. You can enter an optional volume label.

   You can optionally choose to register and mount the file system when it is created. Select a mount point from available directories.

   Enter the host credentials needed to run the command when prompted. To register or mount a file system, you need root or administrator privileges.

   After a directory has been selected, click **Show Command** to have Oracle Enterprise Manager generate the commands that you can run at an operating system prompt. Examples of commands on a Linux system are:

```
/sbin/mkfs -t acfs -b 4k /dev/asm/volume1-361
/sbin/acfsutil registry -f -a /dev/asm/volume1-361 /oracle/acfsmounts/acfs2
/bin/mount -t acfs -rw /dev/asm/volume1-361 /oracle/acfsmounts/acfs2
```

   You can also generate the commands to register or mount an Oracle ACFS file system on the **ASM Cluster File System** tab.

See Also:

- ASMCMD Volume Management Commands for information about creating a volume with the ASMCMD `volcreate` command
- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- About the Oracle ACFS Mount Registry for information about the registry process

Viewing and Modifying Oracle ACFS Volumes and File Systems

To view and modify information about Oracle ACFS file systems, click the **ASM Cluster File System** link in the Oracle ASM home page.

The **ASM Cluster File System** tab in Oracle Enterprise Manager lists all of the Oracle ACFS associated with the Oracle ASM instance.

On this page, you can choose to mount, dismount, delete, create snapshot, view content, register, and deregister a selected file system. In addition, you can create a file system, mount all file systems, or dismount all file systems.

For each Oracle ACFS, the columns provide information for Mount Point, Availability, State, Snapshots, Used (%), Used (GB), Size (GB), Allocated Space, Redundancy, Volume, and Disk Group. Redundancy, Volume, and Disk Group columns are on the far right of the page and are not shown in the illustration. The mount point, snapshots, volume, and disk group are provided as links to further information.

To view information about a specific volume, click the volume name in the **Volume** column on the **ASM Cluster File System** page to display the **General** tab of the ASM **Volumes** page.

To view information about a file system, click the link in the **Mount Point** column on the **ASM Cluster File System** page.

To view information about a disk group for an Oracle ACFS, click the disk group name in the **Disk Group** column on the **ASM Cluster File System** page. The **General** tab of the **Disk Group** page displays.

To view information about volumes in a disk group, click the **Volumes** tab at the top of the **Disk Group** page.

Managing Oracle ACFS Snapshots with Oracle Enterprise Manager

This section describes how to manage snapshots with Oracle Enterprise Manager.

- Creating, Modifying, and Viewing Snapshots
- Converting Snapshots Between Read-Only and Read-Write
- Creating Child Snapshots from Existing Snapshots

For more information about Oracle ACFS snapshots, refer to "About Oracle ACFS Snapshots".

Creating, Modifying, and Viewing Snapshots

To view and modify information about existing snapshots or create snapshots for a file system, perform the following steps.

1. Click the **ASM Cluster File System** tab on the Oracle ASM home page
2. Click a mount point link in the **Mount Point** column.
3. Click the **Snapshots** tab.

Optionally, you can click a number link for existing snapshots in the **Snapshots** column on the **ASM Cluster File System** page to display the **Snapshots** page.

On the **Snapshots** page, you can create snapshots or search for and display specific snapshots. To search for a snapshot, enter a name with optional wildcard characters in the search field then click **Search**.

To create a snapshot, perform the following steps.

1. Click **Create** in the **Snapshots** page.
2. Complete the information on the **Create Snapshot** page.

   Accept the default snapshot name, or provide a name. Optionally, you can choose to delete the oldest snapshot.
3. When you have completed the screen, you can click **OK** to run the command, or click **Show Command** to view the generated command.

   For example, the following is a generated command for creating a snapshot:

   ```
   /sbin/acfsutil snap create "snapshot_20090702_142135" /oracle/acfsmounts/acfs1
   ```

   To run the generated command, you need the appropriate privileges on the host computer. Oracle Enterprise Manager prompts you to enter host credentials if they have not been set up.

   You can also open a Telnet session from the **Create Snapshot** page to manually enter the generated operating system command.

To drill down in a snapshot directory, click the snapshot name in the **Snapshots** page to display the **Search and List Directory** page.

Converting Snapshots Between Read-Only and Read-Write

You can use edit snapshot to convert the snapshot attribute between Read Only and Read Write. To edit a snapshot, follow these steps:

1. From the Oracle ASM home page, select the ASM Cluster File System tab to display the mount points in the cluster file system.
2. Click the file or directory in which you want to edit a snapshot.

   Enterprise Manager displays ASM Cluster File System page.
3. Click the **Snapshots** tab.
4. Select the snapshot you want to edit, then click **Edit**.

The Snapshot page displays.

5. Change the Attribute. You can choose **Read Only** or **Read Write**.
6. Click **OK**.

Creating Child Snapshots from Existing Snapshots

Use the Create Child snapshot operation to create a point-in-time copy of an existing Oracle ACFS snapshot. Future changes to the parent snapshot are not inherited by the child snapshot. To create a child snapshot of an existing snapshot, follow these steps:

1. From the Oracle ASM home page, select the ASM Cluster File System tab to display the mount points in the cluster file system.
2. Click the file or directory Mount Point in which you want to create a child snapshot.

   Enterprise Manager displays ASM Cluster File System page.
3. Click the **Snapshots** tab.
4. Select the parent snapshot you want to use to create a child snapshot, then click **Create Child**.

   The Create Child Snapshot on ASM Cluster File System page displays.
5. The Snapshot Name field is automatically populated with a default child snapshot name. You can replace the name with a name you designate.
6. Set the Attribute to **Read Only** or **Read Write**.
7. Optionally you can turn on the option to delete the oldest snapshot by turning on the delete option. Oracle Enterprise Manager tabulates the number of remaining snapshots before the maximum limit is reached.
8. Click **OK** to create the child snapshot.

   Oracle Enterprise Manager returns to the Snapshots tab where you can see the new snapshot with its Parent Name listed in the table.

Managing Encryption Features with Oracle Enterprise Manager

This section describes how to manage encryption with Oracle Enterprise Manager.

- Initializing Oracle ACFS Encryption
- Enabling, Disabling, and Setting Parameters for Encryption On an Oracle ACFS
- Viewing Encryption Status

For more information about Oracle ACFS encryption, refer to "Oracle ACFS Encryption".

Initializing Oracle ACFS Encryption

To initialize Oracle ACFS encryption, follow these steps:

1. From the Oracle ASM home page, select the ASM Cluster File System tab to display the mount points in the cluster file system.

2. Click a mount point in the list.
3. Click the **Security/Encryption** tab to display the Security and Encryption section.
4. In the **Encryption Configuration** section, you can specify various options, such as **Unset Encryption Parameters**.
5. If Oracle ACFS encryption is not initialized in the cluster, the **Initialize Encryption** button displays. Click the button to initialize Oracle ACFS encryption. This operation needs to be performed only once in the cluster.

Enabling, Disabling, and Setting Parameters for Encryption On an Oracle ACFS

To enable, disable, or set parameters for Oracle ACFS encryption, follow these steps:

1. From the Oracle ASM home page, select the ASM Cluster File System tab to display the mount points in the cluster file system.
2. Click the **Mount Point** that contains the realms you want to view. The Oracle ACFS home page is displayed for this mount point.
3. Click the **Security/Encryption** tab.
4. Click **Encryption Configuration** link to display the Encryption configuration section.

Viewing Encryption Status

To view encryption status, follow these steps:

1. From the Oracle ASM home page, select the ASM Cluster File System tab to display the mount points in the cluster file system.
2. Click **Show Security and Encryption** to display the Security and Encryption section.

   The section displays the mount points and the security and encryption settings for each.

Managing Tagging Features with Oracle Enterprise Manager

This section describes how to manage tagging with Oracle Enterprise Manager.

- Adding a Tag in Oracle ACFS
- Removing a Tag in Oracle ACFS
- Searching for Tags in Oracle ACFS

For more information about Oracle ACFS tagging, refer to "Oracle ACFS Tagging".

Adding a Tag in Oracle ACFS

You can add tags to directories and files that reside in a single Oracle ACFS file system or across multiple Oracle ACFS file systems. With the tag names, you can organize your files or perform operations, such as search and replication.

To add tags for specific files or directories from the ASM Cluster File System Page, follow these steps:

1. From the Oracle ASM home page, select the ASM Cluster File System tab to display the mount points in the cluster file system.
2. Select the file or directory you want to tag and select **Add Tag** from the Actions menu, then click **Go**.

   Enterprise Manager displays the Add Tag page.
3. Add a Tag Name (or a comma-delimited list of names) in the Tag Name box. Optionally, you can turn on the option that enables you to add tags recursively to subdirectories and files.
4. In the Directories and Files section, enter either the directory paths directly as a comma-delimited list in the **Enter Directory** box, or search for directory paths with the ASM Cluster File System by selecting **Search Directory**. You can add a directory path by clicking **Add**.
5. Click **OK** to add tags.

Removing a Tag in Oracle ACFS

You can remove tags for specific files or directories in an Oracle ACFS file system. Optionally, you can recursively apply the remove operation to all subdirectories and files.

To remove tags, follow these steps:

1. From the Oracle ASM home page, select the ASM Cluster File System tab to display the mount points in the cluster file system.
2. Select the file or directory from which you want to remove tags and select **Remove Tag** from the Actions menu, then click **Go**.

   Enterprise Manager displays the Remove Tag page.
3. In the Tag Name box, enter the name of the tag you want to remove or specify a comma-delimited list of tag names to remove. Optionally, you can remove tags recursively from all directories and files that exist in the specified directory paths by turning on the option.
4. In the Directories and Files section, select **Enter Directory** to enter the directory paths directly as a comma-delimited list, or select **Search Directory** to search for directory paths within an Oracle ACFS file system. To add directory paths repeatedly from different Oracle ACFS file systems, choose the file system and click **Add**.
5. Click **OK** to remove tags.

Searching for Tags in Oracle ACFS

You can search for tags from selected files and directories in an Oracle ACFS file system. Optionally, you can recursively apply this operation to all subdirectories and files. To search for tags for specific files or directories in an Oracle ACFS file system, follow these steps:

1. From the Oracle ASM home page, select the ASM Cluster File System tab to display the mount points in the cluster file system.
2. From the Action menu, select **Search Tags,** then click **Go**.

   Enterprise Manager displays the Search Tags page.

3. In the Search Criteria section, specify a comma-delimited list of tag names in the Tag Name box. Optionally, you can search tags recursively in all directories and files by turning on the option.
4. In the Directories and Files section, select **Enter Directory** to enter the directory path as a comma-delimited list, or select **Search Directory** to choose the Oracle ACFS file system and click **Add** to add directory paths repeatedly from different Oracle ACFS file systems.
5. Click **Search**.

   The results of the search appear in the Search Results table.
6. Click **Return** to navigate back to the Oracle ASM home page displaying the ASM Cluster File System tab.

4 Managing Oracle ACFS and Oracle ADVM With ASMCA

Oracle ASM Configuration Assistant (ASMCA) provides utilities for managing Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM).

Oracle ASM Configuration Assistant (ASMCA) supports installing and configuring Oracle ASM instances, Oracle ASM disk groups, Oracle ASM Dynamic Volume Manager (Oracle ADVM) volumes, and Oracle Advanced Cluster File System (Oracle ACFS) file systems. In addition, you can use the ASMCA command-line interface.

This chapter discusses the following topics:

- ASMCA GUI Tool for Managing Oracle ACFS and Oracle ADVM
- ASMCA Command-Line Interface for Managing Oracle ACFS and Oracle ADVM

> **See Also:**
>
> - *Oracle Automatic Storage Management Administrator's Guide* for information about managing Oracle ASM instances and disk groups with ASMCA

> **Note:**
>
> Some ASMCA functionality and menu options are only available in specific environments, such as an Oracle RAC configuration. For details about a specific ASMCA page, access the online help provided with the **Help** button.

ASMCA GUI Tool for Managing Oracle ACFS and Oracle ADVM

This section contains the following topics:

- Managing Oracle ADVM Volumes with ASMCA
- Managing Oracle ACFS File Systems with ASMCA
- Managing Security and Encryption for Oracle ACFS with ASMCA
- Creating an Oracle ACFS File System for Database Use

Managing Oracle ADVM Volumes with ASMCA

> **Note:**
>
> When creating an Oracle ADVM volume for an Oracle ACFS file system that is intended to store database files, use the ASMCMD `volcreate` command or the SQL `ALTER DISKGROUP ADD VOLUME` SQL statement rather than the ASMCA tool to ensure that the column striping is set to 1.

Oracle ASM Configuration Assistant enables you to create or configure an Oracle ADVM volume.

Right click a selected volume in the Oracle ASM Configuration Assistant screen to display the configuration options menu.

The menu includes options to:

- Enable disabled volumes and disable enabled volumes
- View status details
- Resize volumes that have not been mounted on an Oracle ACFS file system
- Delete volumes

There are options for creating a volume, enabling all volumes, and disabling all volumes.

Click **Create** to display the dialog for creating a volume. You must provide a unique volume name for the existing Oracle ASM disk group that you select from the list of disk groups. You must also specify the size for the volume. You can optionally choose to display advanced options for creating a volume which enables you to specify the redundancy level and striping characteristics.

The compatibility parameters `COMPATIBLE.ASM` and `COMPATIBLE.ADVM` must be set to `11.2` or higher for the disk group to contain an Oracle ADVM volume. To use Oracle ACFS encryption, replication, security, or tagging, the disk group on which the volume is created for the file system must have compatibility attributes for `ASM` and `ADVM` set to `11.2.0.2` or higher.

Before creating an Oracle ADVM volume on AIX, ensure that the necessary user authorizations have been created.

See Also:

- volcreate for information about the advanced options when creating a volume
- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- Oracle ACFS Command-Line Tools for the AIX Environment for information about creating an Oracle ADVM volume on the AIX operating system
- Managing Oracle ADVM with ASMCMD for information about creating a volume with the ASMCMD `volcreate` command
- Introducing Oracle ACFS and Oracle ADVM for an overview of Oracle ACFS
- *Oracle Automatic Storage Management Administrator's Guide* for information about the `ALTER DISKGROUP ADD VOLUME` SQL statement to administer volumes

Managing Oracle ACFS File Systems with ASMCA

Oracle ASM Configuration Assistant enables you to create or configure an Oracle ACFS file system.

To configure an existing file system, right click a selected file system in the Oracle ASM Configuration Assistant screen to display the configuration options menu.

The menu includes options to:

- View status details
- Mount and dismount the file system
- Resize the file system
- Delete the file system
- Display or run the security commands
- Create or delete snapshots

Some commands require root privileges, such as mounting a file system. ASMCA generates the command for you to run manually as `root`.

There are buttons for Create, Mount All, Dismount All, and Security and Encryption commands.

Oracle ASM Configuration Assistant enables you to create an Oracle ACFS file system.

Select **Create**, then select the type of Oracle ACFS file system you want to create and an existing Oracle ADVM volume from the lists. Specify the mount point field; the mount point must be an existing directory. The file system must be mounted to make it available. You can choose to enable the **Auto Mount** option. You can also provide a description of the file system in the **Description** field.

The mount command must be manually run as root at an operating system prompt.

You can choose to create an Oracle ACFS file system for an Oracle Database.

See Also:

- mount for information about mounting an Oracle ACFS file system on Linux
- Creating an Oracle ACFS File System for Database Use for more information about creating a file system for a database home
- Introducing Oracle ACFS and Oracle ADVM for additional information about Oracle ACFS
- Basic Steps to Manage Oracle ACFS Systems for a summary of the basic steps for creating an Oracle ACFS file system

Managing Security and Encryption for Oracle ACFS with ASMCA

Oracle ASM Configuration Assistant enables you to configure security and encryption for an Oracle ACFS file system.

You must initialize the security system as the first step in configuring security for an Oracle ACFS file system. You must also initialize the encryption system as the first step in encrypting an Oracle ACFS file system. You do not have to use both Oracle ACFS security and encryption on the same file system. If you decide to use both security and encryption, then encryption must be initialized and set before enabling encryption on a security realm.

In the dialog that displays, enter the information to specify the security administrator and the operating system group of the security administrator. You can choose to create a password protected wallet. After you complete the entry fields in the dialog, click **Show Command** to display the commands you must run as a root or Administrator user at an operating system prompt. For example:

```
# /sbin/acfsutil sec init -u grid -g asmadmin
# /sbin/acfsutil encr init
```

After security has been initialized, you can use the menu options to manage security and encryption for an Oracle ACFS file system.

The menu includes options to:

- Set encryption
- Enable and disable encryption
- Prepare and enable security
- Enable and disable security

See Also:

-
- acfsutil encr init for information about the `acfsutil encr init` command
- Oracle ACFS Encryption for information about Oracle ACFS encryption
- Encrypting Oracle ACFS File Systems using OCR as Encryption Key Store for a summary of the basic steps for setting up encryption on an Oracle ACFS file system
- Oracle ACFS Command-Line Utilities for information about the `acfsutil encr` command

Creating an Oracle ACFS File System for Database Use

To create an Oracle ACFS file system for database use, you can select the **Create ACFS for Database Use** from the disk group configuration options menu to display the dialog entry box.

Note:

When creating an Oracle ADVM volume for an Oracle ACFS file system that is intended to store database files, use the ASMCMD `volcreate` command or the SQL `ALTER DISKGROUP ADD VOLUME` SQL statement rather than the ASMCA tool to ensure that the column striping is set to 1.

In this dialog, you must enter:

- Volume Name

  This is the name of the Oracle ADVM volume you want to create.
- Mount Point

  This is the mount point for the file system where you want to install the database home. The file system that contains the database home should not be located under the Oracle Grid Infrastructure base (`ORACLE_BASE` for grid) directory.
- Size in gigabytes (GB)

  The default is 7 GB and the minimum recommended size.
- Owner Name

  This is the operating system name of the user that installs the database and owns the software in the database home.
- Owner Group

  This is the operating system group of the owner of the database home.

The mount point must be an existing directory. The file system must be mounted to make it available.

Select **Automatically run configuration commands** to run ASMCA configuration commands automatically. To use this option, you must provide the `root` credentials on the ASMCA Settings page.

The mount command may also be run manually as `root` at an operating system prompt.

> **See Also:**
>
> - volcreate for information about the `volcreate` command
> - Overview of Oracle ASM Dynamic Volume Manager for information about Oracle ADVM volumes
> - About Oracle ACFS and Oracle Database Homes for information about mount points and database homes
> - mount for information about mounting an Oracle ACFS file system on Linux
> - About the Oracle ACFS Mount Registry for information on the mount registry
> - *Oracle Automatic Storage Management Administrator's Guide* for information about the `ALTER DISKGROUP ADD VOLUME` SQL statement to administer volumes

ASMCA Command-Line Interface for Managing Oracle ACFS and Oracle ADVM

The ASMCA command-line interface provides non-GUI support for configuring Oracle ASM disk groups, volumes, and Oracle ACFS.

- ASMCA Commands for Oracle ACFS and Oracle ADVM

> **See Also:**
>
> - *Oracle Automatic Storage Management Administrator's Guide* for general information about running the ASMCA command-line interface and about options that are common to multiple commands

ASMCA Commands for Oracle ACFS and Oracle ADVM

This section describes the commands that can be run with ASMCA command-line to manage Oracle ACFS objects.

- Create an Oracle ACFS Snapshot
- Delete an Oracle ACFS Snapshot

- Create a Volume
- Create an Oracle ACFS File System

Create an Oracle ACFS Snapshot

`–createACFSSnapshot` creates an Oracle Advanced Cluster File System (Oracle ACFS) snapshot.

Syntax

```
asmca -silent
   -createACFSSnapshot
       (-acfsMountPoint mount_path )
       (-snapshotName snapshot_name )
       [-snapshotMode { r|w } ]
       [-parentSnapshotName parent_snapshot_name ]
```

Table 4-1 contains the options available with the `–createACFSSnapshot` command.

Table 4-1 Options for the -createACFSSnapshot command

Option	Description
`-acfsMountPoint` *mount_path*	Specifies an existing path to be used as the mount point. This is required in an Oracle RAC environment.
`-snapshotName` *anapshot_name*	Specifies the name of the Oracle ACFS snapshot.
`-snapshotMode r|w`	Specifies the mode of the Oracle ACFS snapshot, either read-only (`r`) or read-write (`w`).
`-parentSnapshotName` *parent_snapshot_name*	Specifies the name of the Oracle ACFS parent snapshot.

Delete an Oracle ACFS Snapshot

`–deleteACFSSnapshot` deletes an existing Oracle Advanced Cluster File System (Oracle ACFS) snapshot.

Syntax

```
asmca -silent
   -deleteACFSSnapshot
       (-acfsMountPoint mount_path )
       (-snapshotName snapshot_name )
```

Table 4-2 contains the options available with the `–deleteACFSSnapshot` command.

Table 4-2 Options for the -deleteACFSSnapshot command

Option	Description
`-acfsMountPoint` *mount_path*	Specifies an existing path to be used as the mount point. This is required in an Oracle RAC environment.
`-snapshotName` *anapshot_name*	Specifies the name of the Oracle ACFS snapshot.

Create a Volume

`-createVolume` creates Oracle ADVM volumes.

Syntax

```
asmca -silent
      -createVolume
        { -volumeName volume_name
         -volumeDiskGroup diskgroup
         -volumeSizeGB size_GB
         [ -volumeRedundancy { INHERIT | HIGH | NORMAL | EXTERNAL } ] ... }
       [-sysAsmPassword sysasm_password ]
```

Table 4-3 contains the options available with the `-createVolume` command.

Table 4-3 Options for the -createVolume command

Option	Description
`-volumeName` *volume_name*	Specifies the name of the volume to create.
`-volumeDiskGroup` *diskgroup*	Specifies the name of the disk group where you want to create the volume.
`-volumeSizeGB` *size_GB*	Specifies the size of the volume in Gigabytes.
`-volumeRedundancy` { `INHERIT` | `HIGH` | `NORMAL` | `EXTERNAL` }	Specifies the redundancy setting for the volume.

Examples

To create an Oracle ADVM volume:

Example 4-1 Using asmca -silent -createVolume

```
$ asmca -silent
        -createVolume
           -volumeName volume1
           -volumeDiskGroup mynewdg
           -volumeSizeGB 1

Volume volume1 created successfully.
```

Create an Oracle ACFS File System

`-createACFS` creates an Oracle Advanced Cluster File System (Oracle ACFS).

This command does not mount the Oracle ACFS file system. For information about mounting an Oracle ACFS file system, refer to mount for Linux environments.

Syntax

```
asmca -silent
      -createACFS
       { -acfsVolumeDevice volume_device }
       { -acfsMountPoint mount_path }
       [-acfsUserName acfs_username ]
```

```
       [-acfsUserGroup acfs_usergroup ]
       [-sysAsmPassword sysasm_password ]
```

Table 4-4 contains the options available with the -`createACFS` command.

Table 4-4 Options for the -createACFS command

Option	Description
`-acfsVolumeDevice` *`volume_device`*	Specifies the name of the Oracle ADVM volume device.
`-acfsMountPoint` *`mount_path`*	Specifies an existing path to be used as the mount point. This is required in an Oracle RAC environment.
`-acfsUserName` *`acfs_username`*	Specifies the Oracle ACFS user name.
`-acfsUserGroup` *`acfs_usergroup`*	Specifies the Oracle ACFS group name.

Examples

To create an Oracle ACFS file system:

Example 4-2 Using asmca -silent -createACFS

```
$ asmca -silent
        -createACFS
           -acfsVolumeDevice /dev/asm/volume1-457
           -acfsMountPoint /acfsmounts/acfs1
```

5

Managing Oracle ADVM with ASMCMD

Oracle ASM Command-Line Utility (ASMCMD) provides commands for managing Oracle ADVM volumes.

You can run the ASMCMD utility in either interactive or noninteractive mode.

This chapter describes the Oracle Automatic Storage Management (Oracle ASM) Command-Line Utility (ASMCMD) volume management commands.

- ASMCMD Volume Management Commands

See Also:

- Overview of Oracle ASM Dynamic Volume Manager for information about Oracle ASM Dynamic Volume Manager (Oracle ADVM)
- *Oracle Automatic Storage Management Administrator's Guide* for information about the `ALTER DISKGROUP ADD VOLUME` SQL statement to administer volumes
- *Oracle Automatic Storage Management Administrator's Guide* for general information about running ASMCA command-line interface

ASMCMD Volume Management Commands

This topic provides a summary of the Oracle ADVM volume management commands.

Table 5-1 lists the Oracle ADVM volume management commands with brief descriptions. To successfully run these commands, the local Oracle ASM instance must be running and the disk group required by this command must have been created and mounted in the Oracle ASM instance.

Table 5-1 Summary of ASMCMD volume management commands

Command	Description
volcreate	Creates an Oracle ADVM volume in the specified disk group.
voldelete	Deletes an Oracle ADVM volume.
voldisable	Disables Oracle ADVM volumes in mounted disk groups.
volenable	Enables Oracle ADVM volumes in mounted disk groups.
volinfo	Displays information about Oracle ADVM volumes.
volresize	Resizes an Oracle ADVM volume.
volset	Sets attributes of an Oracle ADVM volume in mounted disk groups.
volstat	Reports volume I/O statistics.

volcreate

Purpose

Creates an Oracle ADVM volume in the specified disk group.

Syntax and Description

```
volcreate -G diskgroup -s size
   [ --column number ] [ --width stripe_width ]
   [--redundancy {high|mirror|unprotected}]
   volume
```

Table 5-2 describes the options for the `volcreate` command.

Table 5-2 Options for the volcreate command

Option	Description
`-G` *diskgroup*	Name of the disk group containing the volume.
`-s` *size*	Size of the volume to be created in units of `K`, `M`, `G`, or `T`. The value must be a positive integer. The unit designation must be appended to the number specified. A space is not allowed between the number and the unit designation. For example: `200M` or `20G`
`--column` *number*	Number of columns in a stripe set. Values range from `1` to `8`. The default value is `8`.
`--width` *stripe*	Stripe width of a volume in units of `K` or `M`. The value can range from 4 KB to 1 MB, at power-of-two intervals. The default is `1M`.
`--redundancy {high|mirror|unprotected}`	Redundancy of the Oracle ADVM volume which can be only specified for normal redundancy disk groups. If redundancy is not specified, the setting defaults to the redundancy level of the disk group and this is the recommended setting. The range of values is as follows: `unprotected` for non-mirrored redundancy, `mirror` for double-mirrored redundancy, or `high` for triple-mirrored redundancy.
volume	Name of the volume. Only alphanumeric characters and underscores are allowed. Hyphens are not allowed. The first character must be alphabetic.

> **WARNING:**
>
> Specifying `--redundancy unprotected` means that Oracle ASM mirroring is not available for data recovery with the Oracle ADVM volume. The redundancy setting (`normal`) of the disk group does not provide mirroring for an unprotected Oracle ADVM volume. The `unprotected` configuration is not recommended for production environments as intermittent storage access failures can result in the loss of data. Backups are strongly recommended.

When creating an Oracle ADVM volume, a volume device name is created with a unique Oracle ADVM persistent disk group number that is concatenated to the end of the volume name. The unique number can be one to three digits.

On Linux, the volume device name is in the format *`volume_name-nnn`*, such as `volume1-123`.

On Linux platforms, the volume name must be less than or equal to eleven alphanumeric characters, starting with an alphabetic character. On AIX platforms, the volume name must be less than or equal to twenty three alphanumeric characters, starting with an alphabetic character. On Solaris platforms, the volume name must be less than or equal to thirty alphanumeric characters, starting with an alphabetic character.

You can determine the volume device name with the `volinfo` command.

If the `--column` option is set to `1`, then striping is disabled and the stripe width equals the default volume extent size (64 MB). Setting the `--column` option to 8 (the default) is recommended to achieve optimal performance with database data files and other files.

A successful volume creation automatically enables the volume device.

The volume device file functions as any other disk or logical volume to mount file systems or for applications to use directly.

When creating an accelerator volume, create the volume on a disk group with storage that is significantly faster than the primary volume's storage.

Before creating an Oracle ADVM volume on AIX, ensure that the necessary user authorizations have been created.

Examples

The following is an example of the `volcreate` command that creates `volume1` in the `data` disk group with the size set to 10 gigabytes.

Example 5-1 Using the ASMCMD volcreate command

```
ASMCMD [+] >  volcreate -G data -s 10G --width 1M --column 8 volume1

ASMCMD [+] > volinfo -G data volume1
Diskgroup Name: DATA

         Volume Name: VOLUME1
         Volume Device: /dev/asm/volume1-123
         State: ENABLED
         Size (MB): 10240
         Resize Unit (MB): 64
         Redundancy: MIRROR
         Stripe Columns: 8
         Stripe Width (K): 1024
         Usage:
         Mountpath:
```

See Also:

- Creating an Oracle ACFS File System for information on mounting the volume device file
- volinfo for information about the `volinfo` command.
- Limits of Oracle ADVM for information about Oracle ADVM limits
- mkfs for information about the accelerator volume
- Oracle ACFS Command-Line Tools for the AIX Environment for information about user authorizations on AIX
- *Oracle Automatic Storage Management Administrator's Guide* for information about mirroring, redundancy, and failure groups

voldelete

Purpose

Deletes an Oracle ADVM volume.

Syntax and Description

```
voldelete -G diskgroup volume
```

Table 5-3 describes the options for the `voldelete` command.

Table 5-3 Options for the voldelete command

Option	Description
`-G` *`diskgroup`*	Name of the disk group containing the volume.
`volume`	Name of the volume.

To successfully run this command, the local Oracle ASM instance must be running and the disk group required by this command must be mounted in the Oracle ASM instance. Before deleting a volume, you must ensure that there are no active file systems associated with the volume.

Examples

The following is an example of the `voldelete` command that deletes `volume1` from the `data` disk group.

Example 5-2 Using the ASMCMD voldelete command

```
ASMCMD [+] > voldelete -G data volume1
```

voldisable

Purpose

Disables Oracle ADVM volumes in mounted disk groups and removes the volume device on the local node.

Syntax and Description

```
voldisable { --all |{ -G diskgroup { -a | volume } } }
```

Table 5-4 describes the options for the `voldisable` command.

Table 5-4 Options for the voldisable command

Option	Description
`--all`	Specifies all volumes within all disk groups.
`-G diskgroup`	Name of the disk group containing the volume.
`-a`	Specifies all volumes within the specified disk group.
`volume`	Name of the volume.

You can disable volumes before shutting down an Oracle ASM instance or dismounting a disk group to verify that the operations can be accomplished normally without including a force option due to open volume files. Disabling a volume also prevents any subsequent opens on the volume or device file because it no longer exists.

Before disabling a volume, you must ensure that there are no active file systems associated with the volume. You must first dismount the Oracle ACFS file system before disabling the volume. Refer to Deregistering, Dismounting, and Disabling Volumes and Oracle ACFS File Systems.

You can delete a volume without first disabling the volume.

Examples

The following is an example of the `voldisable` command that disables `volume1` in the `data` disk group.

Example 5-3 Using the ASMCMD voldisable command

```
ASMCMD [+] > voldisable -G data volume1
```

volenable

Purpose

Enables Oracle ADVM volumes in mounted disk groups.

Syntax and Description

```
volenable {  --all |{ -G diskgroup { -a | volume } } }
```

Table 5-5 describes the options for the `volenable` command.

Table 5-5 Options for the volenable command

Option	Description
`--all`	Specifies all volumes within all disk groups.
`-G` *`diskgroup`*	Name of the disk group containing the volume.
`-a`	Specifies all volumes within the specified disk group.
`volume`	Name of the volume.

A volume is enabled when it is created.

Examples

The following is an example of the `volenable` command that enables `volume1` in the `data` disk group.

Example 5-4 Using the ASMCMD volenable command

```
ASMCMD [+] > volenable -G data volume1
```

volinfo

Purpose

Displays information about Oracle ADVM volumes.

Syntax and Description

```
volinfo {  --all |{ -G diskgroup { -a | volume } } }
volinfo {--show_diskgroup |--show_volume}  volumedevice
```

Table 5-6 describes the options for the `volinfo` command.

Table 5-6 Options for the volinfo command

Option	Description
`--all`	Specifies all volumes within all disk groups.
`-G` *`diskgroup`*	Name of the disk group containing the volume.
`-a`	Specifies all volumes within the specified disk group.
`volume`	Name of the volume.
`--show_diskgroup`	Returns only the disk group name. A volume device name is required.
`--show_volume`	Returns only the volume name. A volume device name is required.
`volumedevice`	Name of the volume device.

Examples

The first example displays information about the `volume1` volume in the `data` disk group. This example was produced in a Linux environment.

The Mountpath field contains the path where the volume is currently mounted or where the volume was last mounted.

Example 5-5 Using the ASMCMD volinfo command

```
ASMCMD [+] > volinfo -G data volume1
Diskgroup Name: DATA
         Volume Name: VOLUME1
         Volume Device: /dev/asm/volume1-123
         State: ENABLED
         Size (MB): 10240
         Resize Unit (MB): 64
         Redundancy: MIRROR
         Stripe Columns: 8
         Stripe Width (K): 1024
         Usage: ACFS
         Mountpath: /acfsmounts/acfs1
```

volresize

Purpose

Resizes an Oracle ADVM volume.

Syntax and Description

```
volresize -G diskgroup -s size [ -f ] volume
```

Table 5-7 describes the options for the volresize command.

Table 5-7 Options for the volresize command

Option	Description
-G *diskgroup*	Name of the disk group containing the volume.
-f	Forces the shrinking of a volume that is not an Oracle ACFS volume and suppresses any warning message.
volume	Name of the volume.
-s *size*	New size of the volume in units of K, M, G, or T.

If the volume is mounted on a non-Oracle ACFS file system, then dismount the file system first before resizing. If the new size is smaller than current, you are warned of possible data corruption. Unless the -f (force) option is specified, you are prompted whether to continue with the operation.

If there is an Oracle ACFS file system on the volume, then you cannot resize the volume with the volresize command. You must use the acfsutil size command, which also resizes the volume and file system. For information, see "acfsutil size".

Examples

The following is an example of the volresize command that resizes volume1 in the data disk group to 20 gigabytes.

Example 5-6 Using the ASMCMD volresize command

```
ASMCMD [+] > volresize -G data -s 20G volume1
```

volset

Purpose

Sets attributes of an Oracle ADVM volume in mounted disk groups.

Syntax and Description

```
volset -G diskgroup [ --usagestring string]
        [--mountpath  mount_path ]
        volume
```

Table 5-8 describes the options for the `volset` command.

Table 5-8 Options for the volset command

Option	Description
`-G` *diskgroup*	Name of the disk group containing the volume.
`--usagestring` *string*	Optional usage string to tag a volume which can be up to 30 characters. This string is set to `ACFS` when the volume is attached to an Oracle ACFS file system and should not be changed.
`--mountpath` *mount_path*	Optional string to tag a volume with its mount path string which can be up to 1024 characters. This string is set when the file system is mounted and should not be changed.
volume	Name of the volume.

When running the `mkfs` command to create a file system, the usage field is set to `ACFS` and `mountpath` field is reset to an empty string if it has been set. The `usage` field should remain at `ACFS`.

When running the `mount` command to mount a file system, the `mountpath` field is set to the mount path value to identify the mount point for the file system. After the value is set by the `mount` command, the `mountpath` field should not be updated.

Examples

The following is an example of a `volset` command that sets the `usage` string for a volume that is not associated with a file system.

Example 5-7 Using the ASMCMD volset command

```
ASMCMD [+] > volset -G data --usagestring 'no file system created' volume1
```

volstat

Purpose

Reports I/O statistics for Oracle ADVM volumes.

Syntax and Description

```
volstat [-G diskgroup] [volume]
```

Table 5-9 describes the options for the `volstat` command.

Table 5-9 Options for the volstat command

Option	Description
`-G` *`diskgroup`*	Name of the mounted disk group containing the volume.
`volume`	Name of the volume.

The following apply when using the `volstat` command.

- If the disk group is not specified and the volume name is specified, all mounted disk groups are searched for the specified volume name.
- If the disk group name is specified and the volume name is omitted, all volumes are displayed for the named disk group.
- If both the disk group name and the volume name are omitted, all volumes on all disk groups are displayed.

Examples

The following is an example of the `volstat` command that displays information about volumes in the `data` disk group.

Example 5-8 Using the ASMCMD volstat command

```
ASMCMD [+] > volstat -G data
DISKGROUP NUMBER / NAME:  1 / DATA
---------------------------------------
  VOLUME_NAME
     READS            BYTES_READ       READ_TIME        READ_ERRS
     WRITES           BYTES_WRITTEN    WRITE_TIME       WRITE_ERRS
  -------------------------------------------------------------
  VOLUME1
     10085            2290573312       22923            0
     1382             5309440          1482             0
```

6 Managing Oracle ACFS with Command-Line Tools

Numerous command-line tools are provided for managing Oracle Advanced Cluster File System (Oracle ACFS) and Oracle ASM Dynamic Volume Manager (Oracle ADVM).

Command-line tools for managing and implementing Oracle Advanced Cluster File System (Oracle ACFS), include the following:

- Basic Steps to Manage Oracle ACFS Systems
- Oracle ACFS Command-Line Tools for Linux Environments
- Oracle ACFS Command-Line Tools for the Solaris Environment
- Oracle ACFS Command-Line Tools for the AIX Environment
- Oracle ACFS Command-Line Tools for Tagging
- Oracle ACFS Command-Line Tools for Replication
- Oracle ACFS Command-Line Tools for Encryption
- Oracle ACFS Command-Line Tools for Snapshots
- Oracle ACFS Command-Line Tools for Compression
- Oracle ACFS Command-Line Utilities
- Oracle ACFS Diagnostic Commands

> **Note:**
>
> Oracle does not recommend using identifiers for Oracle Database object names that must be quoted. While the use of quoted identifiers may be valid as names in some command-line tools or in SQL statements, the names may not be valid when using other tools that manage the object.

> **See Also:**
>
> - About Using Oracle ACFS Command-Line Tools for information about running Oracle ACFS `acfsutil` commands
> - *Oracle Database SQL Language Reference* for more information about naming Oracle Database objects

Oracle ACFS Diagnostic Commands

This topic provides a summary of the Oracle ACFS command-line utilities for diagnostic purposes.

Oracle ACFS provides various `acfsutil` command-line utilities for diagnostic purposes.

> **Note:**
>
> Run the diagnostic commands only when Oracle Support requests diagnostic data for analysis.

The following table lists the Oracle ACFS utilities with brief descriptions.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

Table 6-1 Summary of Oracle ACFS diagnostic commands

Command	Description
acfsdbg	Debugs an Oracle ACFS file system.
acfsutil blog	Writes text to a blog file.
acfsutil dumpstate	Collects internal Oracle ACFS state information.
acfsutil info ftrace	Display the trace entries for open files.
acfsutil lockstats	Displays lock contention statistics.
acfsutil log	Retrieves memory diagnostic log files and manages debug settings.
acfsutil meta	Copies metadata from an Oracle ACFS file system into a separate output file.
acfsutil plogconfig	Manages Oracle ACFS persistent logging configuration settings.
acfsutil tune	Modifies or displays Oracle ACFS tunable parameters.
advmutil tune	Modifies or displays Oracle ADVM parameters.

acfsdbg

Purpose

Debugs an Oracle ACFS file system.

Syntax and Description

```
acfsdbg [-r] [-l] [-x] volume_device
acfsdbg -h
```

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

The following table contains the options available with the `acfsdbg` command.

Table 6-2 Options for the acfsdbg command

Option	Description
`-h`	Prints out the usage message which displays the various options that are available when invoking the `acfsdbg` command, then exits.
`-r`	Operates in read-only mode. No data is modified on the file system and all write commands are disabled. If the device is mounted anywhere, `acfsdbg` may not display the latest data because some data is cached by the file system mounts.
`-l`	Processes kernel log files. The default is to not process the log files.
`—x` *`file_name`*	Specified for accelerator data collected by `acfsutil meta`. Only used for this type of data.
`volume_device`	Specifies the device name of the volume.

You must be the administrator or a member of the Oracle ASM administrator group to run `acfsdbg`.

Subcommands

Table 6-3 lists the subcommands of `acfsdbg`.

Table 6-3 Subcommands for acfsdbg

Option	Description	Syntax
`calculate`	Calculates simple arithmetic expressions Valid operators: + - * / % & | ^ ~ << >> White space starts a new expression 0-1 represents a negative 1	`calculate` [`-v`] `expr` [...] `-v` Verbose mode `expr` Simple 2+2 expression
`cksum`	Generates and replaces checksum in header Header offset can be an expression as used by the `calculate` subcommand White space starts a new header offset Command is disabled in read-only mode	`cksum` [`-C` | `-CE`] *`header_offset`* [...] `-C` Regenerate for normal structure checksum `-CE` Re-generate for Extent structure checksum *`header_offset`* Offset of the on disk structure header. The value can be an expression as used by the `calculate` subcommand
`close`	Closes the open handle to the device	`close`
`echo`	Echoes text on command line to stdout	`echo`

Table 6-3 (Cont.) Subcommands for acfsdbg

Option	Description	Syntax
`fenum`	Displays the specified File Entry TAble (FETA) entry	`fenum` [`-f` | `-e` | `-d`] *`FETA_entry_number`* `-f` Displays all on disk structures related to this structure `-e` Displays all on disk extent information related to this structure `-d` Casts the structure as a directory and displays its contents *`FETA_entry_number`* The File Entry Table number used to identify a file on the file system
`help`	Displays help message	`help`
`offset`	Displays structure at disk offset	`offset` [`-c cast`] [`-f` | `-d`] *`disk_offset`* `-f` Displays all on disk structures related to this structure `-d` Casts the structure as a directory and displays its contents *`disk_offset`* Disk offset to display. The value can be an expression as used by the `calculate` subcommand
`open`	Opens a handle to a device. The default is the volume device name entered on the command line	`open` [*`volume_device`*]
`primary`	Sets the context of commands to the primary file system	`primary`
`prompt`	Sets the prompt to the specified string	`prompt` "*`prompt_string`*"
`quit`	Exits the `acfsdbg` debugger command	`quit`
`read`	Reads value from offset The default size to read in is 8 bytes The default count to read is 1	`read` [`-1` | `-2` | `-4` | `-8` | `-s`] [*`count`*] *`offset`* `-1` Read byte value `-2` Read 2 byte (short) value `-4` Read 4 byte (int) value `-8` Read 8 byte (long) value `-s` Read null- terminated string *`count`* Number of values to read. If not specified, the default is 1 *`offset`* Disk offset to read. The value can be an expression as used by the `calculate` subcommand
`snapshot`	Sets the context of commands to the specified snapshot	`snapshot` *`snapshot_name`*

Table 6-3 (Cont.) Subcommands for acfsdbg

Option	Description	Syntax
`write`	Writes hexadecimal, octal, or decimal values at the disk offset, estimating how many bytes to write based on value size or number of digits in leading 0 hexadecimal values The disk offset can be an expression used by the `calculate` subcommand Numeric values can also be an expression as used by the `calculate` subcommand This command is disabled in read-only mode	`write` [`-1` | `-2` | `-4` | `-8` | `-c` | `-s`] [`-C` | `-CE`] *`offset value`* `-1` Write byte value `-2` Write 2 byte (short) value `-4` Write 4 byte (int) value `-8` Write 8 byte (long) value `-c` Write text (no null termination). Enclose string in single-quotes (') `-s` Write null-terminated string. Enclose string in quotes (") `-C` Regenerate normal structure checksum `-CE` Regenerate extent structure checksum *`offset`* Disk offset to write. The value can be an expression used by the `calculate` subcommand *`value`* The value to write. If numeric, the value can be an expression as used by the `calculate` subcommand

Examples

Example 6-1 shows the use of the `acfsdbg` subcommand.

Example 6-1 Using the acfsdbg command

```
$ /sbin/acfsdbg /dev/asm/voume1-123
acfsdbg: version                    = 11.2.0.3.0
Oracle ASM Cluster File System (ACFS) On-Disk Structure Version: 39.0
The ACFS volume was created at  Mon Mar  2 14:57:45 2011
acfsdbg>

acfsbdg> calculate 60*1024
    61,440
    61440
    61440
    0xf000
    0170000
    1111:0000:0000:0000

acfsdbg> prompt "acfsdbg test>"
acfsdbg test>

echo "offset 64*1024" | acfsdbg /dev/asm/volume1-123
```

acfsutil blog

Purpose

Writes text to the blog file.

Syntax and Description

```
acfsutil [-h] blog

acfsutil blog {-t text | -u} mount_point
```

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

The following table contains the options available with the `acfsutil blog` command.

Table 6-4 Options for the acfsutil blog command

Option	Description
`-t` *`text`*	Writes text to the blog file at the specified mount point.
`-u`	Updates blog debug levels from `dbg` file.
`mount_point`	Specifies the mount point.

The `acfsutil blog` command enables you to write text to a blog file.

.

Examples

The following example illustrates how to run the `acfsutil blog` command. Running `acfsutil blog` with the `–h` option displays help.

Example 6-2 Using acfsutil blog

```
$ /sbin/acfsutil -h

$ /sbin/acfsutil -t "this is a blog test" blog my_mount_point

$ /sbin/acfsutil -u blog my_mount_point
```

acfsutil dumpstate

Purpose

Collects internal Oracle ACFS state information for diagnosis by Oracle support.

Syntax and Description

```
acfsutil [-h] dumpstate
acfsutil dumpstate {acfs_path | [-d [-p {file_name | -}]] [-z] [acfs_path]}
```

`acfsutil -h dumpstate` displays help text and exits.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

The following table contains the options available with the `acfsutil dumpstate` command.

Table 6-5 Options for the acfsutil dumpstate command

Option	Description
acfs_path	Specifies the directory path to an Oracle ACFS file system.
`-d`	Dump statistics to `acfs.dumpstats` in the current directory or the output specified by the `-p` option.
`-p { file_name | - }`	Modifies the `-d` option to specify a file name instead of `acfs.dumpstats`. You can also specify `-` in place of a file name to display the output instead of writing to a file.
`-z`	Clears all current statistics.

The `acfsutil dumpstate` command collects internal Oracle ACFS state information for a specified file system. The state information is written to a binary incident file in a logging directory. The binary log incident file is specific to the file system mounted at the specified path. The `acfs.dumpstats` statistics file, or the specified alternate output, contains statistics for the entire Oracle ACFS kernel module.

> **Note:**
>
> Run the `acfsutil dumpstate` command only when Oracle Support requests diagnostic and debugging data for analysis.

Examples

The following example shows the use of the `acfsutil dumpstate` command.

Example 6-3 Using the acfsutil dumpstate command

The following command execution creates a binary incident file for the specified file system.

```
$ /sbin/acfsutil dumpstate /acfsmounts/acfs1/
```

The following command execution dumps file system statistics to `acfs.dumpstats` and creates a binary incident file for the specified file system.

```
$ /sbin/acfsutil dumpstate -d /acfsmounts/acfs1/
```

The following command execution clears statistics for all file systems.

```
$ /sbin/acfsutil dumpstate -z
```

The following command execution dumps file system statistics to `acfs.dumpstats`, creates a binary incident file, and clears all file system statistics for the specified file system.

```
$ /sbin/acfsutil dumpstate -d -z /acfsmounts/acfs1/
```

acfsutil info ftrace

Purpose

Display the trace entries for open files associated with the Oracle ACFS file system specified by the mount point.

Syntax and Description

```
acfsutil info ftrace -h
acfsutil info ftrace [-s] mount_point
```

`acfsutil info ftrace —h` displays help text and exits.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

The following table contains the options available with the `acfsutil info ftrace` command.

Table 6-6 Options for the acfsutil info ftrace command

Option	Description
`—s`	Display only file IDs of open files.
`mount_point`	Specifies the directory where the file system is mounted.

The `acfsutil info ftrace` command displays a list of open files on a mounted Oracle ACFS file system.

The Oracle ACFS kernel driver keeps track of which files are loaded in memory. These files may not have an active open and may just be cached. The open file tracing is able to determine which of those cached File Control Blocks (FCBs) have an active open or reference. The intent of this command is to enable you to determine if, and which, files are still being referenced that could prevent a file system unmount from occuring.

When `acfsutil info ftrace` initially runs, the command attempts to purge any cached files that are no longer referenced. This operation may require some time to complete because the modified metadata and user data for each file has to be flushed to disk.

The following describes the output of the `acfsutil info ftrace` command. Note that a file can refer to a regular file or directory.

The basic format of the output is:

```
Fileid: %ID%, Pathname: %PATH%
          [%OP%] Pid:  %PID% Ppid:   %PPID% Elapsed time: %TIME% Cmd: %CMD%
          ...
```

The fields are described in the following list.

- `%ID%`: The numeric file identifier. This is the same number that is used with `acfsutil info id`. This value is also known as the inode number on Linux.

- `%PATH%`: The generated pathname for the file based on the `%ID%`. `N/A` may be displayed if it is not available.
- `%OP%`: The type of operation that accessed the file. The values may be the following:
  - `LOOKUP`: The specified process looked up this file via the pathname.
  - `CREATE`: The specified process created the file.
  - `NFS`: A NFS process has accessed the file on behalf of client.
  - `OPEN`: The specified process opened the file.
  - `MAP`: The specified process mapped the file into memory.
- `%PID%`: The process id `%PPID%`: The parent process id. This output item may not be available.
- `%TIME%`: The elapsed time from when the operation occurred. The format is: d (days), h (hours), m (minutes), s (seconds)
- `%CMD%`: The name of the process that performed the operation.

Each file listed may have more than one operation listed depending on the system workload. The amount of operations displayed is limited to conserve memory. The Oracle ACFS driver keeps a rotating log for each operation and the operation entries may wrap. As a result, the oldest operation may not be the first one displayed.

Examples

The following example shows the `acfsutil info ftrace` command run on the `/mnt` mount point.

Example 6-4 Using the acfsutil info ftrace command

```
$ acfsutil info ftrace /mnt
Fileid: 42, Pathname: /mnt/yum.conf
        [LOOKUP] Pid:  27009 Ppid:  14999 Elapsed time: 0d 00h 00m 03s Cmd: tail
        [OPEN  ] Pid:  27009 Ppid:  14999 Elapsed time: 0d 00h 00m 03s Cmd: tail

Fileid: 155, Pathname: /mnt/bash
        [LOOKUP] Pid:   9731 Ppid:  19588 Elapsed time: 0d 00h 00m 08s Cmd: cp
        [OPEN  ] Pid:   9731 Ppid:  19588 Elapsed time: 0d 00h 00m 08s Cmd: cp
        [OPEN  ] Pid:   9735 Ppid:  19588 Elapsed time: 0d 00h 00m 05s Cmd: bash
        [MAP   ] Pid:   9735 Ppid:  19588 Elapsed time: 0d 00h 00m 05s Cmd: bash
        [MAP   ] Pid:   9735 Ppid:  19588 Elapsed time: 0d 00h 00m 05s Cmd: bash
        [MAP   ] Pid:   9735 Ppid:  19588 Elapsed time: 0d 00h 00m 05s Cmd: bash

Fileid: 43, Pathname: /mnt/dir1
        [LOOKUP] Pid:  14485 Ppid:   7829 Elapsed time: 0d 12h 20m 13s Cmd: mkdir
        [LOOKUP] Pid:   7829 Ppid:   7828 Elapsed time: 0d 12h 20m 06s Cmd: bash
        [LOOKUP] Pid:   7829 Ppid:   7828 Elapsed time: 0d 12h 20m 06s Cmd: bash
```

acfsutil lockstats

Purpose

Displays lock contention statistics.

Syntax and Description

```
acfsutil lockstats [-h]
```

```
       Command Subcmd              Arguments
-------------- ------------------ --------------------------------------------
     lockstats lh                 [-b] [-e] [-z] [-t <top>] [-s <sort column>]
     lockstats lh                 -b                  - Begin collecting lock 
statistics
     lockstats lh                 -e                  - End   collecting lock 
statistics
     lockstats lh                 -z                  - Zero out the lock 
statistics collected so far
     lockstats lh                 -t <n>              - Print only the top 'n' 
lock statistics
     lockstats lh                 -s < acquires       - Sort the data on the 
specified column
     lockstats lh                     |totalwait
     lockstats lh                     |maxwait
     lockstats lh                     |totalheld
     lockstats lh                     |maxheld
```

`acfsutil lockstats lh —h` displays help text and exits.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

The following table contains the options available with the `acfsutil lockstats` command.

Table 6-7 Options for the acfsutil lockstats command

Option	Description
`—b`	Begins (enables) collecting lock statistics.
`—e`	Stops (disables) collecting lock statistics.
`—z`	Zeroes out (clears) the current collected lock statistics.
`—t` *`top_n`*	Displays the top *n* lock statistics.
`—s` *`sort_column`*	Sorts the lock statistics on the specified sort column. Valid sort column values are: `acquires`, `totalwait`, `maxwait`, `totalheld`, and `maxheld`. By default, the statistics are sorted by `Total Wait`.

The command's output is displayed in a tabular format with eight columns.

- The first column is the lock hierarchy group name.
- The second column is the number of locks acquired in that particular group.
- The third column is the maximum time waited among all the lock acquires.
- The fourth and seventh columns are file/line number references.
- The fifth column is the cumulative time waited for all the lock acquires.
- The sixth column is the maximum number of locks held per type during the specified period.

Examples

The following example shows multiple ways to use of the `acfsutil lockstats` command.

Example 6-5 Using the acfsutil lockstats command

```
# Enable lock statistics collection in the kernel. No data is displayed.
$ acfsutil lockstats lh -b

# Zero out any and all the lock statistics collected. No data is displayed.
$ acfsutil lockstats lh -z

# Disable lock statistics collection in the kernel. No data is displayed.
$ acfsutil lockstats lh -e

# Displays lock statistics.
$ acfsutil lockstats lh
 Stats time period: Mon Apr  8 14:23:41 2024 -> Mon Apr  8 14:25:50 2024 (129 sec)
+---------------------------+---- ----+----------+---------------------+------------
+----------+--------------------+------------+
|         Lock Type         | Acquires| Max Wait |    @ File/Line      | Total Wait |
Max Held |      @ File/Line     | Total Held |
+---------------------------+---------+----------+---------------------+------------
+----------+--------------------+------------+
| LH_RangeDLMLock           |  458752 |        1 | ofsgensnap.c/14091 |       4507
|        9 | ofsgensnap.c/25847 |      23972 |
| LH_FileDLMLock            |  131091 |       16 | ofslinfops.c/9812  |       1440
|      487 | ofslinfops.c/9812  |     121508 |
| LH_SnapCowBufferLock      |       9 |      283 | ofsgensnap.c/31458 |       1275
|       63 | ofsgensnap.c/31458 |        358 |
| LH_MetabufLock            | 2715309 |        1 | ofsgenmeta.c/8364  |         98
|      158 | ofsgenmeta.c/8364  |      19880 |
| LH_SnapMasksLock          |  399725 |        6 | ofsgensnap.c/27393 |         31
|        7 | ofsgensnap.c/3161  |       2643 |
| LH_IgnoreCompletely       |94075009 |        0 |            */*      |          0
|        * |            */*     |          * |
...
| LH_GlobalVCBListResource |      18 |         0 | ofslinmisc.c/2164   |           0
|        0 | ofslinmisc.c/2164 |          0 |\
+---------------------------+---------+-----------+---------------------+------------
+----------+-------------------+------------+

# Displays lock statistics, sorted on the ‘Acquires’ column.
$ acfsutil lockstats lh -s acquires
Stats time period: Mon Apr  8 14:23:41 2024 -> Mon Apr  8 14:28:31 2024 (290 sec)
+---------------------+----------+----------+---------------------+------------
+----------+---------------------+------------+
|      Lock Type      | Acquires | Max Wait |    @ File/Line      | Total Wait | Max
Held |      @ File/Line     | Total Held |
+---------------------+----------+----------+---------------------+------------
+----------+---------------------+------------+
| LH_IgnoreCompletely | 94405610 |        0 |          */*        |          0
|        * |            */*       |          * |
| LH_OFSBUFHashChain  |  2748263 |        0 |          */*        |          0
|        * |            */*       |          * |
| LH_MetabufLock      |  2734918 |        1 |ofsgenmeta.c/8364    |         98 |
158 |ofsgenmeta.c/8364    |      19941 |
| LH_DLMSpinLock      |  2512603 |        0 |          */*        |          0
|        * |            */*       |          * |
| LH_FCBSpinLock      |   963192 |        0 |          */*        |          0
```

```
|        * |           */*        |           * |
| LH_TrackOnly        |   820099 |        0 | ofsgensnapops.c/626 |          0
|        7 | ofsgensnapops.c/626 |         57 |

# Displays all of the lock statistics, sorted on the ‘Max Wait’ column.
$ acfsutil lockstats lh -s maxwait
Stats time period: Mon Apr  8 14:23:41 2024 -> Mon Apr  8 14:29:25 2024 (344 sec)
+-----------------------+----------+----------+---------------------+------------
+----------+---------------------+------------+
|      Lock Type        | Acquires | Max Wait |       @ File/Line   | Total Wait |
Max Held |       @ File/Line     | Total Held |
+-----------------------+----------+----------+---------------------+------------
+----------+---------------------+------------+
| LH_SnapCowBufferLock |        9 |      283 | ofsgensnap.c/31458 |       1275
|       63 |   ofsgensnap.c/31458 |        358 |
| LH_DRSLScan_lock      |        7 |       27 | ofsgendrsl.c/2427   |         27
|       27 | ofsgensnapops.c/5217 |         42 |
| LH_FileDLMLock        |   131092 |       16 | ofslinfops.c/9812   |       1440
|      487 |    ofslinfops.c/9812 |     121508 |
| LH_RangeMetaDataLock |   458752 |        9 | ofsgendlm.c/12260   |         26
|        9 |    ofsgendlm.c/12255 |      21869 |
| LH_SnapMasksLock      |   399738 |        6 | ofsgensnap.c/27393 |         31
|        7 |   ofsgensnap.c/-3161 |       2716 |
+-----------------------+----------+----------+---------------------+------------
+----------+---------------------+------------+

# Displays lock statistics, sorted on the ’Total Wait’ column.
$ acfsutil lockstats lh -s totalwait
Stats time period: Mon Apr  8 14:23:41 2024 -> Mon Apr  8 14:29:03 2024 (322 sec)
+----------------------+----------+----------+---------------------+------------
+------------+--------------------+------------+
|      Lock Type       | Acquires | Max Wait |     @File/Line      | Total Wait
|  Max Held  |      @ File/Line     | Total Held |
+----------------------+----------+----------+---------------------+------------
+------------+--------------------+------------+
| LH_RangeDLMLock      |   458752 |        1 |  ofsgensnap.c/14091 |       4507
|          9 | ofsgensnap.c/25847 |      23972 |
| LH_FileDLMLock       |   131092 |       16 |   ofslinfops.c/9812 |       1440
|        487 |  ofslinfops.c/9812 |     121508 |
| LH_SnapCowBufferLock|        9 |      283 |  ofsgensnap.c/31458 |       1275
|         63 | ofsgensnap.c/31458 |        358 |
| LH_MetabufLock       |  2741093 |        1 |   ofsgenmeta.c/8364 |         98
|        158 |  ofsgenmeta.c/8364 |      19962 |
| LH_SnapMasksLock     |   399737 |        6 |  ofsgensnap.c/27393 |         31
|          7 |  ofsgensnap.c/3161 |       2711 |

# Displays lock statistics, sorted on the ’Total Held’ column.
$ util lockstats lh -s totalheld
 Stats time period: Mon Apr  8 14:23:41 2024 -> Mon Apr  8 14:29:48 2024 (367
sec)
+--------------------------------------+--------------+-----------
+-----------------------------+------------+------------
+-----------------------------+------------+
|              Lock Type               |   Acquires   |  Max Wait |
@ File/Line     | Total Wait |  Max Held  |              @ File/Line     | Total
Held |
+--------------------------------------+--------------+-----------
+-----------------------------+------------+------------
+-----------------------------+------------+
| LH_FileDLMLock                       |       131092 |        16 |
ofslinfops.c/9812    |       1440 |        487 |         ofslinfops.c/9812
```

```
|      121508 |
| LH_SnapsModifyingVOPsLock           |       131076 |         1 |
ofsgenmisc.c/25924   |          4 |        490 |         ofsgenmisc.c/25924   |
121487 |
| LH_FileMetaDataLock                 |       131139 |         1 |
ofslinfops.c/9812    |          4 |        487 |         ofslinfops.c/9812    |
116919 |
| LH_SnapSyncDIOCowLock               |        65536 |         0 |
ofslinfops.c/10484   |          0 |         15 |         ofslinfops.c/10484   |
60020 |
| LH_RangeDLMLock                     |       458752 |         1 |
ofsgensnap.c/14091   |       4507 |          9 |         ofsgensnap.c/25847   |
23972 |

# Displays lock statistics, sorted on the 'Max Held' column.
$ util lockstats lh -s maxheld
 Stats time period: Mon Apr  8 14:23:41 2024 -> Mon Apr  8 14:30:02 2024 (381 sec)
+-------------------------------------+--------------+-----------
+-----------------------------+------------+------------
+-----------------------------+------------+
|              Lock Type              |   Acquires   |  Max Wait |              @ 
File/Line     | Total Wait |  Max Held  |              @ File/Line    | Total Held |
+-------------------------------------+--------------+-----------
+-----------------------------+------------+------------
+-----------------------------+------------+
| LH_SnapsModifyingVOPsLock           |       131076 |         1 |
ofsgenmisc.c/25924   |          4 |        490 |         ofsgenmisc.c/25924   |
121487 |
| LH_FileMetaDataLock                 |       131139 |         1 |
ofslinfops.c/9812    |          4 |        487 |         ofslinfops.c/9812    |
116919 |
| LH_FileDLMLock                      |       131092 |        16 |
ofslinfops.c/9812    |       1440 |        487 |         ofslinfops.c/9812    |
121508 |
| LH_ResizeDeferLock                  |          259 |         0 |
ofsgenfshare.c/13101   |          0 |        204 |       ofsgenfshare.c/13101
|        6777 |
| LH_SnapMapMarkerSerializeDLMLock    |          125 |         0 |
ofsgensnapops.c/626      |          0 |        204 |      ofsgensnapops.c/626
|        6726 |
```

acfsutil log

Purpose

Retrieves memory diagnostic log files and manages debug settings.

Syntax and Description

```
acfsutil [-h] log

acfsutil log [-f filename] [-s] [-r n{K|M|G|T|P}] [-p {avd|ofs|oks}] [-l debuglevel]
         [-n consolelevel] [-o wait_time] [-q] [-c debugcontext] [-T file_type]
         [-m mount_point] [-a] [-C] [-t]
```

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

The following table contains the options available with the `acfsutil log` command.

Table 6-8 Options for the acfsutil log command

Option	Description
`-f` *`filename`*	Write the in-memory log to the specified file. The default file is `oks.log` in the current directory.
`-s`	Shows the size of the in-memory log file.
`-r` *`n`*`{K|M|G|T|P}`	Sets the size of the in-memory log file.
`-p {avd|ofs|oks}`	Specifies the product for setting the level or querying settings. The is default all products: Oracle ADVM (`avd`), Oracle ACFS (`ofs`), and Oracle Kernel Services (`oks`)
`-l` *`debuglevel`*	Sets the in-memory debug level. The default debug level is `2`. Valid values are `0-6`.
`-n` *`consolelevel`*	Sets the debug level for persistent logging . Other persistent log configuration settings are managed by the `acfsutil plogconfig` command.
`-o` *`wait_time`*	Sets the log size, the debug level, and the product values on all nodes; waits for the number of seconds specified by *wait_time*; dumps in the memory log on all nodes; and then resets the debug level and the log size.
`-q`	Queries the debug settings for a specified product. For example: `acfsutil log —p avd —q`
`-c` *`debugcontext`*	Sets the debug context, internal only.
`-T` *`file_type`*	Sets the debug file type, internal only.
`—m` *`mount_point`*	Specifies to debug only the file system at the specified mount point.
`-a`	Resets the debug logging to log for all file systems.
`-C`	Dumps a memory log on all cluster nodes, and also can be added to `-t` option.
`-t`	Dumps all Hang Manager thread information to in-memory and persistent logs.

The `acfsutil log` command enables you to manage memory diagnostic log files. With none of the options specified, the `acfsutil log` command retrieves and writes the `./oks.log` memory log by default.

The `-o` option performs the following:

1. Sets the log size to `500M` , the log level to `5`, and the product to `ofs` (acfs) for the in-memory log on all nodes
2. Displays an informational message, such as *Blocking for 180 seconds, reproduce problem now*
3. After waiting for the specified number of seconds, then displays *Dumping log on all nodes*
4. Initiates a clusterwide dump of logs
5. Resets the log level to `2` and resets the log size to the default

The `-o` option can be combined with the `-p`, `-l`, and `-r` options if the default product, debug level, or log size settings should be changed.

You must be the `root` user or an Oracle ASM administrator user to run this command.

Examples

The following example shows various ways to run the `acfsutil log` command.

Example 6-6 Using acfsutil log

```
#increase internal log size to 100Mb
$ acfsutil log -r 100M

#increase log level for acfs to 5
$ acfsutil log -l 5 -p ofs

#increase log level for oks to 5
$ acfsutil log -l 5 -p oks

#collect in memory log and place it into /tmp/logfile
$ acfsutil log -f /tmp/logfile

#put trace level back to default, level 2
$ acfsutil log -l 2 -p ofs
$ acfsutil log -l 2 -p oks

# increase log level to 5, wait 3 seconds, and then automatically dump a log on all
nodes,
# log will be in a dated file in directory specified by acfsutil plog -q
$ acfsutil log -l 5 -o 3
Blocking for 3 seconds, reproduce problem now
Dumping log on all nodes

# dump out the stacks of all acfs threads running on the system on all nodes into log
files
# in the directory specified by acfsutil plog -q
$ acfsutil log -t -C
```

acfsutil meta

Purpose

Copies metadata from an Oracle ACFS file system into a separate output file.

Syntax and Description

```
acfsutil meta -h
acfsutil meta  [-v]
               [-g]
               [-g -O -C -S]
               [-O]
               [-C COW_filepath]
               [-S COW_size]
               [-q nn[K|M|G|T]]
               [-l log_file_path]
               [-o acfs_extent_offsets]
               {-f record_oriented_metadata_output_file} [-a accel_device]
volume_device
```

```
acfsutil meta  {-e record_oriented_metadata_input_file [-i]}
               {-f output_filesystem_meta_file_prefix_name}
```

`acfsutil meta -h` displays help text and exits.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

The following table contains the options available with the `acfsutil meta` command. The options are shown in Linux format (also AIX and Solaris format).

Table 6-9 Options for the acfsutil meta command

Option	Description
`–f` *`record_oriented_metadata_output_file`*	Specifies the path name of the output file into which the metadata is copied.
`–g`	Do not perform a block scan looking for lost metadata on the entire volume. Instead, only reads the known metadata blocks. This option is only recommended for file systems in good health.
`–g -O -C -S`	Same as the `-g` option, except also run Oracle ACFS online checker (`fsck` on Linux).
`-O`	Specifies to run the Oracle ACFS online checker (`fsck` on Linux).
`-C` *`COW_filepath`*	Specifies the path to the Copy-On-Write (COW) file for the Oracle ACFS online checker. The path must be on a different Oracle ACFS file system.
`-S` *`COW_size`*	Specifies the size of the Copy-On-Write (COW) file for Oracle ACFS online checker. The size must be large enough so the original blocks can be preserved when the file system modifications are made.
`volume_device`	Specifies a volume device name of the file system which is to be copied.
`–v`	Specifies verbose mode to generate additional diagnostic messages.
`–q` *`nn`*`[K|M|G|T]`	Invokes the metadata collector in *quick sccan* mode. The scanning of the volume stops at the specified size. The number specified must be a positive integer and the value must be at least 200 M. The units are K (Kilobytes), M (Megabytes), G (Gigabytes), or T (Terabytes). If the unit indicator is specified, then it must be appended to the integer. If omitted, the default unit is bytes.
`-l` *`log_file_path`*	Specifies the path to the log file. If not specified, the log file is generated in the current directory with a default name of `acfs.meta.log`.
`–o` *`acfs_extent_offsets`*	Specifies a list of comma separated file offsets from which the meta collector additionally copies data.
`–a` *`accel_device`*	Specifies the location of any associated accelerator device, to be used if the file system is unmountable.

Table 6-9 (Cont.) Options for the acfsutil meta command

Option	Description
`—e` *`record_oriented_metadata_input_file`*	Expands the specified record-oriented metadata file into files that can be used with `fsck` or `acfschkdsk`.
`—i`	The `—i` option with the *-e* option lists the metadata record headers (flags, volume, offset, size) for each record-oriented metadata file.
`—f` *`output_filesystem_meta_file_prefix_name`*	Specifies the path name of the output file from `—e` option operation.

The `acfsutil meta` command operates as a metadata collector to partially copy an Oracle ACFS file system into a separate specified record-oriented output file. The metadata collector reads the contents of the file system specified by the volume device name of an Oracle ACFS file system. This input file system is searched for Oracle ACFS metadata and then all metadata found is written into the specified record-oriented output file. The generated record-oriented output file can be easily transferred to a another system, where it can expanded for diagnostics and analysis, without impact to the original file system at the customer site.

The `-g` option collects only the known good metadata. The `-g` option should not be used with a corrupted file system because the `-g` option does not find lost metadata. Any lost metadata may be important in diagnosing a file system corruption. If the file system is in good repair, the `-g` option may collect the metadata much faster because it does not need to scan the entire physical volume looking for lost metadata blocks.

When the `acfsutil meta -g -O -C -S` command is run on Linux, the Oracle ACFS online checker (`fsck`) runs automatically. The Oracle ACFS online checker, running on behalf of `acfsutil meta -g`, transverses the Oracle ACFS file system metadata using the metadata on-disk pointers, and writes metadata that has been read into the `acfsutil meta-g` metadata collection file. For information about the online `fsck` command on Linux, refer to Oracle ACFS Command-Line Tools for Linux Environments.

To obtain the best copy of the file system with `acfsutil meta`, dismount the file system before running `acfsutil meta`. If it is not possible to dismount the file system, avoid modifying the contents or performing a volume resizing operation while `acfsutil meta` is running.

If the original file system is very large, then the output file can also be very large. Compress the output file when possible to reduce storage space and transmission time.

If the file system has an accelerator device associated with it, `acfsutil meta` also copies the accelerator device data into the record-oriented output file. This operation occurs automatically.

In most circumstances, `acfsutil meta` automatically copies the accelerator device into the record-oriented output file. However, if you have think that the meta collector is not able to find the accelerator device on its own, you can specify the name on the command line with the `-a` option. For example, this situation could occur if the file system is corrupt. Note that using the `-a` option overrides how the meta collector operates automatically, so `-a` should be used carefully.

The output file should not be placed on the Oracle ACFS device that is specified as the input device because the metadata command might process the output file also. The output file should be placed on a file system that can support an output file which is the size of the Oracle ACFS input volume device. The output file should not need all that storage unless the file system is full and contains all metadata and almost no user data, which is unlikely, but not impossible.

The `-q` flag should be used with caution. When `-q` is specified, the meta collector does not scan and copy the entire input file system. Instead, it only scans and copies a predetermined number of bytes and certain data structures which are considered important. The primary use for the `-q` flag is for situations where there is not sufficient time to run the full version of the metadata collector. The `-q` flag should not be used unless it is recommended by the support personnel investigating the problem.

Expanding the record-oriented output file should be performed on the system where diagnosis and analyzes is to be performed. For example, the following command expands record-oriented metadata file on another file system that has adequate storage space.

```
acfsutil meta -e record_oriented_metadata_input_file -f
output_filesystem_meta_file_prefix_name
```

The output of the command provides sparse files suitable for use with `fsck` or `acfschkdsk`. If the record oriented metadata input file includes an accelerator volume, a second sparse output file is created using the same output file name prefix with `.acc` suffix appended. The file system used for the expanded files should support sparse files. Otherwise, the resulting expanded files could be extremely large containing useless zeros where sparse holes could be saving space.

The `acfsutil meta` expanded output file can be read by the `fsck` command in most cases. However, the Oracle ACFS specific `fsck` command on some OS platforms might not access the output file correctly or might not work with a specified flag. You can use a slightly modified `fsck` command form in these cases. For example:

- On Linux, run the command in this format if you are using the `–x` flag:

  ```
  /sbin/fsck.acfs -x filesystem_meta_file.acc filesystem_meta_file
  ```

- On Solaris, run the command in this format if you are using the `–o x` flag:

  ```
  /usr/lib/fs/acfs/fsck -o x=filesystem_meta_file.acc filesystem_meta_file
  ```

- On AIX, run the command in this format:

  ```
  /sbin/helpers/acfs/fsck filesystem_meta_file
  ```

Examples

Example 6-7 shows the use of the `acfsutil meta` command to copy and expand metadata into output files.

Example 6-7 Using the acfsutil meta command

```
$ /sbin/acfsutil meta -f /acfsmounts/critical_apps/
record_oriented_metadata_file /dev/asm/volume1-123
```

You can then expand the output file on the system where diagnostics and analysis are performed.

```
$ /sbin/acfsutil meta -e record_oriented_metadata_file -f filesystem_meta_file
```

acfsutil plogconfig

Purpose

Manages Oracle ACFS persistent logging configuration settings.

Syntax and Description

```
acfsutil plogconfig [-h] [-d persistent_log_directory] [-t] [-q ] [-i seconds]
                    [-s buffer_size] [-l low_water_percent] [-u high_water_percent]
                    [-m max_logfile_size] [-n max_logfile_number] [-w]
```

`acfsutil -h plogconfig` displays help and exits.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

The following table contains the options available with the `acfsutil plogconfig` command.

Table 6-10 Options for the acfsutil plogconfig command

Option	Description
`-d` *`persistent_log_directory`*	Specifies an alternative logging directory. If not specified, the default directory is `$ORACLE_BASE/crsdata/`*`hostname`*`/acfs`.
`-t`	Terminates logging.
`-q`	Queries for and then displays the persistent logging configuration settings.
`-i` *`seconds`*	Specifies the number of seconds for the interval timer.
`-s` *`buffer_size`*	Sets the log buffer size in kilobytes.
`-l` *`low_water_pecent`*	Sets the file write trigger as a percentage.
`-u` *`high_water_pecent`*	Sets the file write throttle as a percentage.
`-m` *`max_logfile_size`*	Sets the maximum log file size in megabytes.
`-n` *`max_logfile_number`*	Sets the maximum number of log files.
`-w`	Reports if the ACFS persistent logs have wrapped.

The `acfsutil plogconfig` command provides a diagnostic tool to manage configuration settings for persistent logging.

All command arguments are optional, but at least one argument must be specified.

> **Note:**
>
> Run the `acfsutil plogconfig` command only when Oracle Support requests configuration of persistent logging settings.

You must be the `root` user or an Oracle ASM administrator user to run this command.

Examples

The following example illustrates the use of the `acfsutil plogconfig` command to display the current configuration settings.

Example 6-8 Using the Oracle ACFS acfsutil plogconfig command

```
# /sbin/acfsutil plogconfig -q

Log Directory Name : /oracle/crsdata/my_host/acfs
Buffer Size (KB) : 64
Low Water Level (percent) : 50
High Water Level (percent) : 75
Timer Interval (Seconds) : 5
Maximum Number of Log Files : 10
Maximum Log File Size (MB) : 100
```

acfsutil tune

Purpose

The `acfsutil tune` command displays or sets the value of Oracle ACFS tunable parameters.

Syntax and Description

```
acfsutil tune -h
acfsutil tune [tunable_name]
acfsutil tune tunable_name=value
```

`acfsutil tune -h` displays help text and exits.

The following table contains the options available with the `acfsutil tune` command.

Table 6-11 Options for the acfsutil tune command

Option	Description
tunable_name	Specifies the name of the tunable parameter.
value	Specifies the value for a tunable parameter.

If a tunable parameter and value are specified, the `acfsutil tune` command sets the value of the tunable parameter in a persistent manner on a particular node.

If a tunable parameter is specified without a value, the `acfsutil tune` command displays the value that is currently assigned to the specified tunable parameter.

If no options are specified, the `acfsutil tune` command displays the tunable parameter values that are currently assigned.

The Oracle ACFS tunable parameter `AcfsMaxOpenFiles` limits the number of open Oracle ACFS files on AIX. Normally you do not have to change the value of this tunable parameter; however, you may want to consider increasing the value if you have a large working set of files in your Oracle ACFS file systems.

The Oracle ACFS tunable parameter `AcfsMaxCachedFiles` sets the maximum number of closed files that remain cached in memory on AIX. Normally you do not have to change value of this tunable parameter; however, you many to consider changing the value to get better performance.

Changing a tunable parameter has an immediate effect and persists across restarts.

You must be a root user to change the value of a tunable parameter.

Examples

The first command displays Oracle ACFS tunable parameters with their values. The second command changes the value of the `AcfsHMTimeOutIntervalSecs` parameter.

Example 6-9 Using the acfsutil tune command

```
$ /sbin/acfsutil tune
AcfsHMTimeOutIntervalSecs    = 60 (0x3c)
AcfsHMSilenceIntervalMins    = 240 (0xf0)

# /sbin/acfsutil tune AcfsHMTimeOutIntervalSecs=120
```

advmutil tune

Purpose

`advmutil tune` displays or sets the value of an Oracle ADVM parameter.

Syntax and Description

```
advmutil -h
advmutil tune [parameter]
advmutil tune parameter=value
```

`advmutil -h` displays help text and exits.

The following table contains the options available with the `advmutil tune` command.

Table 6-12 Options for the advmutil tune command

Option	Description
parameter	Specifies the parameter for which you want to set or display the value.
value	Specifies the value of the specified parameter.

If no options are specified, the `advmutil tune` command displays the parameter values that are currently assigned.

If a parameter is specified without a value, the `advmutil tune` command displays the value that is currently assigned to the specified parameter.

You must be a privileged user to set a parameter.

> **Note:**
>
> Parameters should be set with caution and usually only by Oracle Support Services.

Examples

A parameter that can be specified with `advmutil tune` is the maximum time in minutes for the deadlock timer (`deadlock_timer`). The first command in the example changes the maximum time in minutes for the `deadlock_timer` parameter. The second command displays the current settings of the Oracle ADVM parameters.

Example 6-10 Using advmutil tune

```
$ /sbin/advmutil tune deadlock_timer=20

$ /sbin/advmutil tune
deadlock_timer        = 20 (0x14)
resilver_power        = 8 (0x8)
resilver_regio        = 32 (0x20)
```

Basic Steps to Manage Oracle ACFS Systems

This topic provides an overview of the basic steps when managing Oracle ACFS file systems using command-line utilities.

The examples in this section show operating system commands that are run in a Linux environment system. ASMCMD commands manage the Oracle ADVM volumes, but you can also use SQL*PLus and Oracle ASM Configuration Assistant (ASMCA) to manage volumes.

This section contains these topics:

- About Using Oracle ACFS Command-Line Tools
- Creating an Oracle ACFS File System
- Accessing an Oracle ACFS File System on a Different Node in the Cluster
- Managing Oracle ACFS Snapshots
- Encrypting Oracle ACFS File Systems using OCR as Encryption Key Store
- Encrypting Oracle ACFS File Systems using Oracle Key Vault as Encryption Key Store
- Tagging Oracle ACFS File Systems
- Replicating Oracle ACFS File Systems
- Deregistering, Dismounting, and Disabling Volumes and Oracle ACFS File Systems
- Removing an Oracle ACFS File System and a Volume

About Using Oracle ACFS Command-Line Tools

This topic provides an overview of the use of Oracle ACFS `acfsutil` commands.

The discussions include:

- Privileges to Run Oracle ACFS acfsutil Commands
- Displaying Help for Oracle ACFS acfsutil Commands
- Displaying Oracle ACFS Version Information

- Managing Trace File Space for acfsutil Commands

Privileges to Run Oracle ACFS acfsutil Commands

To run many Oracle ACFS `acfsutil` commands, you must be a system administrator or an Oracle ASM administrator user that has been enabled to run the commands. These privileges are described as follows:

- For system administrator privileges, you must be the `root` user.
- For Oracle ASM administrator user privileges, you must belong to the `OSASM` group and the `oinstall` group (for the `OINSTALL` privilege).

Displaying Help for Oracle ACFS `acfsutil` Commands

You can display help and usage text for Oracle ACFS `acfsutil` commands with the `h` option. When you include a command or a subcommand with the command, the help and usage display is specific to the command and subcommand entered.

The following example illustrates several different ways to display help and usage text, from the most general to more specific. This example shows the `–h` format to display help on a Linux platform.

Example 6-11 Displaying help for Oracle ACFS acfsutil commands

```
$ /sbin/acfsutil -h

$ /sbin/acfsutil -h compress
$ /sbin/acfsutil compress -h

$ /sbin/acfsutil -h repl info
$ /sbin/acfsutil repl info -h

$ /sbin/acfsutil -h sec admin info
$ /sbin/acfsutil sec admin info -h
```

Displaying Oracle ACFS Version Information

You can run `acfsutil version` to display the Oracle ACFS version. For example:

```
$ /sbin/acfsutil version
acfsutil version: 12.2.0.0.3
```

For more information about displaying Oracle ACFS version details, refer to acfsutil version.

Managing Trace Files for acfsutil Commands

The Automatic Diagnostic Repository (ADR) generates a separate internal file for each `acfsutil` command invocation to trace the operation of the command. The space consumed by these trace files can increase significantly, and some features, such as snapshot-based replication, may generate a significant number of trace files.

To limit the number of trace files and the space consumed by them, you can set policy attributes with the Automatic Diagnostic Repository Command Interpreter (ADRCI) utility to purge trace files after a specified retention period. ADRCI considers trace files to be short-lived files and the retention period is controlled by the setting of the `SHORTP_POLICY` attribute. You can view the current retention period for these trace files with the ADRCI `show control` command.

By default, the short-lived files are retained for `720` hours (30 days). The value in hours specifies the number of hours after creation when a given file is eligible for purging. To limit the number of these files and the space consumed by them, you can update the number of hours set for the `SHORTP_POLICY` retention period, such as `240` hours (10 days).

The following steps summarize how to update the retention period for short-lived trace files. These steps should be performed on each node where features like replication will be active.

- Start the Automatic Diagnostic Repository Command Interpreter (ADRCI) utility.

  ```
  $ adrci
  ```

- Display the ADR home directory paths (ADR homes):

  ```
  ADRCI> show homes
  ```

- If more than one home is shown, then set the appropriate home for the trace files you want to administer:

  ```
  ADRCI> set homepath my_specified_homepath
  ```

- Display the current configuration values.

  ```
  ADRCI> show control
  ```

- Update a specific ADRCI configuration value. For example, set `SHORTP_POLICY` to `240` hours (10 days).

  In the displayed `show control` output, check the value of the `SHORTP_POLICY` attribute, which is the retention period in hours for short-lived files. If necessary, set a new retention period for short-lived trace files with the following:

  ```
  ADRCI> set control (SHORTP_POLICY=240)
  ```

If you want to start an immediate purge of the trace files in the current ADR home path, you can use the following command:

```
ADRCI> purge -type TRACE -age number_of_minutes
```

The value *number_of_minutes* controls which files are purged based on the age of the files. Files older than the specified number of minutes are targeted for the purge operation.

Note that automated purges of files in ADR occur on a fixed schedule that is not affected by changes in retention period. In other words, changing the retention period changes how soon after creation files are eligible to be purged, but does not change when a purge occurs. To force a purge, you must request it manually, as shown above.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges
- *Oracle Database Utilities* for information about the Automatic Diagnostic Repository Command Interpreter (ADRCI) utility

Creating an Oracle ACFS File System

You can create an Oracle ACFS file system using the steps in this topic.

To create and verify a file system, perform the following steps:

1. Create an Oracle ADVM volume in a mounted disk group with the ASMCMD `volcreate` command.

   The compatibility parameters `COMPATIBLE.ASM` and `COMPATIBLE.ADVM` must be set to `11.2` or higher for the disk group to contain an Oracle ADVM volume. To use Oracle ACFS encryption, replication, security, or tagging, the disk group on which the volume is created for the file system must have compatibility attributes for `ASM` and `ADVM` set to `11.2.0.2` or higher.

   Start ASMCMD connected to the Oracle ASM instance. You must be a user in the OSASM operating system group.

   When configuring Oracle ADVM volume devices within a disk group, Oracle recommends assigning the Oracle Grid Infrastructure user and Oracle ASM administrator roles to users who have root privileges.

   To create a volume:

   ```
   ASMCMD [+] > volcreate -G data -s 10G volume1
   ```

   When creating an Oracle ADVM volume, a volume device name is created that includes a unique Oracle ADVM persistent disk group number. The volume device file functions in the same manner as any other disk or logical volume to mount file systems or for applications to use directly.

   The format of the volume name is platform-specific.

2. Determine the device name of the volume that was created.

   You can determine the volume device name with the ASMCMD `volinfo` command or from the `VOLUME_DEVICE` column in the `V$ASM_VOLUME` view.

   For example:

   ```
   ASMCMD [+] > volinfo -G data volume1
   Diskgroup Name: DATA

            Volume Name: VOLUME1
            Volume Device: /dev/asm/volume1-123
            State: ENABLED
        ...

   SQL> SELECT volume_name, volume_device FROM V$ASM_VOLUME
        WHERE volume_name ='VOLUME1';

   VOLUME_NAME        VOLUME_DEVICE
   -----------------  --------------------------------------
   VOLUME1            /dev/asm/volume1-123
   ```

3. Create a file system with the Oracle ACFS `mkfs` command.

   Create a file system using an existing volume device.

   For example:

```
$ /sbin/mkfs -t acfs /dev/asm/volume1-123
mkfs.acfs: version                   = 19.0.0.0.0
mkfs.acfs: on-disk version           = 46.0
mkfs.acfs: volume                    = /dev/asm/volume1-123
mkfs.acfs: volume size               = 10737418240  (   10.00 GB )
mkfs.acfs: Format complete.
```

The `root` privilege is not required to run `mkfs`. The ownership of the volume device file dictates who can run this command.

4. Register the file system.

   In an Oracle Grid Infrastructure Clusterware configuration, you can run the `srvctl add filesystem` command to register and automount a file system. For example:

   ```
   # srvctl add filesystem -device /dev/asm/volume1-123 -path /acfsmounts/acfs1
          -user user1,user2,user3 -mtowner sysowner -mtgroup sysgrp -mtperm 755
   ```

   You can also register a file system with the `acfsutil registry` command. For example:

   ```
   $ /sbin/acfsutil registry -a /dev/asm/volume1-123 /acfsmounts/acfs1
   ```

   After registering an Oracle ACFS file system in the cluster mount registry, the file system is mounted automatically on each cluster member listed in the registry entry during the next registry check action. This automatic process runs every 30 seconds and eliminates the requirement to manually mount the file system on each member of the cluster. Registering an Oracle ACFS file system also causes the file system to be mounted automatically whenever Oracle Clusterware or the system is restarted.

   > **Note:**
   >
   > - The `srvctl add filesystem` command is required when an Oracle Database home is installed on an Oracle ACFS file system. In this case, the file system should not be explicitly added to the registry with the Oracle ACFS registration command (`acfsutil registry`).
   > - Oracle ACFS registration is not supported in an Oracle Restart (standalone) configuration, which is a single-instance (non-clustered) environment.
   > - The `root` or `asmadmin` privileges are required to modify the registry.

5. Mount or start the file system.

   If you have previously registered the file system, then start the file system with SRVCTL. For example:

   ```
   $ srvctl start filesystem -device /dev/asm/volume1-123
   ```

   If you have not previously registered the file system, then mount the file system with the Oracle ACFS `mount` command. For example:

   ```
   # /bin/mount -t acfs /dev/asm/volume1-123 /acfsmounts/acf1
   ```

   After an unregistered file system has been mounted, ensure that the permissions are set to allow access to the file system for the appropriate users. For example:

```
# chown -R oracle:dba /acfsmounts/acfs1
```

The `root` privilege is required to run the `mount` command.

6. Create a test file in the file system.

   The user that creates the test file should be a user that is intended to access the file system. This test ensures that the appropriate user can write to the file system.

   For example:

   ```
   $ echo "Oracle ACFS File System" > /acfsmounts/acfs1/myfile
   ```

7. List the contents of the test file that was created in the file system.

   For example:

   ```
   $ cat /acfsmounts/acfs1/myfile
   Oracle ACFS File System
   ```

See Also:

- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- mkfs (Linux environments) for information about the command to create an Oracle ACFS file system
- Managing Oracle ADVM with ASMCMD for information about the `volcreate` command and the `volinfo` command
- acfsutil registry for information about the `acfsutil registry` command to register an Oracle ACFS file system
- About the Oracle ACFS Mount Registry for about information registering an Oracle ACFS file system
- mount (Linux environments) for information about the command to mount an Oracle ACFS file system
- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges
- *Oracle Database Reference* for information about the `V$ASM_VOLUME` view
- *Oracle Clusterware Administration and Deployment Guide* for information about Server Control Utility (SRVCTL) commands

Accessing an Oracle ACFS File System on a Different Node in the Cluster

If the node is part of a cluster, perform the following steps on node 2 to view the test file you created on node 1.

> **Note:**
>
> If the file system has been registered with the Oracle ACFS mount registry, you can skip steps 1 to 3.

1. Enable the volume that was previously created and enabled on node 1.

   Start ASMCMD connected to the Oracle ASM instance. You must be a user in the OSASM operating system group.

   For example:

   ```
   ASMCMD [+] > volenable -G data volume1
   ```

2. View information about the volume that you created on node 1.

   For example:

   ```
   ASMCMD [+] > volinfo -G data volume1
   ```

3. Mount the file system using the Oracle ACFS `mount` command.

   For example:

   ```
   # /bin/mount -t acfs /dev/asm/volume1-123 /acfsmounts/acfs1
   ```

   The `root` privilege is required run the `mount` command.

   After the file system has been mounted, ensure that the permissions are set to allow access for the appropriate users.

4. List the contents of the test file you previously created on the file system.

   For example:

   ```
   $ cat /acfsmounts/acfs1/myfile
   Oracle ACFS File System
   ```

   The contents should match the file created previously on node 1.

> **See Also:**
>
> - Managing Oracle ADVM with ASMCMD for information about the `volenable` command
> - Managing Oracle ADVM with ASMCMDfor information about the `volinfo` command
> - mount (Linux environments) for information about the command to mount Oracle ACFS file systems
> - *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges

Managing Oracle ACFS Snapshots

To create and verify a snapshot on node 1:

1. Create snapshot of the new file system created on node 1.

   For example:

   ```
   $ /sbin/acfsutil snap create mysnapshot_20090725 /acfsmounts/acfs1
   ```

   See "acfsutil snap create".

2. Update the test file in the file system so that it is different than the snapshot.

   For example:

   ```
   $ echo "Modifying a file in Oracle ACFS File System" > /acfsmounts/acfs1/myfile
   ```

3. List the contents of the test file and the snapshot view of the test file.

   For example:

   ```
   $ cat /acfsmounts/acfs1/myfile
   ```

   ```
   $ cat /acfsmounts/acfs1/.ACFS/snaps/mysnapshot_20090725/myfile
   ```

   The contents of the test file and snapshot should be different. If node 1 is in a cluster, then you can perform the same list operation on node 2.

Encrypting Oracle ACFS File Systems using OCR as Encryption Key Store

This topic discusses basic operations to manage encryption on an Oracle ACFS file system on Linux while using the OCR as encryption key store.

The examples in this section show a scenario in which the medical history files are encrypted in an Oracle ACFS file system.

Because the `acfsutil encr set` and `acfsutil encr rekey -v` commands modify the encryption key store, you should back up the Oracle Cluster Registry (OCR) after running these commands to ensure there is an OCR backup that contains all of the volume encryption keys (VEKs) for the file system.

The disk group on which the volume is created for the file system has compatibility attributes for `ASM` and `ADVM` set to `11.2.0.3` or higher.

For the examples in this section, various operating system users, operating system groups, and directories must exist.

The basic steps to manage encryption are:

1. Initialize encryption.

   Run the `acfsutil encr init` command to initialize encryption and create an encryption key store within the OCR. This command must be run one time for each cluster on which encryption is set up.

   For example, the following command initializes encryption for a cluster.

   ```
   # /sbin/acfsutil encr init
   ```

   This command must be run before any other encryption command and requires root or administrator privileges to run.

2. Set encryption parameters.

   Run the `acfsutil encr set` command to set the encryption parameters for the entire Oracle ACFS file system.

For example, the following command sets the `AES` encryption algorithm and a file key length of `128` for a file system mounted on the `/acfsmounts/acfs1` directory.

```
# /sbin/acfsutil encr set -a AES -k 128 -m /acfsmounts/acfs1/
```

The `acfsutil encr set` command also transparently generates a volume encryption key which is stored in the OCR encryption key store that was previously configured with the `acfsutil encr init` command.

This command requires root or administrator privileges to run.

3. Enable encryption.

   Run the `acfsutil encr on` command to enable encryption for directories and files.

   For example, the following command enables encryption recursively on all files in the `/acfsmounts/acfs1/medicalrecords` directory.

```
# /sbin/acfsutil encr on -r /acfsmounts/acfs1/medicalrecords
                         -m /acfsmounts/acfs1/
```

   For users that have appropriate permissions to access files in the `/acfsmounts/acfs1/medicalrecords` directory, they can still read the decrypted files.

   This command can be run by an administrator or the file owner.

4. Display encryption information.

   Run the `acfsutil encr info` command to display encryption information for directories and files.

```
# /sbin/acfsutil encr info -m /acfsmounts/acfs1/
                           -r /acfsmounts/acfs1/medicalrecords
```

   This command can be run by an administrator or the file owner.

Auditing and diagnostic data for Oracle ACFS encryption is saved to log files. .

See Also:

- Oracle ACFS Encryption for more information about Oracle ACFS encryption, including log files
- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- acfsutil encr init for information about initializing encryption
- acfsutil encr set for information about setting encryption parameters
- acfsutil encr on for information about enabling encryption
- acfsutil encr info for information displaying encryption information

Encrypting Oracle ACFS File Systems using Oracle Key Vault as Encryption Key Store

This topic discusses basic operations to manage encryption on an Oracle ACFS file system on Linux while using the Oracle Key Vault as encryption key store.

The examples in this section show a scenario in which the medical history files are encrypted in an Oracle ACFS file system. The steps in this section assume Oracle ACFS security is not configured for the file system; however, you can use both Oracle ACFS security and encryption on the same file system. If you decide to use both security and encryption, then both encryption and security must be initialized for the cluster containing the file system. After security is initialized on the file system, then an Oracle ACFS security administrator runs `acfsutil sec` commands to provide encryption for the file system.

The disk group on which the volume is created for the file system has compatibility attributes for `ASM` and `ADVM` set to `11.2.0.3` or higher.

For the examples in this section, various operating system users, operating system groups, and directories must exist.

The basic steps to manage encryption are:

1. Initialize encryption.

   Run the `acfsutil encr init -o` command to initialize encryption and create an autologin wallet for the Oracle Key Vault. This command must be run one time for each cluster on which encryption is set up.

   For example, the following command initializes encryption for a cluster.

   ```
   # /sbin/acfsutil encr init -o
   ```

   If the Oracle Kev Vault endpoint requires a password for login, the command will prompt for the password and save it within the Oracle Key Vault autologin wallet. The saved password will be used by ACFS to autologin to the Oracle Key Vault. Note that all Oracle Key Vault endpoints within the cluster must have the same endpoint password.

   This command must be run before any other encryption command and requires root or administrator privileges to run.

2. Set encryption parameters.

   Run the `acfsutil encr set` command to set the encryption parameters for the entire Oracle ACFS file system.

   For example, the following command sets the `AES` encryption algorithm and a file key length of `128` for a file system mounted on the `/acfsmounts/acfs1` directory.

   ```
   # /sbin/acfsutil encr set -a AES -k 128 -m /acfsmounts/acfs1/
   ```

   The `acfsutil encr set` command also transparently generates a volume encryption key which is stored in the Oracle Key Vault that was previously configured with the `acfsutil encr init -o` command.

   This command requires root or administrator privileges to run.

3. Enable encryption.

   Run the `acfsutil encr on` command to enable encryption for directories and files.

For example, the following command enables encryption recursively on all files in the /acfsmounts/acfs1/medicalrecords directory.

```
# /sbin/acfsutil encr on -r /acfsmounts/acfs1/medicalrecords
                         -m /acfsmounts/acfs1/
```

For users that have appropriate permissions to access files in the /acfsmounts/acfs1/medicalrecords directory, they can still read the decrypted files.

This command can be run by an administrator or the file owner.

4. Display encryption information.

   Run the acfsutil encr info command to display encryption information for directories and files.

```
# /sbin/acfsutil encr info -m /acfsmounts/acfs1/
                           -r /acfsmounts/acfs1/medicalrecords
```

   This command can be run by an administrator or the file owner.

Auditing and diagnostic data for Oracle ACFS encryption is saved to log files. .

See Also:

- Oracle ACFS Encryption for more information about Oracle ACFS encryption, including log files
- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- acfsutil encr init for information about initializing encryption
- acfsutil encr set for information about setting encryption parameters
- acfsutil encr on for information about enabling encryption
- acfsutil encr info for information displaying encryption information

Encrypting Oracle ACFS File Systems using OCR as Encryption Key Store and encrypting Volume Encryption Keys with an OCI Vault Master Encryption Key

Encrypting Oracle ACFS File Systems using OCR as Encryption Key Store and encrypting Volume Encryption Keys with an OCI Vault Master Encryption Key.

This topic discusses basic operations to manage encryption on an Oracle ACFS file system on Linux while using the OCR as encryption key store and while encrypting Volume Encryption Keys with an OCI Vault Master Encryption Key. Note that this setup is only possible within an OCI environment.

The examples in this section show a scenario in which the medical history files are encrypted in an Oracle ACFS file system.

Because the acfsutil encr set and acfsutil encr rekey -v commands modify the encryption key store, you should back up the Oracle Cluster Registry (OCR) after running these commands to ensure there is an OCR backup that contains all of the volume encryption keys (VEKs) for the file system.

The disk group on which the volume is created for the file system has compatibility attributes for `ASM` and `ADVM` set to `11.2.0.3` or higher.

For the examples in this section, various operating system users, operating system groups, and directories must exist.

The basic steps to manage encryption are:

1. 1. Create an OCI Vault. See Creating a Vault for further instruction.
2. Create a Master Encryption Key within the OCI Vault. See Creating a Master Encryption Key for further instruction.
3. Grant the current OCI instance access to the OCI Vault Master Encryption Key. See Granting OCI Instances access to an OCI Vault Master Encryption Key for further instruction.

   This command can be run by an administrator or the file owner.
4. From the OCI Console page for the OCI Vault, save the OCID of the created Master Encryption Key and the cryptographic/management endpoint URLs for the OCI Vault.
5. Initialize encryption.

   Run the `acfsutil encr init` command initialize encryption and create an encryption key store within the OCR. This command must be run one time for each cluster on which encryption is set up.

   ```
   # /sbin/acfsutil encr init
   ```

   This command must be run before any other encryption command and requires root or administrator privileges to run.
6. Set encryption parameters.

   Run the `acfsutil encr set` command to set the encryption parameters for the entire Oracle ACFS file system. Additionally, provide the previously saved OCID of the Master Encryption Key and the cryptographic/management endpoint URLs for the OCI Vault.

   For example, the following command sets the AES encryption algorithm and a file key length of 128 for a file system mounted on the `/acfsmounts/acfs1` directory.

   ```
   # /sbin/acfsutil encr set -a AES -k 128 -K <OCID of Master Encryption Key> -C 
   <Cryptographic
         Endpoint URL> -M <Management Endpoint URL> -m /acfsmounts/acfs1/
   ```
7. Enable encryption.

   Run the `acfsutil encr on` command to enable encryption for directories and files.

   For example, the following command enables encryption recursively on all files in the `/acfsmounts/acfs1/medicalrecords` directory.

   ```
   # /sbin/acfsutil encr on -r /acfsmounts/acfs1/medicalrecords
                            -m /acfsmounts/acfs1/
   ```
8. Display encryption information.

   Run the `acfsutil encr info` command to display encryption information for directories and files.

```
# /sbin/acfsutil encr set -a AES -k 128 -K <OCID of Master Encryption Key> -
C <Cryptographic
      Endpoint URL> -M <Management Endpoint URL> -m /acfsmounts/acfs1/
```

This command can be run by an administrator or the file owner.

Auditing and diagnostic data for Oracle ACFS encryption is saved to log files.

Granting OCI Instances access to an OCI Vault Master Encryption Key

This topic discusses basic operations to grant OCI Instances access to an OCI Vault Master Encryption Key.

In an OCI environment, the Volume Encryption Keys (VEKs) for an encrypted ACFS file system can be encrypted by an OCI Vault Master Encryption Key. To enable this functionality, the OCI instances with the ACFS file system mounted must be granted access to the OCI Vault Master Encryption Key. To grant this access, an IAM group must be created that encompasses the OCI instances, then an IAM policy must be created to grant the IAM dynamic group access to the OCI Vault Master Encryption Key.

To simplify configuration, it is recommended that all OCI instances requiring access to OCI Vaults Master Encryption Keys be placed in a dedicated OCI compartment. It is also recommended that all OCI Vaults and Master Encryption Keys be placed in a dedicated OCI compartment.

If these recommendations are followed, follow these steps to create an IAM dynamic group containing all of the OCI instances:

1. Copy the OCID of the OCI compartment containing the OCI instances. You can find this OCID on the Compartment Details page of the compartment.
2. Create a dynamic group by following the instructions in To create a dynamic group in Oracle Cloud Infrastructure Documentation.

   When following these instructions, enter a matching rule of this format:

   ```
   ALL {resource.compartment.id ='<compartment-ocid>'}
   ```

Then follow these steps to create an IAM policy granting the IAM dynamic group access to the OCI Vault:

1. Navigate to (or create) an IAM policy in a compartment higher up in your compartment hierarchy than the compartment containing your vaults and keys.
2. Add a policy statement of this format:

   ```
   allow dynamic-group <dynamic-group-name>
   to manage keys
   in compartment <vaults-and-keys-compartment>
   where all {
   target.key.id='<master_encryption_key_ocid>',
   request.permission!='KEY_DELETE',
   request.permission!='KEY_MOVE',
   request.permission!='KEY_IMPORT',
   request.permission!='KEY_BACKUP’
   }
   ```

Other configurations are possible, but will require further understanding of IAM dynamic groups and IAM policies. See the following resources for further information:

- Managing Dynamic Groups

- How Policies Work

Tagging Oracle ACFS File Systems

The operations to manage tagging on directories and files in an Oracle ACFS file system on Linux are discussed in this topic.

The disk group on which the volume is created for the file system has compatibility attributes for `ASM` and `ADVM` set to `11.2.0.3` or higher.

Oracle ACFS implements tagging with Extended Attributes. There are some requirements when using Extended Attributes that should be reviewed.

The steps to manage tagging are:

1. Specify tag names for directories and files.

   Run the `acfsutil tag set` command to set tags on directories or files. You can use these tags to specify which objects are replicated.

   For example, add the `comedy` and `drama` tags to the files in the subdirectories of the `/acfsmounts/repl_data/films` directory.

   ```
   $ /sbin/acfsutil tag set -r comedy /acfsmounts/repl_data/films/comedies

   $ /sbin/acfsutil tag set -r drama /acfsmounts/repl_data/films/dramas

   $ /sbin/acfsutil tag set -r drama /acfsmounts/repl_data/films/mysteries
   ```

   In this example, the `drama` tag is purposely used twice and that tag is changed in a later step.

   You must have system administrator privileges or be the file owner to run this command.

2. Display tagging information.

   Run the `acfsutil tag info` command to display the tag names for directories or files in Oracle ACFS file systems. Files without tags are not be displayed.

   For example, display tagging information for files in the `/acfsmounts/repl_data/films` directory.

   ```
   $ /sbin/acfsutil tag info -r /acfsmounts/repl_data/films
   ```

   Display tagging information for files with the `drama` tag in the `/acfsmounts/repl_data/films` directory.

   ```
   $ /sbin/acfsutil tag info -t drama -r /acfsmounts/repl_data/films
   ```

   You must have system administrator privileges or be the file owner to run this command.

3. Remove and change tag names if necessary.

   Run the `acfsutil tag unset` command to remove tags on directories or files. For example, unset the `drama` tag on the files in the `mysteries` subdirectory of the `/acfsmounts/repl_data/films` directory to apply a different tag to the subdirectory.

   ```
   $ /sbin/acfsutil tag unset -r drama /acfsmounts/repl_data/films/mysteries
   ```

   Add the `mystery` tag to the files in the `mysteries` subdirectory of the `/acfsmounts/repl_data/films` directory.

```
$ /sbin/acfsutil tag set -r mystery /acfsmounts/repl_data/films/mysteries
```

You must have system administrator privileges or be the file owner to run these commands.

> **See Also:**
>
> - Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
> - Oracle ACFS Tagging for information about tagging an Oracle ACFS file system, including requirements for using Extended Attributes in tagging
> - acfsutil tag set for information about specifying tag names
> - acfsutil tag info for information about displaying tag name and details
> - acfsutil tag unset for information about changing and removing tag names

Replicating Oracle ACFS File Systems

The operations to manage Oracle ACFS snapshot-based replication on an Oracle ACFS file system on Linux are discussed in this topic.

The disk groups on which volumes are created for the primary and standby file systems must have compatibility attributes for `ASM` and `ADVM` set to `12.2` or higher. To use a snapshot as a storage location, or to use replication role reversal, the compatibility attributes for Oracle ASM and Oracle ADVM must be set to `18.0` or higher.

The steps to manage replication are:

1. Choose which transport to use for replication. In the 23ai release, on the OL8 / X64 platform, replication supports either Secure Sockets Layer (SSL) or Secure Shell (ssh) as a transport. On other platforms, only ssh is supported as a transport.

   For more information, refer to Choosing a Transport for Oracle ACFS Replication.

2. Determine the user to be employed for replication. Choose or create the replication user, `repluser`. When `ssh` is used as the transport, this is the user who logs in with `ssh` to the standby cluster to apply data replicated from the primary location to the standby location. This user is defined only at the operating system (OS) level and not within Oracle. The user should belong to the groups defined for Oracle ASM administrator access.

   The same user and group identities (including all uids and gids) must be specified for the replication user on both your primary cluster and your standby cluster.

3. Ensure that either SSL or ssh has been configured for replication. The use of SSL by replication involves sharing credentials between the primary and standby clusters. The use of ssh by replication involves the user identity repluser. Configuring ssh involves the following high-level steps:

   - Configuring a user key for repluser on each cluster, then ensuring that key is authorized to log in as repluser on the other cluster.

- Ensuring that a host key for each node in each cluster is known to the user repluser in the other cluster.

For more information on configuring either transport, see Configuring Oracle ACFS Snapshot-Based Replication.

4. Ensure that the snapshots needed by replication can be created at all times.

   At any given point, replication may need to be able to use two concurrent snapshots of the primary location, and one snapshot of the standby location. You can check how many snapshots are in use in the primary and standby file systems using the `acfsutil snap info` command.

   You can confirm how many snapshots are available in each file system (usually 1024) by looking at the flags value in the output of the `acfsutil info fs` command. If the value contains the string `KiloSnap`, then 1024 snapshots are available.

5. Ensure that there is adequate network connectivity between the primary and standby sites. You should verify that the achievable network data transfer rate from primary to standby is substantially larger than the rate of change of data on the primary location.

   One way to estimate network data transfer rate is to start with an observed transfer rate, then reduce it to account for known sources of overhead. For example, you can calculate the elapsed time needed to FTP a 1 GB file from the primary location to the intended standby location, during a period when network usage is low. This provides an estimate of the maximum achievable transfer rate. This rate should be reduced to account for other demands on the network.

   To estimate the average rate of change on the primary, you can use the command `acfsutil info fs` with the `-s` option. This command should be run on each node where the file system that contains the primary location is mounted. The command displays the amount and rate of change to the file system on that node. To compute the total rate of change for the file system, the rate of changed for each node must be aggregated. A reasonable value to use for `-s` is `900`, which would yield a 15 minute sampling interval.

   With the output from `acfsutil info fs` with the `-s` option, you can determine the average rate of change, the peak rate of change, and how long the peaks last. A conservative approach to using this data is to choose the peak rate of change as the target rate that must be accommodated.

   Because replication must transfer all data changed on the primary to the standby, obviously the achievable network transfer rate must be higher, ideally significantly higher, than the target rate of change on the primary. If this is not the case, you should increase network capacity before implementing replication for this primary location and workload.

   For example, assume you have a four node primary cluster and you determine that a 1 GB file can be transferred in 30 seconds, yielding a current FTP transfer rate of 33 MB per second. An estimate of the current replication transfer rate would be approximately 20 MB per second, calculated as follows:

```
33 MB/sec * (1 - 0.2 - (4 * 0.05)) = 33 * 0.6 = ~20 MB/sec
```

   Also, you find that the average rate of change to the primary is 8 GB per hour, with a peak rate of 25 GB per hour. Using the peak rate, you can calculate a target rate of change of approximately 7 MB per second as follows:

```
(25 GB/hour * 1024) / 3600 = ~7 MB/sec
```

   In the scenario that was discussed in this step, you can reasonably expect the network to be able to handle the additional workload from replication.

6. Ensure that there is adequate storage capacity on the primary and standby sites.

   Estimate the storage capacity needed for replication on the sites hosting the primary and standby locations. In the general case, the primary site must store two snapshots of the primary location on an ongoing basis and the standby site must store a single snapshot of the standby location. The space occupied by these snapshots mostly consists of user data or metadata preserved in the snapshot, that has since been modified which triggers a new copy of the data to be created.

   The space occupied by replication-related snapshots can be directly viewed using the command `acfsutil snap info`. On the primary, check for snapshots with the names starting with the string `REPL`. On the standby, look for snapshots for names starting with `SDBACKUP`.

   If you use interval-based replication, the `-i` option to `acfsutil repl init primary`, and if the replication operations are successfully completing within the specified interval, then the size of replication-related snapshots is related to the rate of change of the primary and the length of the interval. For example, with an average rate of change of 8 GB per hour and a two hour replication interval, you would expect that snapshot storage usage is in the range of 16 GB per snapshot.

   Snapshot size does vary with the rate of change of the primary. Another factor is that snapshot size depends in part on the number of files in the file system, as well as the rate of change. Potentially more importantly, if you use constant mode replication, the `-C` option to `acfsutil repl init primary`, or if replication operations are not completing successfully in the interval given with interval—based replication because the interval is too small, the size of replication-related snapshots is difficult to predict in advance. In these cases, observe the size of the snapshots being generated over time and adjust the file system size as needed with the `acfsutil size` command to accommodate normal storage needs in addition in the presence of the snapshots. When collecting this information, a good starting point is to accommodate space for the snapshots to contain the data that is multiple times larger than the collection period, at the average rate of change of the primary.

   While collecting this information, choose a conservative starting point for the amount of space to allow for replication snapshots. For example, you can compute the space needed to store changes to the file system over the collection period as described previously, then you can allocate several times that space for future snapshots.

7. Optionally set tags on directories and files to replicate only selected files in an Oracle ACFS primary location. You can also add tags to files after replication has started.

8. Configure the site hosting the standby location.

   Before replicating an Oracle ACFS a primary storage location, configure the site hosting the standby location by performing the following:

   - To use the file system as a standby location, create a new *standby* file system of adequate size to hold the files replicated from the primary location, as well as a single replication snapshot, and mount the file system. For example:

     ```
     /standby/repl_data
     ```

   - To use a snapshot of an existing file system as a standby location, create a new read-write snapshot, and ensure that the file system is of adequate size to hold the files replicated from the primary location, as well as a single replication snapshot.

- For either kind of standby location, run the `acfsutil repl init standby` command on the site hosting the standby location. For example:

  ```
  $ /sbin/acfsutil repl init standby -u repluser /standby/repl_data
  ```

  > **Note:**
  >
  > If the `acfsutil repl init standby` command is interrupted for any reason, the user must re-create the file system or snapshot used for the location, re-mount the file system if needed, and re-run the command.

  This command requires the name of the replication user and the standby location. When ssh is used as a transport, the specified user is the user under which `ssh`, invoked from the primary cluster, logs in to the standby cluster to apply changes. This user is specified with the **-u** option. For example: `-u` *`repluser`*.

  If the standby location is a file system, it is named with its mount point. For example: `/standby/repl_data`.

  If the standby location is a read-write snapshot, it is named with the snapshot name and the mount point of the containing file system, with the two separated by the `@` character. For example: `drsnap1101@/standby/repl_data`.

  In addition, for either kind of standby location, if the standby cluster contains multiple nodes, then specify a VIP, such as the SCAN VIP, as the network endpoint that replication uses on the standby to receive information from the primary. A hostname should be used as this network endpoint in single-node clusters only.

  You may run this command as either `root` or *repluser*. This is the same for all `acfsutil repl` commands except for the following commands that read, but never modify the replication state:

  – The `acfsutil repl info` and `acfsutil repl bg info` commands may be run by any Oracle ASM administrator user.

  – The `acfsutil repl compare` command is allowed to be run by any Oracle ASM administrator user, but should be run as `root` to maximize its access to the files being compared.

9. After the standby location has been set up, configure the site hosting the primary location and start replication.

   Run the `acfsutil repl init primary` command on the site hosting the primary location. For example:

   ```
   $ /sbin/acfsutil repl init primary -i 2h -s repluser@standby12_vip -m /standby/
   repl_data /acfsmounts/repl_data
   ```

   This command requires the following configuration information.

   - The replication mode:

     – Interval-based, during which a replication operation starts once for a specified interval

     – Constant-mode, during which a new replication operation starts as soon as the previous one ends

  - Manual-mode, during which replication occurs only when requested using the `acfsutil repl sync` command

  If an interval is specified, the option value is the minimum amount of time that elapses between replication operations.

  In all cases, at the start of each operation, replication takes a new snapshot of the primary and compares it to the previous snapshot, if one exists. The changes needed to update the standby to match the primary are then sent to the standby.

  For example, to set up a replication interval of two hours, specify `-i 2h`.

- The user name and network endpoint (VIP name or address, or host name or address) to be used to connect to the site hosting the standby location, specified with the `–s` option. For example: `-s repluser@standby12_vip`
- If the primary location is a file system, then specify the name of the mount point of the file system. For example: `/acfsmounts/repl_data`
- If the primary storage location is a snapshot, then specify the snapshot name plus the mount point of the containing file system, the two separated by the `@` character. For example: `drsnap1101@/acfsmounts/repl_data`
- If the mount point, or snapshot name with the mount point, is different on the site hosting the standby location than it is on the site hosting the primary location, then specify the name of the standby location with the `-m` option. For example: `-m /standby/repl_data`

Because replication is unidirectional, when it is first initiated only the network endpoint specified for the standby cluster is immediately used. However, to support failover (described in a later step), in which the direction of replication may be reversed,`acfsutil repl init primary` also sets up a network endpoint for the primary cluster. The command looks for a SCAN VIP and uses it as the endpoint if present. If no SCAN VIP is identified, then the command uses the hostname of the node where the command runs as the endpoint instead. If the primary cluster contains multiple nodes, then a VIP should always be used as the network endpoint. A hostname should be used as this endpoint only in single-node clusters. You can specify the endpoint to be used for the primary using the `-p` option to `acfsutil repl init primary`.

You can verify the endpoint being used for either cluster using the `acfsutil repl info -c` command. You can change the endpoint at any time using the `acfsutil repl update primary` command.

10. Monitor information about replication on the location.

    The `acfsutil repl info` command displays information about the state of the replication processing on the primary or standby location.

    For example, you can run the following on the site hosting the primary location to display configuration information.

    ```
    $ /sbin/acfsutil repl info -c -v /acfsmounts/repl_data
    ```

    You must have system administrator (the user root) or Oracle AM administrator privileges to run this command.

11. Pause replication momentarily if necessary.

    Run the `acfsutil repl pause` to momentarily stop replication. Run the `acfsutil repl resume` command as soon as possible to resume replication.

For example, the following command pauses replication on the `/acfsmounts/repl_data` file system.

```
$ /sbin/acfsutil repl pause /acfsmounts/repl_data
```

The following command resumes replication on the `/acfsmounts/repl_data` file system.

```
$ /sbin/acfsutil repl resume /acfsmounts/repl_data
```

You must have system administrator or Oracle AM administrator privileges to run the `acfsutil repl pause` and `acfsutil repl resume` commands.

12. Failing over to a standby or turning a standby location into an active location.

    A replication standby can be converted to a replication primary, or can be used by itself as a read/write storage location without replication active. The `acfsutil repl failover` command provides the key support for these operations. This command is run on the standby cluster.

    The `acfsutil repl failover` command begins by verifying the status of the original replication primary. If it finds that the primary is unavailable, then it can optionally retry for a specified period to see if the primary becomes available.

    When both the standby location and corresponding primary location are operating normally, `acfsutil repl failover` reverses the replication relationship. That is, the original standby becomes the current primary, and the original primary becomes the current standby. There is no data loss. Note that failover fails in this case if replication is paused. To enable this case to succeed, run `acfsutil repl resume`.

    If `acfsutil repl failover` has determined that the primary location is not available, then the command restores the standby location to its state as of the last successful replication transfer from the primary, then converts the standby into a primary. Some data loss may occur. After the standby has been converted to a primary, you can do any of the following next:

    - You can wait until the original primary location becomes available. In this case, the original primary is aware that the failover command has been run and converts itself to the replication standby location. Replication is restored, but in the opposite direction.
    - If you do not want to wait, but do want to continue replication, then you can specify a new standby location using the `acfsutil repl update` command. This command also restores replication. Note that the operation is harmless if the original primary returns (as a standby) after you have specified the new standby. The original primary remains idle (as a standby) until you run `acfsutil repl terminate standby` for it.
    - If you want to terminate replication, then run the `acfsutil repl terminate primary` command on the current primary (the original standby).

13. Manage the replication background process.

    Run the `acfsutil repl bg` command to start, stop, or retrieve information about the replication background process.

    For example, run the following command to display information about the replication process for the `/acfsmounts/repl_data` file system.

    ```
    $ /sbin/acfsutil repl bg info /acfsmounts/repl_data
    ```

Note:

When replication is in use, replication snapshots can be viewed using the `acfsutil snap info` command, just as any other snapshot can. You can use this command to get an approximate idea of the space currently occupied by replication snapshots.

Note:

When the Oracle Grid Infrastructure software must be upgraded on the primary and standby sites, either site may be upgraded first, so long as the upgrade starts from GI version 19c or later. Replication will continue to run correctly while this upgrade is in progress. However, any upgrade should be completed on both sites as quickly as possible, so that both sites are running the same GI version on an ongoing basis.

See Also:

- Oracle ACFS Replication for information about replicating Oracle ACFS file systems or snapshots
- Configuring Oracle ACFS Snapshot-Based Replication for information about configuring replication
- Oracle ACFS Features Enabled by Compatibility Attribute Settings for information about Oracle ACFS features and disk group compatibility attribute settings
- Configuring `ssh` for Use With Oracle ACFS Replication
- acfsutil repl init for information initiating replication
- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM user privileges
- acfsutil repl bg for more information about managing replication background operations
- acfsutil repl pause and acfsutil repl resume for more information about pausing and resuming replication operations
- acfsutil repl info for information about displaying replication details
- Creating an Oracle ACFS File System for information about creating a file system
- Tagging Oracle ACFS File Systems for information about the steps to tag files

Deregistering, Dismounting, and Disabling Volumes and Oracle ACFS File Systems

This topic discusses the operations to deregister or dismount a file system and disable a volume.

Deregistering an Oracle ACFS File System

You can deregister an Oracle ACFS file system if you do not want the file system to be automatically mounted.

For example:

```
$ /sbin/acfsutil registry -d /acfsmounts/acfs1
```

If you deregister a file system, then you must explicitly mount the file system after Oracle Clusterware or the system is restarted.

For more information about the registry, refer to About the Oracle ACFS Mount Registry. For information about `acfsutil registry`, refer to acfsutil registry.

Dismounting an Oracle ACFS File System

You can dismount a file system without deregistering the file system or disabling the volume on which the file system is mounted.

For example, you can dismount a file system and run `fsck` to check the file system.

```
# /bin/umount /acfsmounts/acfs1

# /sbin/fsck -y -t acfs /dev/asm/volume1-123
```

After you dismount a file system, you must explicitly mount the file system.

Use `umount` on Linux systems. For information about the command to dismount a file system, refer to umount.

Use `fsck` on Linux systems to check a file system. For information about the command to check a file system, refer to fsck (offline mode).

Disabling a Volume

To disable a volume, you must first dismount the file system on which the volume is mounted.

For example:

```
# /bin/umount /acfsmounts/acfs1
```

After a file system is dismounted, you can disable the volume and remove the volume device file.

For example:

```
ASMCMD> voldisable -G data volume1
```

Dismounting the file system and disabling a volume does not destroy data in the file system. You can enable the volume and mount the file system to access the existing data. For information about `voldisable` and `volenable`, refer to Managing Oracle ADVM with ASMCMD.

Removing an Oracle ACFS File System and a Volume

You can remove an Oracle ACFS file system and volume with `acfsutil` and ASMCMD commands.

To permanently remove a volume and Oracle ACFS file system, perform the following steps. These steps destroy the data in the file system.

1. Deregister the file system with `acfsutil registry -d`.

   For example:

   ```
   $ /sbin/acfsutil registry -d /acfsmounts/acfs1
   acfsutil registry: successfully removed ACFS mount point
      /acfsmounts/acfs1 from Oracle Registry
   ```

2. Dismount the file system with the `umount` command.

   For example:

   ```
   # /bin/umount /acfsmounts/acfs1
   ```

   You must dismount the file system on all nodes of a cluster.

3. Remove the file system with `acfsutil rmfs`.

   If you were not planning to remove the volume in a later step, this step is necessary to remove the file system. Otherwise, the file system is removed when the volume is deleted.

   For example:

   ```
   $ /sbin/acfsutil rmfs /dev/asm/volume1-123
   ```

4. Optionally you can disable the volume with the ASMCMD `voldisable` command.

   For example:

   ```
   ASMCMD> voldisable -G data volume1
   ```

5. Delete the volume with the ASMCMD `voldelete` command.

   For example:

   ```
   ASMCMD> voldelete -G data volume1
   ```

> **See Also:**
>
> - acfsutil registry for information about running `acfsutil registry`
> - umount for information about running the `umount` command
> - acfsutil rmfs for information about running the `acfsutil rmfs` command
> - Managing Oracle ADVM with ASMCMD for information about running the `voldisable` command
> - Managing Oracle ADVM with ASMCMD for information about running the `voldelete` command

Oracle ACFS Command-Line Tools for Linux Environments

This topic provides a summary of the Oracle ACFS commands for Linux environments.

Table 6-13 lists the Oracle ACFS commands for Linux environments with brief descriptions. The commands in Table 6-13 have been extended with additional options to support Oracle ACFS. All other Linux file system commands operate without change for Oracle ACFS.

For example, Oracle ACFS adds a set of Oracle ACFS-specific mount options to those provided with the base operating system platform. You should review both the mount options for the Linux platforms in addition to the Oracle ACFS-specific options for the complete set of file system mount options.

File systems on Oracle ADVM volumes that are not Oracle ACFS file systems, such as `ext3`, are managed with the same Linux commands that are listed in Table 6-13 using the file-specific options for the type of file system. You can refer to the `man` pages for options available for the Linux commands in Table 6-13.

> **Note:**
>
> When using Security-Enhanced Linux (SELinux) in enforcing mode with Oracle ACFS, ensure that the Oracle ACFS file systems are mounted with an SELinux default context. Refer to your Linux vendor documentation for information about the context mount option.

Table 6-13 Summary of Oracle ACFS commands for Linux environments

Command	Description
fsck (offline mode)	Checks and repairs a dismounted Oracle ACFS file system.
fsck (online mode)	Checks and repairs a mounted Oracle ACFS file system.
mkfs	Creates an Oracle ACFS file system.
mount	Mounts an Oracle ACFS file system.
umount	Dismounts an Oracle ACFS file system.

fsck (offline mode)

Purpose

Checks and repairs a dismounted Oracle ACFS file system.

Syntax and Description

```
fsck -t acfs -h /dev/null
fsck [-a|-f] [-v]  -t acfs [-n|-y] [-x file_name] volume_device
```

`fsck -t acfs -h /dev/null` displays usage text and exits.

Table 6-14 contains the options available with the `fsck` command.

Table 6-14 Options for the offline fsck command on Linux

Option	Description
`-a`	Specifies to automatically repair the file system.
`-f`	Forces the file system into mountable state without completing a file system check or repair.
`-v`	Specifies verbose mode. The progress is displayed as the operation occurs. Running in verbose mode can impact performance.
`-t acfs`	Specifies the type of file system on Linux. `acfs` designates the Oracle ACFS type.
`-n`	Answers no to any prompts.
`-y`	Answers yes to any prompts.
`-x` *`file_name`*	Specified for accelerator data collected by `acfsutil meta`. Only used for this type of data.
`volume_device`	Specifies the primary Oracle ADVM volume device.

`fsck` checks and repairs an existing Oracle ACFS file system. This topic describes an offline mode of the `fsck` command that can only be run on a dismounted file system. For information about running `fsck` on a mounted file system, refer to fsck (online mode).

`root` privileges are required to run `fsck`. The Oracle ACFS driver must be loaded for `fsck` to work.

By default, `fsck` only checks for and reports any errors. The `-a` flag must be specified to instruct `fsck` to repair errors in the file system. Do not interrupt `fsck` during the repair operation.

In a few cases, `fsck` prompts for questions before proceeding to check a file system. These cases include:

- If `fsck` detects that another `fsck` is in progress on the file system
- If `fsck` detects that the Oracle ACFS driver is not loaded
- If the file system does not appear to be Oracle ACFS

In checking mode, `fsck` also prompts if there are transaction logs that have not been processed completely due to an incomplete shutdown. To run in a non-interactive mode, include either the `-y` or `-n` options to answer yes or no to any questions.

`fsck` creates working files before it checks a file system. These working files are created in `/usr/tmp` if space is available. `/tmp` is used if `/usr/tmp` does not exist. If insufficient space is available in the `tmp` directory, `fsck` attempts to write to the current working directory. The files that `fsck` creates are roughly the size of the file system being checked divided by 32K. At most three such files are allocated. For example, a 2 GB file system being checked causes `fsck` to generate one to three 64K working files in the `/usr/tmp` directory. These files are deleted after `fsck` has finished.

In the event that `fsck` finds a file or directory in the file system for which it cannot determine its name or intended location (possibly due to a corruption in its parent directory), it places this object in the `/lost+found` directory when `fsck` is run in fix mode. For security reasons only the `root` user on Linux can read files in `/lost+found`. If the administrator can later determine the original name and location of the file based on its contents, the file can be moved or copied into its intended location.

The file names in the `/lost+found` directory are in the following formats:

```
parent.id.file.id.time-in-sec-since-1970
parent.id.dir.id.time-in-sec-since-1970
```

The *`id`* fields are the internal Oracle ACFS numeric identifiers for each file and directory in the file system.

You can use `acfsutil info id` *`id mount_point`* to attempt to determine the directory associated with `parent.`*`id`*. This directory is assumed to be where the deleted object originated. For information about `acfsutil info`, see acfsutil info file.

If the parent directory is not known, the parent `id` field is set to `UNKNOWN`.

> **Note:**
>
> It is not possible to see the contents of the `/lost+found` directory from a snapshot.

Examples

The following example shows how to check and repair a dismounted Oracle ACFS file system.

Example 6-12 Using the fsck command in offline mode

```
# /sbin/fsck -a -y -t acfs /dev/asm/volume1-123
```

fsck (online mode)

Purpose

Checks and repairs a mounted Oracle ACFS file system.

Syntax and Description

```
fsck -t acfs -h /dev/null

fsck [-a] [-v] -t acfs -O mount_point -- -C cow_file_path -S cow_file_size
fsck.acfs [-a] [-v] -O -C cow_file_path -S cow_file_size mount_point
```

> **Note:**
>
> The latter two syntax formats are equivalent. When invoking the Linux `fsck` command, the `--` separator must be placed before either the `-C` or `-S` options to ensure that the options are correctly passed to the Oracle ACFS `fsck` command. When directly invoking the ACFS `fsck.acfs` command, the `--` separator is not required.

`fsck -t acfs -h /dev/null` displays usage text and exits.

Table 6-15 contains the options available with the `fsck` command.

Table 6-15 Options for the online fsck command on Linux

Option	Description
`-a`	Specifies to automatically repair the file system.
`-v`	Specifies verbose mode. The progress is displayed as the operation occurs. Running in verbose mode can impact performance.
`-t acfs`	Specifies the type of file system on Linux. `acfs` designates the Oracle ACFS type.
`-O`	Specifies to run in an online mode that enables checking of a mounted Oracle ACFS file system.
`-C` *`cow_file_path`*	Specifies the location of the Copy-On-Write (COW) file required by online `fsck`.
`-s` *`cow_file_size`*	Specifies the size of the Copy-On-Write (COW) file required by online `fsck`. The quantity specified can be in units of `K` (kilobytes), `M` (megabytes), `G` (gigabytes), `T` (terabytes), or `P` (petabytes). If a unit is not specified, the default is bytes.
`mount_point`	Specifies the directory where the file system is mounted.

`fsck` checks and repairs an existing Oracle ACFS file system. This topic describes an online mode of the `fsck` command that can be run on a mounted file system. For information about running `fsck` in offline mode, refer to fsck (offline mode).

When online `fsck` is run on a mounted file system, online `fsck` initially creates a point-in-time snapshot of the file system metadata. Online `fsck` then runs file system checks against this metadata snapshot. Because the file system is mounted and live, file system metadata can be created or updated underneath online `fsck`. However, any new or updated metadata is not checked by online `fsck` because online `fsck` checks only the initial metadata snapshot.

Online `fsck` utilizes a special file called a Copy-On-Write (COW) file. While online `fsck` is running on a file system, whenever any file system metadata must be updated, the original version of the metadata is first copied into the COW file to preserve it for file system checks. The COW file helps preserve the metadata snapshot that is checked by online `fsck`.

The size of the COW file must be chosen in advance. As a result, the COW file has a limited amount of space for preserving copies of metadata. If the COW file runs out of space while preserving copies of metadata, then online `fsck` may be forced to stop. The size of the COW file must be chosen carefully to avoid running out of space during execution of online `fsck`. The appropriate size of the COW file depends on the amount of file system metadata that may change during the execution of online `fsck`, which is workload dependent and may need to be determined empirically. Oracle recommends the size of the COW file to start at 5% of the file system undergoing online `fsck`.

The COW file must be placed on a separate Oracle ACFS file system. The separate Oracle ACFS file system must be mounted on the same nodes as the Oracle ACFS file system undergoing online `fsck`.

The COW file is a temporary file that supports the execution of online `fsck` and is deleted after online `fsck` is complete.

The Oracle ACFS file system undergoing online `fsck` must have the same mount point path on all nodes. The Oracle ACFS file system storing the COW file must also have the same mount point path on all nodes.

Online `fsck` has the following restrictions and limitations while the command is running on an Oracle ACFS file system:

- Another instance of online `fsck` cannot be started within the same cluster.
- A mount of the file system on a new node causes online `fsck` to stop.
- If the separate Oracle ACFS file system containing the COW file is unmounted on any node, then that situation may cause online `fsck` to stop.
- Deleted snapshots within the file system do not have their storage freed until the completion of online `fsck`.
- Resizing the file system with `acfsutil size` is not allowed and returns an error. Automatic resizing is also not allowed until online `fsck` completes.
- A freeze of the file system with `acfsutil freeze` causes online `fsck` to wait for the freeze to end.

By default, online `fsck` only checks for and reports any errors. The `-a` flag must be specified to instruct online `fsck` to repair errors in the file system. Note that online `fsck` can only repair a limited set of file system errors. Most file system errors require offline `fsck` for repair.

Examples

The following example shows how to check and repair a mounted Oracle ACFS file system.

Example 6-13   Using the fsck command in online mode

```
# /sbin/fsck -a -t acfs -O /acfs_mountpoint_1 -- -C /acfs_mountpoint_2/cow_file -S 512M

# /sbin/fsck.acfs -O -C /acfs_mountpoint_2/cow_file -S 512M /acfs_mountpoint_1
```

mkfs

Purpose

Creates an Oracle ACFS file system.

Syntax and Description

```
mkfs -t acfs -h
mkfs [-v] [-f] -t acfs [-i {512 | 4096}] [-n name ] [-a accelerator_volume]
     [-c release_version] volume_device [size]
```

`mkfs -t acfs -h` displays usage text and exits.

Table 6-16 contains the options available with the `mkfs` command.

Table 6-16 Options for the Linux mkfs command

Option	Description
`-t acfs`	Specifies the type of file system on Linux. `acfs` designates the Oracle ACFS type.
`-v`	Specifies verbose mode. The progress is displayed as the operation occurs.
`-i {512 | 4096}`	Specifies a file system with metadata block size of 512 or 4096 bytes.
`-n` *name*	Specifies the name for the file system. A name can be a maximum of 64 characters. `acfsutil info fs` returns the name if one was specified.
`-f`	Specifies the force option. This action creates the file system even if there is an existing Oracle ACFS on the volume device, although only if the file system is dismounted. This option overwrites structures on the original file system. Use this option with caution.
`-h`	Displays the usage help text and exits.
size	Specifies the size that the file system should consume on the named device. The quantity specified can be in units of `K` (kilobytes), `M` (megabytes), `G` (gigabytes), `T` (terabytes), or `P` (petabytes). If a unit is not specified, the default is bytes. If this option is not specified, the entire device is consumed.
`-a` *accelerator_volume*	Specifies a secondary Oracle ACFS volume that is used by Oracle ACFS to store critical metadata.
`-c` *release_version*	Creates an Oracle ACFS file system with Oracle ACFS release compatibility set to the value of *release_version*.
volume_device	Specifies an existing Oracle ADVM device file that is to be formatted. This device is the primary volume.

You can use `mkfs` to create the on disk structure needed for Oracle ACFS file system to be mounted. The `mkfs` command is the traditional Linux command used to build a file system. After `mkfs` runs successfully, the `USAGE` column in the `V$ASM_VOLUME` view displays `ACFS`. Oracle ADVM volumes are created with the ASMCMD `volcreate` command. For information about the `volcreate` command, refer to volcreate.

The value specified with the `-c` *`release_version`* option must be greater than or equal to the value of `COMPATIBLE.ADVM` for the disk group, and must be less than or equal to the running Oracle Grid infrastructure release version. If `–c` *`release_version`* is not specified, the value of `COMPATIBLE.ADVM` is used. After the compatibility is set, it cannot be downgraded. The `–c` *`release_version`* option can be used in situations where it is not possible, or desirable, to update `COMPATIBLE.ADVM` and `COMPATIBLE.ASM` for the disk group, but you want to use an Oracle ACFS feature that requires a compatibility increase. After the Oracle ACFS compatibility is updated, it is no longer possible to mount the file system with an older Oracle Grid infrastructure release. To change the compatibility of an existing file system, refer to acfsutil compat set. For information about Oracle ASM disk group compatibility attributes, see Oracle ACFS Features Enabled by Compatibility Attribute Settings.

The minimum file system size is 200 MB for the 512 byte metadata block size format and 512 MB for the 4 KB metadata block size format.

The Oracle ACFS driver must be loaded for `mkfs` to operate successfully.

`root` privilege is not required. The ownership of the volume device file dictates who can run this command.

Oracle ACFS Accelerator Volume

Using an accelerator volume can improve performance by reducing the time to access and update Oracle ACFS metadata. You should create the accelerator volume on a disk group with storage that is significantly faster than the storage of the primary volume. The accelerator volume is created with the `-a` option of the `mkfs` command. For more information, refer to Oracle ACFS Accelerator Volume.

4 KB Sector/Metadata

The following list summarizes the use of the `-i` option and metadata block size based on the value of the `COMPATIBLE.ADVM` disk group attribute.

- If `COMPATIBLE.ADVM` is set to `20.1` or greater, then the primary and accelerator volumes can use a mix of different logical sector sizes, such as 512 bytes and 4096 bytes.
- If `COMPATIBLE.ADVM` is set to `12.2` or greater, then the metadata block size is `4096` bytes by default.
- If `COMPATIBLE.ADVM` is set to less than `12.2`, then the block size is set to 512 bytes.
- If `COMPATIBLE.ADVM` is set to less than `12.2` and the logical sector size is not 512 bytes, then the command fails.

User data IO continues to support transfers as small as 512 bytes for normal user IO requests. When the Oracle ADVM volume of the file system has a logical disk sector size of 4 K, user Direct IO requests should be aligned on 4 K file offsets and be a multiple of 4 KB lengths for the best performance. Note that only 4 K metadata block size formats support ADVM volumes with 4 K logical disk sectors.

The `COMPATIBLE.ADVM` value must be set to `12.2` or greater to format an Oracle ADVM volume whose logical disk size is 4096 or to use the `-i 4096` option to format an Oracle ADVM volume whose logical disk sector size is 512 bytes.

Examples

Before creating an Oracle ACFS file system, first determine which Oracle ADVM volume devices are available. You can use the ASMCMD `volinfo` command to display information about the volumes and volume devices. For example:

```
ASMCMD [+] > volinfo -a
...
         Volume Name: VOLUME1
         Volume Device: /dev/asm/volume1-123
         State: ENABLED
...
```

For information about the `volcreate` and `volinfo` commands, refer to Managing Oracle ADVM with ASMCMD.

This command creates an Oracle ACFS file system on a volume device file.

Example 6-14 Using the mkfs command

```
$ /sbin/mkfs -t acfs /dev/asm/volume1-123
```

Example 6-15 Using the mkfs command to create an accelerator volume

This command creates an Oracle ACFS file system with an accelerator volume.

```
$ /sbin/mkfs -t acfs -a /dev/asm/volume2-130 /dev/asm/volume1-127
```

mount

Purpose

Mounts an Oracle ACFS file system.

Syntax and Description

```
mount -h
mount [-v] -n -t acfs [-o options] volume_device mount_point
mount
```

`mount -h` displays usage text and exits.

Table 6-17 contains the options available with the `mount` command.

Table 6-17 Options for the Linux mount command

Option	Description
`-h`	Displays the usage help text and exits.
`-t acfs`	Specifies the type of file system on Linux. `acfs` designates the Oracle ACFS type.
`-v`	Specifies verbose mode. The progress is displayed as the operation occurs.
`-n`	Specifies not to update the `/etc/mtab` file.

Table 6-17 (Cont.) Options for the Linux mount command

Option	Description
`-o`	Options are specified with the `-o` flag followed by a comma-delimited string of options. The following options are available: • `all` Reads the Oracle ACFS managed file systems in the Oracle Clusterware namespace, mounting all file systems with the `AUTO_START=always`. When the `-o all` option is specified, other `-o` options are ignored. To specify mount options for a registry entry, include those options with the `acfsutil registry` command when you add the entry to the registry. • `ro` Mounts the file system in read-only mode. • `norootsuid` Fails the execution of binaries by non-root users whose permissions allow set user Id execution, and are owned by root. An attempt to run these executables as a non-root user fails with a permission denied error. • `rootsuid` Allows the execution of binaries by non-root users of set user Id files owned by root. This is the default action. • `nodeleteopen` Fails the deletion of any file that is open.
`volume_device`	Specifies a primary Oracle ADVM volume device file that has been formatted by `mkfs`. *`device`* is required but can be a dummy value.
`mount_point`	Specifies the directory where the file system is mounted. This directory must exist before you run the `mount` command.

`mount` attaches a file system to the Oracle ACFS hierarchy at the mount point that is the name of a directory. The mount happens on the node where the mount command was issued. The mount command returns an error if the file system is not in a dismounted state on this node.

It is not always possible to return the cause of a mount failure to the `mount` command. When this happens Oracle ACFS writes the cause of the failure to the system console and associated system log file.

After `mount` runs successfully, the `MOUNTPATH` field in the `V$ASM_VOLUME` view displays the directory name on which the file system is now mounted.

An Oracle ACFS file system should only be mounted on one mount point. The same mount point name should be used on all cluster members.

The `mount` command lists all mounted file systems if it is run with no parameters.

`root` privilege is required to run `mount`.

Examples

The first example shows how to mount `volume1-123` on the mount point `/acfsmounts/acfs1`. The second example shows how to mount all the registered Oracle ACFS file systems. The dummy names (`none`) have been entered for the device and directory as they are required, but not used, when the `all` option is specified.

Example 6-16 Using the mount command

```
# /bin/mount -t acfs /dev/asm/volume1-123 /acfsmounts/acfs1

# /bin/mount -t acfs -o all none none
```

umount

Purpose

Dismounts an Oracle ACFS file system.

Syntax and Description

```
umount -h
umount [-v] -l -n volume_device |mount_point
umount -a -l -n [-t acfs]
```

`umount -h` displays usage text and exits.

Table 6-18 contains the options available with the `umount` command.

Table 6-18 Options for the Linux umount command

Option	Description
`-h`	Displays the usage help text and exits.
`-t acfs`	Specifies the type of file system on Linux. `acfs` designates the Oracle ACFS type.
`-v`	Specifies verbose mode. The progress is displayed as the operation occurs.
`-a`	Specifies to dismount all Oracle ACFS file systems on this node.
`-l`	Specifies a lazy unmount. The Oracle ACFS file system is dismounted, but the cleanup of all references to the file system occurs when the system is no longer busy.
`-n`	Specifies not to update the `/etc/mtab` file.
`volume_device`	Specifies an Oracle ADVM volume device file that has been formatted by `mkfs`.
`mount_point`	Specifies the directory where the file system is mounted. This directory must exist before you run the `mount` command.

`umount` detaches an Oracle ACFS from the file system hierarchy on the current node.

The `umount` command does not return control back to the operating system prompt until the file system has been completely unmounted.

If the file system is busy, `umount` fails.

`root` privileges are required to run the `umount` command.

Examples

The following examples show how to dismount an Oracle ACFS file system. The first example uses the volume device file and the second example uses the file system.

Example 6-17 Using the umount command

```
# /bin/umount /dev/asm/volume1-123

# /bin/umount /acfsmounts/acfs1
```

Oracle ACFS Command-Line Tools for the Solaris Environment

This topic provides a summary of the Oracle ACFS commands for Solaris.

> **Note:**
>
> If a disk group is dismounted with the force option on Solaris, any Oracle ADVM volumes device files that were on that disk group remain on the system. These files are removed when the disk group is remounted.

Table 6-19 lists the Oracle ACFS commands for Solaris with brief descriptions. The commands in Table 6-19 have been extended with additional options to support Oracle ACFS on Solaris.

Table 6-19 Summary of Oracle ACFS commands for Solaris

Command	Description
fsck	Checks and repairs an Oracle ACFS file system on Solaris.
mkfs	Creates an Oracle ACFS file system on Solaris.
mount	Mounts an Oracle ACFS file system on Solaris.
umount/umountall	Dismounts an Oracle ACFS file system on Solaris.

fsck

Purpose

Checks and repairs an Oracle ACFS file system on the Solaris operating system.

Syntax and Description

```
fsck -F acfs -o h /dev/null
fsck -F acfs [{-n|N}|{-y|Y}] [-o options] volume_device
```

`fsck -F acfs -o h /dev/null` displays usage text and exits.

Table 6-20 contains the options available with the `fsck` command.

Table 6-20 Options for the Solaris fsck command

Option	Description
`-F acfs`	Specifies the type of file system on Solaris. `acfs` designates the Oracle ACFS type.
`-n |N`	Answers no to any prompts.
`-y|Y`	Answers yes to any prompts.
`-o`	Specifies that options follow (a, f, h, v). Options are preceded with the `-o` flag and entered as a comma-delimited string. For example: `-o a,v` • `a` Specifies to automatically fix the file system. • `f` Forces the file system into mountable state without completing a file system check or fix. • `h` Displays the usage text and exits. • `v` Specifies verbose mode. The progress is displayed as the operation occurs. Running in verbose mode can impact performance.
`volume_device`	Specifies an Oracle ADVM device file.

`fsck` checks and repairs an existing Oracle ACFS file system. This command can only be run on a dismounted file system. `root` privileges are required to run `fsck`. The Oracle ACFS driver must be loaded for `fsck` to work.

By default, `fsck` only checks for and reports any errors. The `-o a` option must be specified to instruct `fsck` to fix errors in the file system. Do not interrupt `fsck` during the repair operation.

In a few cases, `fsck` prompts for questions before proceeding to check a file system. These cases include:

- If `fsck` detects that another `fsck` is in progress on the file system
- If `fsck` detects that the Oracle ACFS driver is not loaded
- If the file system does not appear to be Oracle ACFS

In checking mode, `fsck` also prompts if there are transaction logs that have not been processed completely due to an incomplete shutdown. To run in a non-interactive mode, include either the `-y` or `-n` options to answer yes or no to any questions.

`fsck` creates working files before it checks a file system. These working files are created in `/usr/tmp` if space is available. `/tmp` is used if `/usr/tmp` does not exist. If insufficient space is available in the `tmp` directory, `fsck` attempts to write to the current working directory. The files that `fsck` creates are roughly the size of the file system being checked divided by 32K. At most three such files are allocated. For example, a 2 GB file system being checked causes `fsck` to generate one to three 64K working files in the `/usr/tmp` directory. These files are deleted after `fsck` has finished.

In the event that `fsck` finds a file or directory in the file system for which it cannot determine its name or intended location (possibly due to a corruption in its parent directory), it places this object in the `/lost+found` directory when `fsck` is run in fix mode. For security reasons only the `root` user on Linux can read files in `/lost+found`. If the administrator can later determine the original name and location of the file based on its contents, the file can be moved or copied into its intended location.

The file names in the `/lost+found` directory are in the following formats:

```
parent.id.file.id.time-in-sec-since-1970
parent.id.dir.id.time-in-sec-since-1970
```

The *id* fields are the internal Oracle ACFS numeric identifiers for each file and directory in the file system.

You can use `acfsutil info id` *id mount_point* to attempt to determine the directory associated with `parent.`*id*. This directory is assumed to be where the deleted object originated. For information about `acfsutil info`, see "acfsutil info file".

If the parent directory is not known, the parent `id` field is set to `UNKNOWN`.

> **Note:**
>
> It is not possible to see the contents of the `/lost+found` directory from a snapshot.

Examples

The following example shows how to check and repair an Oracle ACFS file system.

Example 6-18 Using the fsck command

```
# /usr/sbin/fsck -F acfs -y -o a /dev/asm/volume1-123
```

mkfs

Purpose

Creates an Oracle ACFS file system on the Solaris operating system.

Syntax and Description

```
mkfs -F acfs -o h /dev/null
mkfs -F acfs [-o options] [-c release_version] volume_device [size]
```

`mkfs -F acfs -o h /dev/null` displays usage text and exits.

Table 6-21 contains the options available with the `mkfs` command.

Table 6-21 Options for the Solaris mkfs command

Option	Description
`-F acfs`	Specifies the type of file system on Solaris. `acfs` designates the Oracle ACFS type.

Table 6-21 (Cont.) Options for the Solaris mkfs command

Option	Description
`-o` *options*	Specifies that options follow (f, h, n, v). Options are preceded with the `-o` flag and entered as a comma-delimited string. For example: `-o f,v` • `h` Displays the usage text and exits. • `f` Specifies the force option. This action creates the file system even if there is an existing Oracle ACFS on the volume device, although only if the file system is dismounted. This option overwrites structures on the original file system. Use this option with caution. • `i=512 | i=4096` Specifies a file system with metadata block size of 512 or 4096 bytes. • `n=`*name* Specifies the name for the file system. A name can be a maximum of 64 characters. `acfsutil info fs` returns the name if one was specified. • `v` Specifies verbose mode. The progress is displayed as the operation occurs.
`-c` *release_version*	Creates an Oracle ACFS file system with Oracle ACFS release compatibility set to the value of *release_version*.
volume_device	Specifies an Oracle ADVM device file.
size	Specifies the size of the file system in 512-byte units or in units of `K`|`M`|`G`|`T`|`P`. Units specified are in `K` (kilobytes), `M` (megabytes), `G` (gigabytes), `T` (terabytes), or `P` (petabytes).

You can use `mkfs` to create the on disk structure needed for Oracle ACFS file system to be mounted. The `mkfs` command is the traditional Linux command used to build a file system. After `mkfs` runs successfully, the `USAGE` column in the `V$ASM_VOLUME` view displays `ACFS`. `root` privilege is not required. The ownership of the volume device file dictates who can run this command.

The value specified with the `-c` *release_version* option must be greater than or equal to the value of `COMPATIBLE.ADVM` for the disk group, and must be less than or equal to the running Oracle Grid infrastructure release version. If `–c` *release_version* is not specified, the value of `COMPATIBLE.ADVM` is used. After the compatibility is set, it cannot be downgraded. The `–c` *release_version* option can be used in situations where it is not possible, or desirable, to update `COMPATIBLE.ADVM` and `COMPATIBLE.ASM` for the disk group, but you want to use an Oracle ACFS feature that requires a compatibility increase. After the Oracle ACFS compatibility is updated, it is no longer possible to mount the file system with an older Oracle Grid infrastructure release. To change the compatibility of an existing file system, refer to “acfsutil compat set”. For information about Oracle ASM disk group compatibility attributes, see Oracle ACFS Features Enabled by Compatibility Attribute Settings.

The minimum file system size is 200 MB for the 512 byte metadata block size format and 512 MB for the 4 KB metadata block size format.

The Oracle ACFS driver must be loaded for `mkfs` to work.

4 KB Sector/Metadata

The following list summarizes the use of the `-o i` option and metadata block size based on the value of the `COMPATIBLE.ADVM` disk group attribute.

- If `COMPATIBLE.ADVM` is set to `12.2` or greater, then the metadata block size is `4096` bytes by default.
- If `COMPATIBLE.ADVM` is set to less than `12.2`, then the block size is set to 512 bytes.
- If `COMPATIBLE.ADVM` is set to less than `12.2` and the logical sector size is not 512 bytes, then the command fails.

User data IO continues to support transfers as small as 512 bytes for normal user IO requests. When the Oracle ADVM volume of the file system has a logical disk sector size of 4 K, user Direct IO requests should be aligned on 4 K file offsets and be a multiple of 4 KB lengths for the best performance. Note that only 4 K metadata block size formats support ADVM volumes with 4 K logical disk sectors.

The `COMPATIBLE.ADVM` value must be set to `12.2` or greater to format an Oracle ADVM volume whose logical disk size is 4096 or to use the `-o i=4096` option to format an Oracle ADVM volume whose logical disk sector size is 512 bytes.

Examples

Before creating an Oracle ACFS file system, first determine which Oracle ADVM volume devices are available. You can use the ASMCMD `volinfo` command to display information about the volumes and volume devices.

```
ASMCMD [+] > volinfo -a
...
         Volume Name: VOLUME1
         Volume Device: /dev/asm/volume1-123
         State: ENABLED
...
```

See Managing Oracle ADVM with ASMCMD .

Next create an Oracle ACFS file system on the volume device file.

Example 6-19   Using the mkfs command

```
$ /usr/sbin/mkfs -F acfs /dev/asm/volume1-123
```

mount

Purpose

Mounts an Oracle ACFS file system on the Solaris operating system.

Syntax and Description

```
mount -F acfs -o h /tmp /dev/null
mount -F acfs [-r] [-o options] volume_device mount_point
```

`mount -F acfs -o h /tmp /dev/null` displays usage text and exits.

Table 6-22 contains the options available with the `mount` command.

Table 6-22   Options for the Solaris mount command

Option	Description
`-F acfs`	Specifies the type of file system on Solaris. `acfs` designates the Oracle ACFS type.
`-r`	Mounts in read-only mode.

Table 6-22 (Cont.) Options for the Solaris mount command

Option	Description
`-o`	Specifies that options follow. Options are preceded with the `-o` flag followed by a comma-delimited string of options. For example: `-o all,v` The following options are available: • `all` Reads the Oracle ACFS managed file systems in the Oracle Clusterware namespace, mounting all file systems with the `AUTO_START=always`. The `-o all` option requires two placeholders to satisfy the command arguments: a dummy argument for the volume device and any valid directory. When the `-o all` option is specified, other `-o` options are ignored. To specify mount options for a registry entry, include those options with the `acfsutil registry` command when you add the entry to the registry. • `devices`/`nodevices` Allows or disallows the opening of any character or block special devices from this mount. • `exec`/`noexec` Allows or disallows the execution of programs in this file system. • `h` Displays the usage help text and exits. • `rootsuid`/`norootsuid` Allows or disallows the execution of `setuid` to root executables (binaries by non-root users whose permissions allow set user Id execution, and are owned by root). `rootsuid` is the default action. If `norootsuid` is specified, an attempt to run these executables as a non-root user fails with a permission denied error. • `ro` Mounts the file system in read-only mode. • `rw` Mounts the file system in read-write mode. • `setuid`/`nosetuid` Allows or disallows the execution of `setuid` and `setgid` programs. • `suid`/`nosuid` Allows or disallows the execution of `setuid` and `setgid` programs and the opening of any character or block special devices from this mount. • `v` Specifies verbose mode. The progress is displayed as the operation occurs.
`volume_device`	Specifies an Oracle ADVM volume device file that has been formatted by `mkfs`. *`device`* is required but can be a dummy value.

Table 6-22 (Cont.) Options for the Solaris mount command

Option	Description
mount_point	Specifies the directory where the file system is mounted. This directory must exist before you run the `mount` command.

`mount` attaches a file system to the Oracle ACFS hierarchy at the mount point that is the name of a directory. The mount occurs on the node where the mount command was issued. The mount command returns an error if the file system is not in a dismounted state on this node.

It is not always possible to return the cause of a mount failure to the `mount` command. When this happens Oracle ACFS writes the cause of the failure to the system console and associated system log file.

After `mount` runs successfully, the `MOUNTPATH` field in the `V$ASM_VOLUME` view displays the directory name on which the file system is now mounted.

An Oracle ACFS file system should only be mounted on one mount point. The same mount point name should be used on all cluster members.

`root` privilege is required to run `mount`.

Examples

The first example shows how to mount `volume1-123` on the mount point `/acfsmounts/acfs1`. The second example shows how to mount all the registered Oracle ACFS file systems. Placeholder arguments must be provided for the volume device name and mount point when specifying the `-o all` option. The volume device can be a dummy value, such as `none`. Any valid directory can be specified for the mount point, such as `/tmp`.

Example 6-20 Using the mount command

```
# /sbin/mount -F acfs /dev/asm/volume1-123 /acfsmounts/acfs1

# /sbin/mount -F acfs -o all none /tmp
```

umount/umountall

Purpose

Dismounts an Oracle ACFS file system on the Solaris operating system.

Syntax and Description

```
umount -V [mount_point | volume_device]
umountall -F acfs
```

`unmountall -F acfs` dismounts all Oracle ACFS file systems.

Table 6-23 contains the options available with the `umount` command.

Table 6-23 Options for the Solaris umount command

Option	Description
`-V`	Echoes the complete command line, but does not run the command. Use this option to verify and validate the command line before execution. Valid only with `umount`.
`mount_point`	Specifies the directory where the file system is mounted. Valid only with `umount`.
`volume_device`	Specifies the Oracle ADVM volume device name associated with the file system. Valid only with `umount`.
`-F acfs`	Specifies the type of file system on Solaris. `acfs` designates the Oracle ACFS type. Valid only with `umountall`.

`umount` and `umountall` detach an Oracle ACFS from the file system hierarchy on the current node.

The `umount` and `umountall` commands do not return control back to the operating system prompt until the file system has been completely unmounted.

If a file system is busy, `umount` and `umountall` fail.

`root` privileges are required to run the `umount` and `umountall` commands.

Examples

The following examples show how to dismount an Oracle ACFS file system. The first example specifies the mount point of the file system to dismount. The second example specifies the volume device associated with the file system to dismount. The third example dismounts all Oracle ACFS file systems.

Example 6-21 Using the umount command

```
# /sbin/umount /dev/asm/volume1-123

# /sbin/umount /acfsmounts/acfs1

# /sbin/umountall -F acfs
```

Oracle ACFS Command-Line Tools for the AIX Environment

This topic provides a summary of the Oracle ACFS commands for the AIX operating system.

During the installation of Oracle ACFS and Oracle ADVM with Oracle Grid Infrastructure on AIX, several system security classes (authorizations) are created for Oracle ACFS and Oracle ADVM operations. These authorizations enable the following operations:

- Configuration of Oracle ACFS and Oracle ADVM devices
- Removal of Oracle ACFS and Oracle ADVM devices
- Definition of Oracle ACFS and Oracle ADVM devices

These authorizations are collected in a role (`oracle_devmgmt`) which is assigned by default to the Oracle Grid Infrastructure user and Oracle ASM administrator. You can run the `lsrole` or `rolelist` command after installation to see this new role. These commands can be run at any time to ensure that the user has the proper authorization to manage Oracle ACFS and Oracle ADVM.

For example, run the following as the Oracle ASM administrator:

```
$ rolelist -e
 oracle_devmgmt
```

After the initial installation on an AIX operating system, the shell from which the root script was run does not inherit the new role. If the role is not present for the user, then the `swrole` command must be run before performing Oracle ACFS or Oracle ADVM operations.

For example, run the following as the Oracle ASM administrator:

```
$ swrole oracle_devmgmt
```

Alternatively, you can open a new shell to perform Oracle ACFS or Oracle ADVM operations.

During the removal of Oracle Grid Infrastructure, the `oracle_devmgmt` role and its associated authorizations are deleted from the system.

> **See Also:**
>
> Oracle Grid Infrastructure on AIX, refer to *Oracle Grid Infrastructure Installation Guide for IBM AIX on POWER Systems (64-Bit)* for more information about installing

> **Note:**
>
> If a disk group is dismounted with the force option on AIX, any Oracle ADVM volumes device files that were on that disk group remain on the system. These files are removed when the disk group is remounted.

Table 6-24 lists the Oracle ACFS AIX commands with brief descriptions. The commands in Table 6-24 have been extended with additional options to support Oracle ACFS on AIX.

Table 6-24 Summary of Oracle ACFS commands for AIX

Command	Description
fsck	Checks and repairs an Oracle ACFS file system on AIX.
mkfs	Creates an Oracle ACFS file system on AIX.
mount	Mounts an Oracle ACFS file system on AIX.
umount/umountall	Dismounts an Oracle ACFS file system on AIX.

fsck

Purpose

Checks and repairs an Oracle ACFS file system on the AIX operating system.

Syntax and Description

```
fsck -V acfs [-n|-y] [-o options] volume_device
```

Table 6-25 contains the options available with the AIX `fsck` command.

Table 6-25 Options for the AIX fsck command

Option	Description
`-V acfs`	Specifies an Oracle ADVM volume on AIX. `acfs` designates the Oracle ACFS type.
`-n`	Answers no to any prompts.
`-y`	Answers yes to any prompts.
`-o` *options*	Specifies that options follow (a, f, v). Options are preceded with the `-o` flag and entered as a comma-delimited string. For example: `-o a,v` • `a` Specifies to automatically fix the file system. • `f` Forces the file system into mountable state without completing a file system check or fix. • `v` Specifies verbose mode. The progress is displayed as the operation occurs. Running in verbose mode can impact performance.
volume_device	Specifies the primary Oracle ADVM volume device.

`fsck` checks and repairs an existing Oracle ACFS. This command can only be run on a dismounted file system. `root` privileges are required to run `fsck`. The Oracle ACFS driver must be loaded for `fsck` to work.

By default, `fsck` only checks for and reports any errors. The `-a` flag must be specified to instruct `fsck` to fix errors in the file system. Do not interrupt `fsck` during the repair operation.

In a few cases, `fsck` prompts for questions before proceeding to check a file system. These cases include:

- If `fsck` detects that another `fsck` is in progress on the file system
- If `fsck` detects that the Oracle ACFS driver is not loaded
- If the file system does not appear to be Oracle ACFS

In checking mode, `fsck` also prompts if there are transaction logs that have not been processed completely due to an incomplete shutdown. To run in a non-interactive mode, include either the `-y` or `-n` options to answer yes or no to any questions.

`fsck` creates working files before it checks a file system. These working files are created in `/usr/tmp` if space is available. `/tmp` is used if `/usr/tmp` does not exist. If insufficient space is available in the `tmp` directory, `fsck` attempts to write to the current working directory. The files that `fsck` creates are roughly the size of the file system being checked divided by 32K. At most three such files are allocated. For example, a 2 GB file system being checked causes `fsck` to generate one to three 64K working files in the `/usr/tmp` directory. These files are deleted after `fsck` has finished.

In the event that `fsck` finds a file or directory in the file system for which it cannot determine its name or intended location (possibly due to a corruption in its parent directory), it places this object in the `/lost+found` directory when `fsck` is run in fix mode. For security reasons only the `root` user on Linux can read files in `/lost+found`. If the administrator can later determine the original name and location of the file based on its contents, the file can be moved or copied into its intended location.

The file names in the `/lost+found` directory are in the following formats:

```
parent.id.file.id.time-in-sec-since-1970
parent.id.dir.id.time-in-sec-since-1970
```

The *id* fields are the internal Oracle ACFS numeric identifiers for each file and directory in the file system.

You can use `acfsutil info id` *id* *mount_point* to attempt to determine the directory associated with `parent.`*id*. This directory is assumed to be where the deleted object originated. For information about `acfsutil info`, see "acfsutil info file".

If the parent directory is not known, the parent `id` field is set to `UNKNOWN`.

> **Note:**
>
> It is not possible to see the contents of the `/lost+found` directory from a snapshot.

Examples

The following example shows how to check and repair an Oracle ACFS file system.

Example 6-22   Using the fsck command

```
# /usr/sbin/fsck -V acfs -y -o a /dev/asm/volume1-123
```

mkfs

Purpose

Creates an Oracle ACFS file system on the AIX operating system.

Syntax and Description

```
mkfs -V acfs -o h /dev/null
mkfs -V acfs [-v name ] [-s size][-o options] volume_device
       [-c release_version]
```

`mkfs -V acfs -o h /dev/null` displays usage text and exits.

Table 6-26 contains the options available with the AIX `mkfs` command.

Table 6-26 Options for the AIX mkfs command

Option	Description
`-V acfs`	Specifies the type of file system on AIX. `acfs` designates the Oracle ACFS type.
`-v` *`name`*	Specifies the name for the file system. A name can be a maximum of 64 characters. `acfsutil info fs` returns the name if one was specified.
`-s` *`size`*	Specifies the size of the file system in 512-byte units or in units of `K`|`M`|`G`|`T`|`P`. Units specified are in `K` (kilobytes), `M` (megabytes), `G` (gigabytes), `T` (terabytes), or `P` (petabytes).
`-o` *`options`*	Specifies that options follow (f, h, v). Options are preceded with the `-o` flag and entered as a comma-delimited string. For example: `-o f,v` • `f` Specifies the force option. This action creates the file system even if there is an existing Oracle ACFS on the volume device, although only if the file system is dismounted. This option overwrites structures on the original file system. Use this option with caution. • `h` Displays the usage text and exits. • `i=512 | i=4096` Specifies a file system with metadata block size of 512 or 4096 bytes. • `v` Specifies verbose mode. The progress is displayed as the operation occurs.
`volume_device`	Specifies an Oracle ADVM device file that is to be formatted.
`-c` *`release_version`*	Creates an Oracle ACFS file system with Oracle ACFS release compatibility set to the value of *release_version*.

You can use `mkfs` to create the on disk structure needed for Oracle ACFS file system to be mounted. The `mkfs` command is the traditional Linux command used to build a file system. After `mkfs` runs successfully, the `USAGE` column in the `V$ASM_VOLUME` view displays `ACFS`. `root` privilege is not required. The ownership of the volume device file dictates who can run this command.

The value specified with the `-c` *`release_version`* option must be greater than or equal to the value of `COMPATIBLE.ADVM` for the disk group, and must be less than or equal to the running Oracle Grid infrastructure release version. If `–c` *`release_version`* is not specified, the value of `COMPATIBLE.ADVM` is used. After the compatibility is set, it cannot be downgraded. The `–c` *`release_version`* option can be used in situations where it is not possible, or desirable, to update `COMPATIBLE.ADVM` and `COMPATIBLE.ASM` for the disk group, but you want to use an Oracle ACFS feature that requires a compatibility increase. After the Oracle ACFS compatibility is updated, it is no longer possible to mount the file system with an older Oracle Grid infrastructure release. To change the compatibility of an existing file system, refer to acfsutil compat set. For information about Oracle ASM disk group compatibility attributes, see Oracle ACFS Features Enabled by Compatibility Attribute Settings.

The minimum file system size is 200 MB for the 512 byte metadata block size format and 512 MB for the 4 KB metadata block size format.

The Oracle ACFS driver must be loaded for `mkfs` to work.

4 KB Sector/Metadata

The following list summarizes the use of the `-o i` option and metadata block size based on the value of the `COMPATIBLE.ADVM` disk group attribute.

- If `COMPATIBLE.ADVM` is set to `12.2` or greater, then the metadata block size is `4096` bytes by default.
- If `COMPATIBLE.ADVM` is set to less than `12.2`, then the block size is set to 512 bytes.
- If `COMPATIBLE.ADVM` is set to less than `12.2` and the logical sector size is not 512 bytes, then the command fails.

User data IO continues to support transfers as small as 512 bytes for normal user IO requests. When the Oracle ADVM volume of the file system has a logical disk sector size of 4 K, user Direct IO requests should be aligned on 4 K file offsets and be a multiple of 4 KB lengths for the best performance. Note that only 4 K metadata block size formats support ADVM volumes with 4 K logical disk sectors.

The `COMPATIBLE.ADVM` value must be set to `12.2` or higher to format an Oracle ADVM volume whose logical disk size is 4096 or to use the `-o i=4096` option to format an Oracle ADVM volume whose logical disk sector size is 512 bytes.

Examples

Before creating an Oracle ACFS file system, first determine which Oracle ADVM volume devices are available. You can use the ASMCMD `volinfo` command to display information about the volumes and volume devices.

```
ASMCMD [+] > volinfo -a
...
         Volume Name: VOLUME1
         Volume Device: /dev/asm/volume1-123
         State: ENABLED
...
```

For more information about `volinfo`, refer to Managing Oracle ADVM with ASMCMD.

Next create an Oracle ACFS file system on the volume device file.

Example 6-23 Using the mkfs command

```
$ /usr/sbin/mkfs -V acfs /dev/asm/volume1-123
```

mount

Purpose

Mounts an Oracle ACFS file system on the AIX operating system.

Syntax and Description

```
mount -V acfs -o h /dev/null /xxx
mount -v acfs [-o options] volume_device mount_point
```

`mount -V acfs -o h /dev/null /xxx` displays usage text and exits.

Table 6-27 contains the options available with the `mount` command.

Table 6-27 Options for the AIX mount command

Option	Description
`-v acfs`	Specifies the type of file system on AIX. `acfs` designates the Oracle ACFS type.
`-o` *options*	Specifies that options follow. Options are preceded with the `-o` flag followed by a comma-delimited string of options. For example: `-o all,ro,v` The following options are available: • `all` Reads the Oracle ACFS managed file systems in the Oracle Clusterware namespace, mounting all file systems with the `AUTO_START=always`. The `-o all` option requires two placeholders to satisfy the command arguments: a dummy argument for the volume device and directory. When the `-o all` option is specified, other `-o` options are ignored. To specify mount options for a registry entry, include those options with the `acfsutil registry` command when you add the entry to the registry. • `h` Displays the usage help text and exits. • `nosuid` Specifies that the execution of `setuid` and `setgid` programs by way of this mount is not allowed. • `nodev` Specifies that you cannot open devices from this mount. • `norootsuid` Fails the execution of binaries by non-root users whose permissions allow set user Id execution, and are owned by root. An attempt to run these executables as a non-root user fails with a permission denied error. • `rootsuid` Allows the execution of binaries by non-root users of set user Id files owned by root. This is the default action. • `ro` Mounts the file system in read-only mode. • `rw` Mounts the file system in read-write mode. • `v` Specifies verbose mode. The progress is displayed as the operation occurs.
volume_device	Specifies an Oracle ADVM volume device file that has been formatted by `mkfs`. *device* is required but can be a dummy value.
mount_point	Specifies the directory where the file system is mounted. This directory must exist before you run the `mount` command.

mount attaches a file system to the Oracle ACFS hierarchy at the mount point that is the name of a directory. The mount happens on the node where the mount command was issued. The mount command returns an error if the file system is not in a dismounted state on this node.

It is not always possible to return the cause of a mount failure to the mount command. When this happens Oracle ACFS writes the cause of the failure to the system console and associated system log file.

After mount runs successfully, the MOUNTPATH field in the V$ASM_VOLUME view displays the directory name on which the file system is now mounted.

An Oracle ACFS file system should only be mounted on one mount point. The same mount point name should be used on all cluster members.

root privilege is required to run mount.

Examples

The first example shows how to mount volume1-123 on the mount point /acfsmounts/acfs1. The second example shows how to mount all the registered Oracle ACFS file systems. The dummy names (none) have been entered as placeholders for the device and directory as these arguments are required, but not used, when the all option is specified.

Example 6-24 Using the mount command

```
# /usr/sbin/mount -v acfs /dev/asm/volume1-123 /acfsmounts/acfs1

# /usr/sbin/mount -v acfs -o all none none
```

umount/umountall

Purpose

Dismounts an Oracle ACFS file system on the AIX operating system.

Syntax and Description

```
umount [mount_point | volume_device]
umountall -F acfs
```

unmountall -F acfs dismounts all Oracle ACFS file systems.

Table 6-28 contains the options available with the umount command.

Table 6-28 Options for the AIX umount command

Option	Description
mount_point	Specifies the directory where the file system is mounted. Valid only with umount.
volume_device	Specifies the Oracle ADVM volume device name associated with the file system. Valid only with umount.
-F acfs	Specifies the type of file system on AIX. acfs designates the Oracle ACFS type. Valid only with umountall.

`umount` and `umountall` detach an Oracle ACFS from the file system hierarchy on the current node.

The `umount` and `umountall` commands do not return control back to the operating system prompt until the file system has been completely unmounted.

If a file system is busy, `umount` and `umountall` fail.

`root` privileges are required to run the `umount` and `umountall` commands.

Examples

The following examples show how to dismount an Oracle ACFS file system. The first example specifies the mount point of the file system to dismount. The second example specifies the volume device associated with the file system to dismount. The third example dismounts all Oracle ACFS file systems.

Example 6-25 Using the umount command

```
# /usr/sbin/umount /dev/asm/volume1-123

# /usr/sbin/umount /acfsmounts/acfs1

# /usr/sbin/umountall -F acfs
```

Oracle ACFS Command-Line Tools for Tagging

This topic provides a summary of the Oracle ACFS tagging commands.

Table 6-29 lists the Oracle ACFS tagging commands with brief descriptions.

On Solaris, `acfsutil tag` commands can set tag names on symbolic link files, but backup and restore utilities do not save the tag names that are explicitly set on the symbolic link files. Also, symbolic link files lose explicitly set tag names if they have been moved, copied, tarred, or paxed.

Table 6-29 Summary of commands for Oracle ACFS tagging

Command	Description
acfsutil tag info	Displays the tags for directories or files in Oracle ACFS file systems.
acfsutil tag set	Adds a tag to directories or files in an Oracle ACFS file system.
acfsutil tag unset	Removes a tag or all tag names from directories or files in an Oracle ACFS file system.

See Also:

- Oracle ACFS Tagging for an overview of Oracle ACFS tagging
- About Using Oracle ACFS Command-Line Tools for information about running Oracle ACFS `acfsutil` commands
- Oracle ACFS Tagging Generic Application Programming Interface for information about Oracle ACFS tagging application programming interfaces (APIs)

acfsutil tag info

Purpose

Displays the tag names for tagged directories or file names in Oracle ACFS file systems.

Syntax and Description

```
acfsutil tag info -h
acfsutil tag info [-r] [-c -t tagname] path [path ...]
acfsutil tag info [-c -t tagname]
```

`acfsutil tag info -h` displays help text and exits.

Table 6-30 contains the options available with the `acfsutil tag info` command.

Table 6-30 Options for the acfsutil tag info command

Option	Description
`-t` *`tagname`*	Specifies the tag name string to display. Enclose the string in quotes if the string contains a space.
`path`	Specifies the path name to one or more files or directories.
`-r`	Specifies recursive action on a specified path.
`-c`	Specifies case-insensitive partial matching on the tag name.

The `acfsutil tag info` command can recursively apply the operation to all child files and subdirectories if the `-r` option is included with specified directory path names.

If no path names are specified, the `acfsutil tag info` command searches all Oracle ACFS mount points for tagged files.

Only the paths of tagged files in an Oracle ACFS file system are displayed. If the `-t` *`tagname`* option is included, only paths of files with a tag name exactly matching the specified *`tagname`* are displayed. If the `-c` option is included, then *`tagname`* can be a case-insensitive substring of a tag name for a successful match. For example, the `acfsutil tag info -c -t AG` command would display path names of files with tag names such as `tag1`, `ag`, or `AG`.

Any user may use this command to display tag info on a directory to which the user has read access.

Examples

The following example show the use of the `acfsutil tag info` command.

Example 6-26 Using the acfsutil tag info command

```
$ /sbin/acfsutil tag info -r /acfsmounts/acfs1/myrepfiles/
```

acfsutil tag set

Purpose

Adds the given tag to the specified files or directories in an Oracle ACFS file system

Syntax and Description

```
acfsutil tag set -h
acfsutil tag set [-v] [-r] tagname path [path ...]
```

`acfsutil tag set -h` displays help text and exits.

Table 6-31 contains the options available with the `acfsutil tag set` command.

Table 6-31 Options for the acfsutil tag set command

Option	Description
tagname	Specifies a tag name string. Enclose the string in quotes if the string contains a space. The tag string can be composed of ASCII characters that include: a-b, A-Z, 0-9, the space character, the hyphen, and the underscore. The maximum length of a tag name is 32 characters. `all` is a reserved tag name and cannot be set on any file or directory.
path	Specifies the path string to one or more files or directories.
`-r`	Specifies recursive action on the specified path.
`-v`	Displays the progress of the operation.

The command can recursively apply the operation to all child files and subdirectories for the specified directory path names.

When adding a tag name to a file or directory, the existing tags on that file or directory remain unchanged. New files that are created after a directory is assigned a tag implicitly inherit tags from their parent directories. Existing files in the directory do not inherit the new tag; these files must be explicitly assigned the tag. Renaming a file or moving a file to a subdirectory within the same file system does not cause the file to inherit tag names from the new parent directory. Moving a file between file systems uses a copy operation and the newly created file does inherit the tag names of the parent directory.

There is no fixed maximum number of tag names for an Oracle ACFS file system or for each file and directory. However, the number of tag names is limited by the tag names that fit in up to 64 KB of special metadata storage for each file and directory. This metadata also contains information to manage the extended attributes. Longer tag names reduce the total of tag names that can be set on a file or directory. For example, if all tag names were 4 bytes long, then it is possible to hold approximately 1730 tag names, assuming no other extended attribute information is present.

Any user who has the privilege to modify the target file or directory may run this command.

Examples

The following example show the use of the `acfsutil tag set` command.

Example 6-27 Using the acfsutil tag set command

```
$ /sbin/acfsutil tag set repl_grp1 -r /acfsmounts/acfs1/myrepfiles/*.dat
```

acfsutil tag unset

Purpose

Removes the given tag name or all tag names from the specified file or directory.

Syntax and Description

```
acfsutil tag unset -h
acfsutil tag unset [-v] [-r] {all | tagname} path [path ...]
```

`acfsutil tag unset -h` displays help text and exits.

Table 6-32 contains the options available with the `acfsutil tag unset` command.

Table 6-32 Options for the acfsutil tag unset command

Option	Description
`all`	Specifies to remove all tags in the path string.
`tagname`	Specifies a tag name string. Enclose the string in quotes if the string contains a space.
`path`	Specifies the path string to one or more files or directories.
`-r`	Specifies recursive action on the specified path.
`-v`	Displays the progress of the operation.

The command can recursively apply the operation to all child files and subdirectories for the specified directory path names.

When removing a tag name from a file or directory, other existing tags on that file or directory remain unchanged.

Any user who has privilege to modify the target file or directory may run this command.

Examples

The following example show the use of the `acfsutil tag unset` command.

Example 6-28 Using the acfsutil tag unset command

```
$ /sbin/acfsutil tag unset repl_grp1 -r
     /acfsmounts/acfs1/myrepfiles/*.log
```

Oracle ACFS Command-Line Tools for Replication

This topic provides a summary of the commands for Oracle ACFS snapshot-based replication.

Table 6-33 lists the Oracle ACFS snapshot-based replication commands with brief descriptions. For an overview of Oracle ACFS replication, refer to Oracle ACFS Replication.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

Most `acfsutil repl` commands may be run either with system administrator privileges or as the replication user *repluser*. This is true for all `acfsutil repl` commands except for the following commands that read but never modify replication state:

- The `acfsutil repl info` and `acfsutil repl bg info` commands may be run by any Oracle ASM administrator user.
- The `acfsutil repl compare` command is allowed to be run by any Oracle ASM administrator user, but should be run as `root` to maximize its access to the files being compared.

> **Note:**
>
> Starting with Oracle ACFS 21c, Oracle ACFS replication protocol version 1 is desupported. Replication protocol version 1 has been replaced with snapshot-based replication version 2, introduced in Oracle ACFS 12c Release 2 (12.2.0.1).

Table 6-33 Summary of commands for Oracle ACFS replication

Command	Description
acfsutil repl bg	Starts, stops, or displays information about the Oracle ACFS replication background processes.
acfsutil repl compare	Verifies that files have been replicated on an Oracle ACFS storage location.
acfsutil repl failover	Converts the role of the replication standby location to that of a primary location in the event of an unplanned failure.
acfsutil repl info	Displays information about replication processing on an Oracle ACFS storage location.
acfsutil repl init	Initiates replication on files in an Oracle ACFS storage location.
acfsutil repl pause	Pauses replication on an Oracle ACFS storage location.
acfsutil repl resume	Resumes replication on an Oracle ACFS storage location.
acfsutil repl switchover	Switches the roles of the replication primary and standby storage location, assuming that both are currently available.
acfsutil repl sync	Synchronizes primary and standby storage locations.
acfsutil repl terminate	Stops all replication activity on the Oracle ACFS storage location.
acfsutil repl trace	Specifies the trace level for replication information.
acfsutil repl update	Updates replication information while replication is running.

acfsutil repl bg

Purpose

Starts, stops, or displays information about the Oracle ACFS replication background processes.

Syntax and Description

```
acfsutil repl bg -h
acfsutil repl bg {start | stop | info} {[snap_shot@]mount_point|

 -o replService}
```

`acfsutil repl bg -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl bg` command.

Table 6-34   Options for the acfsutil repl bg command

Option	Description
`start`	Starts the replication background processing.
`stop`	Stops the replication background processing
`info`	Displays information about the replication background process through the associated clusterware resource. If the resource is `ONLINE` on a node in the primary cluster, the process is running. Otherwise, the process is not running.
`-o replService`	Administer the replication service used by SSL-based replication, rather than the background process present on the primary cluster.
[*snap_shot@*]*mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary file system.

Oracle ACFS replication uses a background process on the primary cluster to control transporting file system changes to the standby location, either a file system or snapshot. This process must be running for replication to function. When replication is started with the `acfsutil repl init` command, the process is started and registered with Oracle Clusterware, enabling an automatic restart after a reboot or system crash. This process is clusterwide.

SSL-based Oracle ACFS replication also creates a background process on both the primary and standby clusters to process replication requests. This `ReplService` process must be running on both clusters in order for replication to function. When SSL-based replication is started, this process is started and registered with Oracle Clusterware, enabling automatic restart after a reboot or system crash. This process is clusterwide.

To determine if the Oracle ACFS replication background process or processes are running at the primary site, you can run the `acfsutil repl bg info` command on the primary cluster. For examples of the use of the command, refer to Example 6-29.

Note:

A primary file system containing a location participating in replication can always be unmounted, even if its replication background process is running. The `umount`command stops the background process if it is necessary to enable the unmount to proceed. Any in-progress replication operation is interrupted. When a primary file system containing a location participating in replication is remounted, replication automatically restarts and the background process is restarted.

A standby file system containing a location participating in replication can be unmounted only if a replication operation is not currently in progress on it. If a replication operation is in progress, then an attempt to unmount the file system fails with an indication that the file system is busy.

Examples

Example 6-29 shows the use of the `acfsutil repl bg` command to display information about replication processes.

Example 6-29 Using the acfsutil repl bg command

```
$ /sbin/acfsutil repl bg info /acfsmounts/acfs1
Resource:      ora.repl.dupd.crsdg1.crsdg1vol1.acfs
Target State:  ONLINE
Current State: ONLINE on primnode1
```

acfsutil repl compare

Purpose

Verifies that files have been replicated on an Oracle ACFS storage location.

Syntax and Description

```
acfsutil repl compare -h
acfsutil repl compare [-v] [ -a [-s] | -t { all | tagname, ...} [-s] ]
[snap_shot@]primary_mount_point [snap_shot@]standby_mount_point
```

`acfsutil repl compare -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl compare` command.

Table 6-35 Options for the acfsutil repl compare command

Option	Description
`-a`	Compares all files and their extended attribute names.
`-t` { `all` | *tagname*, `...` }	Compares files with all or some tag names specified in the `acfsutil repl init` operation. The `all` option specifies all tag names supplied in the `acfsutil repl init` operation. A comma-delimited list of tag names must only use tag names specified during the `acfsutil repl init` operation.

Table 6-35 (Cont.) Options for the acfsutil repl compare command

Option	Description
`-s`	Skips comparisons on symlinks for extended attributes and tags when using the `-a` or `-t` option. The symlinks are still checked for everything else, such as permissions and ownership.
`-v`	Causes the name of the file in each location to be output after the file is compared.
`[`*snap_shot@*`]`*primary_mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary file system.
`[`*snap_shot@*`]`*standby_mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted standby file system.

The `acfsutil repl compare` command verifies all or part of the primary location has been replicated to the standby location. The standby location must be mounted locally for comparison. Best results are obtained when the primary location is changing as little as possible. The command always tries to check all files on the primary location against those on the standby location. To maximize the ability of the command to check all files, it should be invoked as the root user. However, this is not required -- the command will try to compare files when invoked as any user.

The `-a` option is typically used when no tag names were specified during the `acfsutil repl init` operation. The `-a` option compares all files on the primary location against those on the standby location. The `-a` option also enables tests for extra files on the standby location that do not exist on the primary. When testing for extra files, the command also attempts to check extended attributes. However, if NFS is used to mount the standby location locally, the standby cannot be checked for matching extended attributes due to limitations in the NFS protocol.

The `-t` option is used when tags were specified during the `acfsutil repl init` operation. The `-t` operation locates all file names on the primary location with the specified tag names and compares them to the corresponding files on the standby. The `-t` option takes as arguments either `all` option or a comma-delimited list of tag names. If the `-t` option is used with the `all` argument, then all tag names supplied during the `acfsutil repl init` operation are selected. Otherwise, any specified tag names must exactly match a tag name supplied during the `acfsutil repl init` operation. The `-t` option also tests for extra files on the standby location that do not have an associated tag name that was specified during the `acfsutil repl init` operation. If NFS is used to mount the standby location locally, the standby cannot be checked for matching tag names and extended attributes due to limitations in the NFS protocol.

You can use the `acfsutil repl info -c` option to determine which tags were specified during the `acfsutil repl init` operation.

The `-a` and `-t` options cannot both be specified at that same time. When `-a` or `-t` is provided, the `–s` option can also be specified to skip extended attribute comparisons for symlinks. If neither the `-a` nor `-t` option is provided, a primary to standby file comparison is done without testing tag names nor extended attributes.

The `-v` option may be specified to output the name of each file after the file is compared.

Examples

Example 6-30 shows the use of the `acfsutil repl compare` command.

Example 6-30 Using the acfsutil repl compare command

```
$ /sbin/acfsutil repl compare /acfsmounts/acfs1 /nfs_mounted_standby
```

acfsutil repl failover

Purpose

Converts the role of the replication standby location to that of a primary location, and potentially the role of the corresponding primary location to that of a standby.

Syntax and Description

```
acfsutil repl failover -h
acfsutil repl failover [-T timeout] [snap_shot@]mount_point
```

`acfsutil repl failover -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl` failover command.

Table 6-36 Options for the acfsutil repl failover command

Option	Description
`-T` *`timeout`*	Specifies the time to wait (minutes) before determining that the primary location is unavailable. If the `-T` option is omitted, the primary location is checked once for availability.
[*`snap_shot@`*]*`mount_point`*	Specifies a location being replicated, either the mount point or a snapshot of a mounted file system.

The `acfsutil repl failover` command is issued at the standby replication site only. The command reverses the role of a replication standby location such that it becomes a replication primary location. The failover command insures that the standby location contains an exact copy of the results of the last successful replication transfer. If necessary, the command restores the location back to its state as of that transfer. `acfsutil repl failover` behaves differently based on the scenario in which it was run:

- Both the standby location and corresponding primary location are operating normally

  In this scenario, the command reverses the replication relationship. There is no data loss. Note that failover fails in this case if replication is paused. To enable this case to succeed, run `acfsutil repl resume`.

- The primary location is not currently available, but you want to wait until it is back online

  In this scenario, the command verifies the status of the replication primary. If the primary is not accessible and the timeout period has expired (if specified), then the command restores the standby location to its state as of the last successful replication transfer and converts it into a replication primary. Some data loss is possible, for example if there was a transfer in process when the primary location became unavailable. When the original primary location becomes available, it is aware that the failover command has been run and converts itself into a replication standby location.

- The primary location is not currently available and you do not want to wait until it is back online

  In this scenario, the command verifies the status of the replication primary. If the primary is not accessible and the timeout period has expired (if specified), then the command restores the standby location to its state as of the last successful replication transfer and converts it into a replication primary. Some data loss is possible, for instance if there was a transfer in process when the primary location became unavailable. After the failover command has been run, you have two options:

  – First, a new standby location can be configured using the `acfsutil repl update` command. Note that it is harmless if the original primary returns (as a standby) after you have specified the new standby. The original primary location remains idle (as a standby) until you run `acfsutil repl terminate standby` for that location.

  – Alternatively, you can terminate replication by running `acfsutil repl terminate primary` on the new primary.

When the current primary location is active, you should quiesce application updates to the primary before running `acfsutil repl failover`. Any updates attempted to the current primary location before its conversion to a standby are discarded. Any updates attempted to it after its conversion fail, just as updates would to any other standby location.

When application updates are resumed, they must be directed to the new primary location.

Examples

The following example shows the use of the `acfsutil repl failover` command. The command is invoked at the standby replication site and specifies the standby location (`/repl_data`).

Example 6-31 Using the acfsutil repl failover command

```
acfsutil repl failover /repl_data
```

> **See Also:**
>
> - acfsutil repl update for information about updating replication information
> - Configuring Oracle ACFS Snapshot-Based Replication for information about configuring `ssh` for use with Oracle ACFS replication

acfsutil repl info

Purpose

Displays information about replication processing on an Oracle ACFS storage location.

Syntax and Description

```
acfsutil repl info -h

acfsutil repl info -c [-v] {[snap_shot@]mount_point}

acfsutil repl info -c -u user [-o sshStrictKey=yn_value] standby_host [standby_host...]
                    [snap_shot@]primary_mount_point

acfsutil repl info -c -u user [-o sshStrictKey=yn_value] primary_host [primary_host...]
                    [snap_shot@]standby_mount_point

acfsutil repl info -s [-v][-l] [-n number{m|h|d|w|y}]
                   {-f eventlog | [snap_shot@]mount_point}}

acfsutil repl info [-a|-e|-t] [-v][-l]
                   {-f eventlog | [snap_shot@]mount_point }

acfsutil repl info {-o replService | -o sslPrintCredentials}

acfsutil repl info -o replServiceState [-s server]

acfsutil repl info -o sslPingReplService [-m message][-p port][-s server]
                   [-T timeout]
```

`acfsutil repl info -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl info` command.

Table 6-37 Options for the acfsutil repl info command

Option	Description
`-a`	Displays only apply records from the replication event log. An apply record contains the date and time that the set of replicated changes were captured on the primary location and the date and time that they were applied on the standby location.
`-c`	Displays configuration information.
`-e`	Displays only error records from the replication event log.
`-f` *eventlog*	Specifies the absolute path to the replication event log to be used as the source of the information. The default location is: *mount_point*`/.ACFS/repl/logs/ReplicationEventsLog`
`-h`	Displays help text.
`-l`	Displays last event only.
`-n` *number*{m|h|d|w|y}	Specifies the interval to display information. *number* specifies the number of minutes (`m`), hours (`h`), days (`d`), weeks (`w`), or years (`y`). For example: `-n 5d` The statistics are summarized in the units of time specified.
`-u` *user*	Validates the host and user keys currently configured between the current host and each standby host specified with a *standby* argument, using *user* as the user for which access is validated.
`-o sshStrictKey=`*yn_value*	Specifies whether `ssh` should use strict host-key checking. A value starting with `y` enables this checking, which is the default setting. A value starting with `n` disables the checking.

Table 6-37 (Cont.) Options for the acfsutil repl info command

Option	Description
`-s`	Displays statistical information. It is important that the primary and standby nodes are running a network time service to ensure that the statistics are meaningful.
`-t`	Displays only transport records from the replication event log.
`-v`	Displays verbose output.
`-o replService`	Displays the configuration information for the local replication service.
`-o replServiceState`	Requests state information from a replication service.
`-o sslPingReplService`	Reports the results of pinging a replication service.
`-o sslPrintCredentials`	Displays the contents of the credentials for the local site.
standby_host	Specifies a hostname or IP address for one of the standby hosts to be processed when the `-u` option is specified.
[*snap_shot@*]*primary_mount_point*	Specifies a location participating in replication, either a directory or a snapshot where the primary file system is mounted.
primary_host	Specifies a hostname or IP address for one of the primary hosts to be processed when the `-u` option is specified.
[*snap_shot@*]*standy_mount_point*	Specifies a location participating in replication, either a directory or a snapshot where the standby file system is mounted.
[*snap_shot@*]*mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary or standby file system.

To display information about replication configuration, use `acfsutil repl info` with the `-c` option. The configuration information includes a `Last sync time with primary` timestamp that identifies the point in time on the primary location that the standby location represents.

When specifying the `-u` option, the command should be run as `repluser`, the replication user. With this option, the command confirms that `ssh`, as used by snapshot-based replication, can connect from the local cluster to the remote cluster. The command should be used in two contexts:

- On each host of the primary cluster, to confirm that `ssh` can connect to all standby hosts and log in as user on each host
- On each host of the standby cluster, to confirm that `ssh` can connect to all primary hosts and log in as user on each host

When the command is run on a primary host, each *standby_host* string specifies a hostname or host IP address for a standby node. The specified [*snap_shot@*]*primary_mount_point* should be the active Oracle ACFS location in the primary cluster that is to be used for replication.

When the command is run on a standby host, each *primary_host* string specifies a hostname or host IP address for a primary node. The specified [*snap_shot@*]*standby_mount_point* should be the active Oracle ACFS location in the standby cluster that is to be used for replication.

In all cases, the user specified should be the same as the *repluser* specified in the `acsfutil repl init` command. If strict host-key checking was or is to be disabled for this instance of replication, for example with `acsfutil repl init primary` using `-o sshStrictKey=no`, then the option should also be specified here to skip host-key validation. `acfsutil repl info -c -u` can be run at any time, whether or not replication is currently active on the specified Oracle ACFS location.

To display information about replication statistics, use `acfsutil repl info` with the `-s` option.

To display information about all replication events, use `acfsutil repl info` with no options. To display information about a specific kind of event, use `acfsutil repl info` with the `-a`, `-e`, or `-t` option.

Examples

Example 6-32 shows the use of the `acfsutil repl info` command.

Example 6-32 Using the acfsutil repl info command

The validation command has the following format:

```
# /sbin/acfsutil repl info -c -u repluser standby-addr1 [standby-addr2 …]
my_primary_mountpoint
```

The command confirms that user `repluser` can use `ssh` to connect to each *standby-addr* given in the same way as replication does when initializing. The *my_primary_mountpoint* given should be the active Oracle ACFS mount point that is to be used for replication.

If you have two standby nodes named *standby1* and *standby2*, and also have a VIP named *standby12_vip* that can designate either node, then you can validate the key setup for the *standby1 /standby2* cluster by running the following command on each node of the primary cluster:

```
# /sbin/acfsutil repl info -c -u repluser standby1 standby2 my_primary_mountpoint
```

The same command would be used for validation using the VIP *standby12_vip* to connect to the cluster.

If you plan to disable strict host key checking, you can skip this checking by adding the `-o sshStrictKey=no` option to the command line.

Additional commands for displaying statistics and apply records are the following:

```
$ /sbin/acfsutil repl info -s -n 5d /acfsmounts/acfs1

$ /sbin/acfsutil repl info -a -v /acfsmounts/acfs1
```

acfsutil repl init

Purpose

Initiates replication on all the files in an Oracle ACFS storage location or only those files with a specified list of tags.

Syntax and Description

```
acfsutil repl init -h

acfsutil repl init standby -u repluser [-d trace_level] [-T {ssl | ssh}]
                           [{-o sslPermitCredentialSync | -o
sslProhibitCredentialSync}]
```

```
                            [-o sslCryptMethod=cryptMethod] [-o
sslMacMethod=macMethod] [-o SvcPort=port][snap_shot@]standby_mount_point

acfsutil repl init primary -s repluser@netname { -C | -M | -i interval} [-T {ssl
| ssh}] [tag_name...]
                            [-m [snap_shot@]standby_mount_point] [-d trace_level]
[-z {on | off}] [{-o sslPermitCredentialSync | -o sslProhibitCredentialSync}]
                            [-o sslCryptMethod=cryptMethod] [-o
sslMacMethod=macMethod] [-o SvcPort=port]
                            [-o sshCmdPath=pathname] [-o sshStrictKey=ynvalue]
                            [-o sshCipher=ciphername] [-p repluser@netname] [-o
sshOptions=options]
                            [-p [repluser@]netname]
[snap_shot@]primary_mount_point
```

`acfsutil repl init -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl init` command.

Table 6-38 Options for the acfsutil repl init command

Option	Description
`primary`	Specifies replication of the primary location.
`standby`	Specifies replication of the standby location.
`-s` *repluser@netname*	Specifies a standard `ssh` connection string for the primary site to contact the standby site. The *repluser* in the connection string is the user that `ssh` logs in as on the standby node. Choose a minimally-privileged user identity. The user should have Oracle ASM administrator privileges and should belong to the Oracle DBA group. **Note:** The same user and group identities (including all `uids` and `gids`) must be specified for the replication user on both your primary cluster and your standby cluster. The *netname* in the connection string specifies a network endpoint, such as a host name, VIP name, or IP address.
`—C`	Specifies replication in constant mode. Snapshots are continuously generated and replicated to the standby site. As soon as the replication of each snapshot completes, the generation of a new snapshot is started.

Table 6-38 (Cont.) Options for the acfsutil repl init command

Option	Description
`-M`	Specifies replication in manual mode. After the initial replication operation is performed, no further replication occurs until requested manually by running `acfsutil repl sync`.
`-i` *`interval`*	Specifies replication in interval (scheduled) mode. A new snapshot is taken and replicated with the frequency specified, if possible. A suffix must be given to specify the units in which interval is measured. The suffix must be either `s` (seconds), `m` (minutes), `h` (hours), `d` (days), or `w` (weeks). For example, `30m` is thirty minutes and `2h` is two hours.
`tag_name`	Specifies one or more tag names to be used in restricting replication. When tag names are specified, only files that are tagged with at least one of the those tag names are replicated. For information about Oracle ACFS tags, refer Oracle ACFS Tagging.
`-m [`*`snap_shot@`*`]`*`standby_mount_point`*	Specifies the standby storage location, if it is different than the primary location.
`-d` *`trace_level`*	Specifies the trace level setting [0..6].
`-z on|off`	Turns on or off compression of the replication data stream sent from primary to standby.
`-T {ssl | ssh}`	Specifies the transport to be used for this instance of replication, either SSL-enabled sockets (indicated by ssl) or ssh. In release 23ai, ssh is the default. When ssh is in use as the transport, the ssh-related options available in the repl init commands are accepted, and SSL-related options are ignored. When SSL is in use as the transport, the SSL-related options available in the repl init commands are accepted, and ssh-related options are ignored.
`-o sslCryptMethod=`*`cryptMethod`*	Specifies the cipher to be used to encrypt the data stream transferred to the standby. Values for cryptMethod may be none, **chacha20-poly1305** or **aesctr256**. The default is **aesctr256**. Note that the primary always determines the cipher to be used for transfers; on the standby, this option is ignored unless the standby becomes a primary.
`-o sslMacMethod=`*`macMethod`*	Specifies the message authentication code to be used to verify the data stream transferred to the standby. Values for *macMethod* may be **none** or **sha256.** If this option is not given, **sha256** will be used. Note that the primary always determines the MAC to be used for transfers; on the standby, this option is ignored unless the standby becomes a primary.

Table 6-38 (Cont.) Options for the acfsutil repl init command

Option	Description
`-o sslPermitCredentialSync`	For `acfsutil repl init standby`, specifies that clients of the replication service may request and receive the service's credentials. For `acfsutil repl init primary`, specifies that the primary will attempt to sync its credentials from the replication service on the standby. If this option is specified on both the standby and the primary, then credentials will be synced between the two with no further action needed. Once a credential sync has been completed, further syncs are disabled. This behavior is not enabled by default.
`-o sslProhibitCredentialSync`	For `acfsutil repl init standby`, specifies that clients of the replication service *may not* request and receive the service's credentials. For `acfsutil repl init primary`, specifies that the primary *will not* attempt to sync its credentials from the replication service on the standby. If this option is specified on either the standby or the primary, then sharing credentials between the two will require manual intervention. This option can be used to pre-empt credential syncing (i.e., undo a previous use of **`-o sslPermitCredentialSync`**) if no sync has yet occurred. This behavior is enabled by default.
`-o svcPort=`*`port`*	Specifies the port on which the replication service will accept requests. The default port number is 3043. Once the service has been configured with a given port, the port number may not be modified.
`-o sshCmdPath=`*`pathname`*	Specifies the pathname to the `ssh` command.
`-p` *`repluser@netname`*	Specifies the network endpoint for the primary location. netname names a network endpoint, such as a host name, VIP name, or IP address. netname is checked to ensure it can be accessed. *repluser* must match the *repluser* specified in the `-s` option. If `-p` is not given, then the hostname of the machine where this command was invoked is used.
`-o sshStrictKey=`*`ynvalue`*	Specifies whether `ssh` should use strict host-key checking. A value starting with `y` enables this checking, which is the default setting. A value starting with `n` disables the checking.
`-o sshCipher=`*`ciphername`*	Specifies the cipher that is passed to `ssh` to encrypt its sessions.

Table 6-38 (Cont.) Options for the acfsutil repl init command

Option	Description
`-o sshOptions=`*options*	Specifies options to be passed to each invocation of `ssh` made by replication. No restrictions are imposed on the contents of *options*. However, the resulting invocation of `ssh` is validated before replication is started, and `acfsutil repl init primary` fails if the validating invocation of `ssh` fails.
`-p [`*repluser*`@]`*netname*	Specifies the network endpoint for the primary location. *netname* names a network endpoint, such as a host name, VIP name, or IP address. *netname* is checked to ensure it can be accessed. *repluser*, if given, must match the *repluser* specified to `acfsutil repl init standby`. If `-p` is not given, then the SCAN VIP associated with the primary cluster, if available, is used as the network endpoint for the primary location. If the SCAN VIP is not available, then the hostname of the machine where this command was invoked is used.
`-u`	Specifies the *repluser*.
`[`*snap_shot*`@]`*standby_mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted standby file system.
`[`*snap_shot*`@]`*primary_mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary file system.

The `acfsutil repl init primary` and `standby` commands initiate replication between a specified primary storage location and a specified standby storage location. The `acfsutil repl init standby` command must be run first. Next, run the `acfsutil repl init primary` command which starts a background process at the primary site to control replication. When SSL is in use as the transport, a replication service process is started on both sites to process operations. The `acfsutil repl bg` commands may be used to administer any of these processes.

When the `acfsutil repl init primary` command completes successfully, processing begins to replicate initial copies of all specified files to the standby location. In addition, any changes to these files and any new files subsequently created in the primary storage location are captured and transported to the standby location. However, successful completion of the command does not indicate that the specified files have been copied in full to the standby location. To monitor the progress of the initial replication of all specified files to the standby location, users can run the `acfsutil repl info -c` command.

The initial replication operation is based on a snapshot of the primary location, which is transferred to the standby location in its entirety. Each subsequent replication operation uses a later snapshot of the primary location, and only the differences between that snapshot and the last previous snapshot are transferred to the standby location.

All directories in the primary location are always replicated to the standby, even if tags are specified.

Symbolic link files are replicated as is. If the symbolic link resolves to an absolute path name, and that path name does not exist on the standby location, then referencing the symbolic link results in errors.

On platforms that support named sockets, character device files or block device files, these file types are not replicated.

The file system containing a primary or standby storage location can be mounted on only one mount point in its cluster. The file system does not have to be mounted on all of the nodes in a cluster, but if it is mounted, it must be mounted on the mount point specified with the `acfsutil repl init` command. In addition, no other file system can be mounted on that mount point on other nodes.

A single Oracle ACFS location cannot be configured both as a primary and a standby location. If replication is configured on the same host or within the same cluster, then the primary and standby locations cannot use the same named mount point.

Replication is allowed between sites running any combination of the Linux, Solaris or AIX operating systems.

Examples

Example 6-33 shows the use of the `acfsutil repl init` to initiate snapshot-based replication from the primary to standby site.

Example 6-33 Using the acfsutil repl init command

```
# /sbin/acfsutil repl init standby -u my_repluser /my_standby/repl_data

# /sbin/acfsutil repl init primary -s my_repluser@my_host -i 2h
                                   -m /my_standby/repl_data /my_primary/repl_data
```

acfsutil repl pause

Purpose

Pauses replication on an Oracle ACFS primary location.

Syntax and Description

```
acfsutil repl pause -h
acfsutil repl pause [snap_shot@]mount_point
```

`acfsutil repl pause -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl pause` command.

Table 6-39 Options for the acfsutil repl pause command

Option	Description
[*snap_shot@*]*mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary file system.

The `acfsutil repl pause` command is issued at the primary replication site only. To continue replication operations, `acfsutil repl resume` should be run at a later time on the same primary site.

`acfsutil repl pause` stops the operation of the replication daemon temporarily. Subsequent running of `acfsutil repl resume` resumes the operation of the daemon.

If `acfsutil repl pause` is run while a replication operation is active, then the active replication operation completes regardless of the pause action. The pause operation is not effective in this case, and the command does not return, until the current data stream has been applied on the standby. After that, operations are not performed for this replication until the `acfsutil repl resume` is issued.

> **Note:**
>
> If replication is paused when using `acfsutil repl switchover`, or when using `acfsutil repl failover` with both the primary and the standby operating normally, then planned failover fails. To enable planned failover to succeed in these situations, run `acfsutil repl resume`.

Examples

Example 6-34 shows the use of the `acfsutil repl pause` command.

Example 6-34 Using the acfsutil repl pause command

```
$ /sbin/acfsutil repl pause /acfsmounts/acfs1
```

acfsutil repl resume

Purpose

Resumes replication on an Oracle ACFS location where replication has been paused.

Syntax and Description

```
acfsutil repl resume -h
acfsutil repl resume [snap_shot@]mount_point
```

`acfsutil repl resume -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl resume` command.

Table 6-40 Options for the acfsutil repl resume command

Option	Description
[*snap_shot@*]*mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary file system.

The `acfsutil repl resume` command resumes the replication process after the `acfsutil repl pause` command has been run. This command should be run on the same primary site soon after replication has been paused with the `acfsutil repl pause` command.

Examples

Example 6-35 shows the use of the `acfsutil repl resume` command.

Example 6-35 Using the acfsutil repl resume command

```
$ /sbin/acfsutil repl resume /acfsmounts/acfs1
```

acfsutil repl switchover

Purpose

Converts the role of the replication standby location to that of a primary location, and the role of the corresponding primary location to that of a standby.

Syntax and Description

```
acfsutil repl switchover -h
acfsutil repl switchover [-T timeout] [snap_shot@]mount_point
```

`acfsutil repl switchover -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl switchover` command.

Table 6-41 Options for the acfsutil repl switchover command

Option	Description
`-T` *`timeout`*	Specifies the time to wait (minutes) before determining that the primary location is unavailable. If the `-T` option is omitted, the primary location is checked once for availability.
[*snap_shot@*]*mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary file system.

The `acfsutil repl switchover` command is issued at the standby replication site only. The command reverses the role of a replication standby location such that it becomes a replication primary location and the role of a replication primary such that it becomes a standby. There is no data loss. The switchover command fails in the case where replication between the primary and standby locations is not operating normally. Such situations include when the standby site can't communicate with the primary site, when the primary location is not available, or if replication is paused. In these situations, the acfsutil repl failover command may be used to perform an unplanned failover. You should quiesce application updates to the primary before running `acfsutil repl switchover`. Any updates attempted to the primary location after its conversion fail, just as updates would to any other standby location. When application updates are resumed, they must be directed to the new primary location.

Examples

The following example shows the use of the `acfsutil repl switchover` command. The command is invoked at the standby replication site and specifies the standby storage location (`/repl_data`).

Example 6-36 Using the acfsutil repl switchover command

```
acfsutil repl switchover /repl_data
```

acfsutil repl sync

Purpose

Synchronizes primary and standby locations.

Syntax and Description

```
acfsutil repl sync -h
acfsutil repl sync [apply] [snap_shot@]mount_point
```

`acfsutil repl sync -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl sync` command.

Table 6-42 Options for the acfsutil repl sync command

Option	Description
`apply`	Specifies that all changes be applied to the standby location. Currently has no effect on the operation of the command.
[*snap_shot@*]*mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary file system.

The `acfsutil repl sync` command can be used to synchronize the state of the primary site and the standby site. This command can only be run on the primary site. Users should first quiesce their applications and issue a `sync(1)` call to ensure that the synchronized state is known and meaningful to the user. Running `acfsutil repl sync` then ensures all outstanding replication data is shipped from the primary site to the standby site. Specifically, the command enables any in-progress replication operation to complete, and then performs replication one final time to ensure that all changes on the primary have been replicated.

The command completes successfully when all of the changes have been successfully applied to the replication location on the standby site. At this point, unless an unmount of the last node on the primary site cluster is about to occur, applications may be restarted.

Successfully unmounting a replicated location on the primary site does not guarantee that all changes made prior to the unmount have been successfully sent to the standby site or applied to the standby location. If the primary location is on a file system that is unmounted on one primary site node, but remains mounted on one or more primary site nodes, changes to the location made before the unmount continue to be transported to the standby site from the other nodes after the unmount. However, if you are in the process of unmounting the primary location on the last primary site node where it is mounted, and if you want to know that all file system changes on the primary location up to that point have been successfully applied to the standby location, then you should perform the following operations to ensure that the standby location is up to date.

- Quiesce applications on the primary site which modify the replication location.
- Run `acfsutil repl sync`.

After `acfsutil repl sync` has returned successfully, you can unmount the file system which contains the participating replication location. Unmounting the file system stops the replication background process for that location.

Examples

Example 6-37 shows the use of the `acfsutil repl sync` command.

Example 6-37 Using the acfsutil repl sync command

```
$ /sbin/acfsutil repl sync /acfsmounts/acfs1
```

acfsutil repl terminate

Purpose

Stops all replication activity on the Oracle ACFS location at the site where it is run.

Syntax and Description

```
acfsutil repl terminate -h
acfsutil repl terminate primary [-d trace_level] [snap_shot@]mount_point
acfsutil repl terminate standby [remote [-f]] [immediate] [-k] [-d trace_level]
[snap_shot@]mount_point
```

`acfsutil repl terminate -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl terminate` command.

Table 6-43 Options for the acfsutil repl terminate command

Option	Description
`primary`	Stops replication on the primary location.
`standby`	Stops replication on the standby location.
`remote [-f]`	Stops replication processing on the primary location before stopping the processing on the standby location. When the remote keyword is given, the `-f` option forces the termination of replication on the standby location even if replication on the primary location cannot be terminated.
`immediate`	Stops replication processing immediately on the standby location.
`—k`	Specifies to keep any backup snapshot that is present for the standby. However, this option does not ensure that a backup snapshot exists, and the absence of such a backup snapshot is not an error. If this option is not specified, then any backup snapshot present is deleted.
`-d` *trace_level*	Specifies the trace level setting [0..6].
[*snap_shot@*]*mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted file system.

The `acfsutil repl terminate` command stops all replication processing. When terminating replication, you should terminate replication for the primary location first, and then the standby location. If you want to ensure all changes are sent to the standby location before terminating the primary site, ensure that all applications are quiesced and run the `acfsutil repl sync` command. The terminate command must be run on both sites if both sites hosting the locations are available.

After `acfsutil repl terminate standby` has been run, you can use the standby location in read-write mode. If you want to restart replication after running `acfsutil repl terminate`, then you must restart replication from the beginning with the `acfsutil repl init` command.

The `repl terminate standby` command waits until replication data at the standby location has been applied. If you want to terminate replication immediately without applying all the replication data, use the `immediate` option. However, this option can leave some files at the standby location in an indeterminate state.

To momentarily stop Oracle ACFS replication, you can use the `acfsutil repl pause` command, followed soon after by the `acfsutil repl resume` command.

> **Note:**
>
> Running `acfsutil repl terminate standby` with the `remote` option fails:
>
> - If the primary location is paused.
> - While the primary location is converting to a standby.
>
>   For example, if an `acfsutil repl failover` or `acfsutil repl switchover` operation is in progress involving the primary location.

Examples

Example 6-38 shows the use of the `acfsutil repl terminate` command.

Example 6-38 Using the acfsutil repl terminate command

```
$ /sbin/acfsutil repl terminate /acfsmounts/acfs1
```

acfsutil repl trace

Purpose

Sets the replication trace level for gathering trace information on an Oracle ACFS location.

Syntax and Description

```
acfsutil repl trace -h
acfsutil repl trace level [snap_shot@]mount_point
```

`acfsutil repl trace -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl trace` command.

Table 6-44 Options for the acfsutil repl trace command

Option	Description
level	Specifies the trace level setting [0..6]. The default level is 2.

Table 6-44 (Cont.) Options for the acfsutil repl trace command

Option	Description
[*snap_shot@*]*mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted file system.

Increasing the trace level can have a performance impact and should be done at the recommendation of Oracle support services.

Trace files for Oracle ACFS replication are stored in the following location:

```
GRID_BASE/diag/crs/hostname/crs/trace
```

In the location above, `GRID_BASE` specifies the `ORACLE_BASE` of the Oracle Grid Infrastructure home.

Examples

Example 6-39 shows the use of the `acfsutil repl trace` command.

Example 6-39 Using the acfsutil repl trace command

```
$ /sbin/acfsutil repl trace 5 /acfsmounts/acfs1
```

acfsutil repl update

Purpose

Updates the replication configuration after replication has started and is running on an Oracle ACFS location.

Syntax and Description

```
acfsutil repl update -h

acfsutil repl update  [-s netname] [-C | -M | -i interval]
                      [-d trace_level] [-z {on | off}] [-p netname]
                      [-o sslCryptMethod=cryptMethod] [-o
sslMacMethod=macMethod]
                      [-o sshCmdPath=pathname] [-o sshStrictKey=ynvalue]
                      [-o sshCipher=ciphername] [-o sshOptions=options] [-P]
                      [-m snap_shot@]standby_mount_point]
[snap_shot@]primary_mount_point

acfsutil repl update [-d trace_level] [snap_shot@]standby_mount_point

acfsutil repl update -T ssl
                     [-o sslPermitCredentialSync ]
                     [-o sslCryptMethod=cryptMethod] [-o sslMacMethod=macMethod]
[-o svcPort=port]
                     [snap_shot@]mount_point

acfsutil repl update [{-o sslPermitCredentialSync | -o
sslProhibitCredentialSync}]
                     [-o sslCryptMethod=cryptMethod] [-o sslMacMethod=macMethod]
                     [{-o sslExportCredentials=pathname | -o
sslImportCredentials=pathname }]
```

```
[-o sslReloadCredentials] [-o sslSyncCredentials]
[{-o sslCreateCredentials | -o sslDestroyCredentials }]
[-o replService -d trace_level] [-o svcPort=port]
```

`acfsutil repl update -h` displays help text and exits.

The following table contains the options available with the `acfsutil repl update` command.

Table 6-45 Options for the acfsutil repl update command

Option	Description
`-s` *netname*	Specifies a new network endpoint for the primary site to contact the standby site. *netname* specifies a network endpoint, such as a host name, VIP name, or IP address. *netname* is checked to ensure it can be accessed.
`–C`	Specifies replication in constant mode. Snapshots are continuously generated and replicated to the standby site. As soon as the replication of each snapshot completes, the generation of a new snapshot is started.
`–M`	Specifies replication in manual mode. After the next replication operation is performed, no further replication occurs until requested manually by running `acfsutil repl sync`.
`–i`*interval*	Specifies replication in interval (scheduled) mode. A new snapshot is taken and replicated with the frequency specified, if possible. A suffix must be given to specify the units in which interval is measured. The suffix must be either `s` (seconds), `m` (minutes), `h` (hours), `d` (days), or `w` (weeks). For example, `30m` is thirty minutes and `2h` is two hours.
`-d` *`trace_level`*	Specifies the trace level setting [0..6].
`-z on|off`	Turns on or off compression of the replication data stream sent from primary to standby.
`-T ssl`	Specifies that this instance of replication is to be updated to use SSL-enabled sockets as its transport. When ssh is in use as the transport, the ssh-related options available in the repl update command are accepted, and SSL-related options are ignored. When SSL is in use as the transport, the SSL-related options available in the repl update command are accepted, and ssh-related options are ignored.
`-o replService`	Used in conjunction with the **-d** option to set the trace level for the local replication service.
`-o sslCreateCredentials`	Creates credentials for the current site, returning an error if credentials already exist. Note that any remote sites or clients that interact with the current site will experience authentication failures until the new credentials are distributed to them. Once the new credentials are distributed to all affected sites, replication operations to and from those sites will resume.

Table 6-45 (Cont.) Options for the acfsutil repl update command

Option	Description
`-o sslCryptMethod=`*`cryptMethod`*	When SSL is in use, specifies the cipher to be used to encrypt the data stream transferred to the standby. Values for *cryptMethod* may be **none**, **chacha20-poly1305** or **aesctr256**. The default is **aesctr256**. Note that the primary always determines the cipher to be used for transfers; on the standby, this option is ignored unless the standby becomes a primary.
`-o sslDestroyCredentials`	Destroys replication credentials for the current site, producing an error if no credentials exist. Remote sites will experience authentication failures upon attempting to contact the site where credentials have been destroyed.
`-o sslExportCredentials=`*`pathname`*	Writes the local credentials to the file *pathname*, returning an error if the file already exists. This option is used for manual, maximally-secure credential distribution. The typical use case is for the user to export credentials at a site, securely copy them to a remote site, then import them at that remote site.
`-o sslImportCredentials=`*`pathname`*	Reads credentials from the file *pathname*, returning an error if the file does not exist. The file name should have been created by the **-o sslExportCredentials** option described just above.
`-o sslMacMethod=`*`macMethod`*	When SSL is in use, specifies the message authentication code to be used to verify the data stream transferred to the standby. Values for *macMethod* may be **none** or **sha256.** If this option is not given, **sha256** will be used. Note that the primary always determines the MAC to be used for transfers; on the standby, this option is ignored unless the standby becomes a primary.
`-o sslPermitCredentialSync`	On the standby, specifies that clients of the replication service may request and receive the service's credentials. On the primary, specifies that the primary will attempt to sync its credentials from the replication service on the standby. If this option is specified on both the standby and the primary, then credentials will be synced between the two with no further action needed. Once a sync has been completed, further syncs are disabled. This behavior is *not* enabled by default.
`-o sslProhibitCredentialSync`	On the standby, specifies that clients of the replication service *may not* request and receive the service's credentials. On the primary, specifies that the primary *will not* attempt to sync its credentials from the replication service on the standby. If this option is specified on either the standby or the primary, then sharing credentials between the two will require manual intervention. This option can be used to pre-empt credential syncing (i.e., undo a previous use of **`-o sslPermitCredentialSync`**) if no sync has yet occurred. This behavior is enabled by default.
`-o sslReloadCredentials`	Reloads the credentials of the local replication service from the local site's repository.

Table 6-45 (Cont.) Options for the acfsutil repl update command

Option	Description
`-o sslSyncCredentials`	Performs a "manual" credential sync, creating or updating the local credentials from those installed at the remote replication site. Both the local and remote sites must permit credential syncing in order for this operation to succeed.
`-o sshCmdPath=`*pathname*	Specifies the pathname to the `ssh` command.
`-o svcPort=`*port*	Specifies the port on which the replication service will accept requests. The default port number is 3043. Once the service has been configured with a given port, the port number may not be modified.
`-o sshStrictKey=`*ynvalue*	Specifies whether `ssh` should use strict host-key checking. A value starting with `y` enables this checking, which is the default setting. A value starting with `n` disables the checking.
`-o sshCipher=`*ciphername*	Specifies the cipher that is passed to `ssh` to encrypt its sessions.
`-o sshOptions=`*options*	Specifies options to be passed to each invocation of `ssh` made by replication. No restrictions are imposed on the contents of *options*. However, the resulting invocation of `ssh` is immediately validated, and `acfsutil repl update` fails if the validating invocation of `ssh` fails.
`-p` *netname*	Specifies a new network endpoint for the primary site. *netname* names a network endpoint, such as a host name, VIP name, or IP address. *netname* is checked to ensure it can be accessed.
`-P`	Removes any network endpoint previously specified with the `-p` option to `acfsutil repl update` or `acfsutil repl init primary`, and restores the default network endpoint. Specifically, the SCAN VIP associated with the primary cluster, if available, is used as the network endpoint for the primary location. If the SCAN VIP is not available, the hostname of the machine where this command was invoked is used.
[-m *snap_shot@*]*standby_mount_point*]	Specifies the standby storage area.
[*snap_shot@*]*primary_mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted primary file system.
[*snap_shot@*]*standby_mount_point*	Specifies a location being replicated, either the mount point or a snapshot of a mounted standby file system.

The `acfsutil repl update` command updates replication information and configuration after replication has been started. For example, a system administrator can change the network interface that is currently being used by replication to connect to the remote site. Issuing the `acfsutil repl update` command to update the configuration in this situation enables replication to continue running uninterrupted.

If `acfsutil repl update` is run on the primary site, then the command can be used to change the replication interval or tracing level, change the cipher used by or the pathname used for `ssh`, and turn compression on or off. The command can also be used to alter the network interface name (hostname or VIP name) that is used to connect to the standby site, or to update the standby storage location. If these updates, with the `-m` or `-s` option, designate a different storage location than was previously in use, then the previous standby storage location is left unchanged. The location is still marked as a replication standby, but can no longer be used for replication. The location is now simply a read-only file system or snapshot. To revert the previous standby to normal read-write status, run `acfsutil repl terminate standby` on it.

If `acfsutil repl update` is run on the standby site, the command can be used to alter the tracing level.

On either site, `acfsutil repl update` can be used to update an instance of replication to use SSL as its transport. The syntax of the command line used for this purpose is the fourth example shown above. On either site, when SSL is already in use as the transport, `acfsutil repl update` can also be used to administer credentials and update other SSL-related parameters. The syntax of the command line used for this purpose is the fifth example shown above. See Configuring SSL-Based Oracle ACFS Replication for more details on the use of the above options to configure the SSL-based transport.

At least one of the options must be specified when issuing this command; the command cannot be run with only the required location.

The `acfsutil repl update` command returns success when the updated information has been accepted on the local site. If the replication interval is changed with the `-C` or `-i` option, a replication operation occurs at the point when `acfsutil repl update` is run. The next replication operation occurs based on the newly-specified interval.

Examples

Example 6-40 shows the use of the `acfsutil repl update` command.

Example 6-40 Using the acfsutil repl update command

```
$ /sbin/acfsutil repl update -i 1h /my_primary/repl_data
```

Oracle ACFS Command-Line Tools for Encryption

This topic provides a summary of the commands for Oracle ACFS encryption.

Table 6-46 lists the Oracle ACFS encryption commands with brief descriptions. For an overview of Oracle ACFS encryption, refer to Oracle ACFS Encryption.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

> **Note:**
>
> Starting with Oracle ACFS 21c, Oracle ACFS encryption is desupported on Solaris and Microsoft Windows operating systems. Oracle ACFS Encryption on Oracle Solaris and Microsoft Windows is based on RSA technology. Retirement of RSA technology has been announced. Oracle ACFS Encryption continues to be supported on Linux, and is unaffected by this deprecation, because Linux uses an alternative technology.

Table 6-46 Summary of commands for Oracle ACFS encryption

Command	Description
acfsutil encr info	Displays encryption-related information about Oracle ACFS file systems.
acfsutil encr init	Initializes ACFS encryption. Creates encryption key store within OCR and sets up Oracle Key Vault as an alternative encryption key store.
acfsutil encr off	Disables encryption for an Oracle ACFS file system.
acfsutil encr on	Encrypts an Oracle ACFS file system.
acfsutil encr passwd	Changes password for password-protected PKCS wallets in OCR encryption key store, or changes password stored in autologin wallet for Oracle Key Vault endpoint.
acfsutil encr rekey	Generates a new key and re-encrypts an Oracle ACFS file system.
acfsutil encr set	Sets or changes encryption parameters for an Oracle ACFS file system.
acfsutil keystore migrate	Migrates OCR encryption key store between password-protected PKCS wallets and passwordless SSO wallets.

acfsutil encr info

Purpose

Displays encryption-related information about Oracle ACFS file systems, directories, or files.

Syntax and Description

```
acfsutil encr info -h
acfsutil encr info -m mount_point [[-r] path [path …]]
```

`acfsutil encr info -h` displays help text and exits.

Table 6-47 contains the options available with the `acfsutil encr info` command.

Table 6-47 Options for the acfsutil encr info command

Option	Description
`-m` *`mount_point`*	Specifies the directory where the file system is mounted.
`-r`	Specifies recursive action under an existing directory folder identified by *`path`*.
`path`	Specifies the absolute or relative path of a directory. Multiple path values are allowed.

If `-m` is specified without a *`path`*, the encryption status, algorithm, and key length are displayed for the file system level.

If `-r` is specified with a *`path`*, the encryption status, algorithm, and key length are displayed for all objects under the directory specified by *`path`*.

The `acfsutil encr info` command displays encryption status and parameters for files in a snapshot if the files are specified with the *`path`* option.

Any user can run this command to display encryption information about a file system, directory, or file.

If the `acfsutil encr info` command is run as a system administrator, then the output displays the types of key store used. The types are ACFS encryption key store with passwordless SSO wallets, ACFS encryption key store with password-protected PKCS wallets, and Oracle Key Vault as the key store (OKV).

If an OCI Vault Master Encryption Key is being used to encrypt the VEKs of the file system and the `acfsutil encr info` command is run as a system administrator, then the output displays the OCID of the OCI Vault Master Encryption Key and the cryptographic/management endpoint URLs for the OCI Vault Master Encryption Key.

Examples

The following are examples of the use of `acfsutil encr info`.

Example 6-41 Using the acfsutil encr info command

```
# /sbin/acfsutil encr info -m /acfsmounts/acfs1

# /sbin/acfsutil encr info -m /acfsmounts/acfs1
                           -r /acfsmounts/acfs1/myfiles
```

acfsutil encr init

Purpose

Initializes ACFS encryption. Creates encryption key store within OCR and sets up Oracle Key Vault as an alternative encryption key store.

Syntax and Description

```
acfsutil encr init -h
acfsutil encr init [-p ] [-o]
```

`acfsutil encr init -h` displays help text and exits.

Table 6-48 contains the options available with the `acfsutil encr init` command.

Table 6-48 Options for the acfsutil encr init command

Option	Description
`-p`	If not specified, create encryption key store within OCR using passwordless SSO wallets for key storage. If specified, create encryption key store within OCR using password-protected PKCS wallets for key storage.
`-o`	Create Oracle Key Vault autologin wallet to allow ACFS to autologin to Oracle Key Vault. This enables automounts and passwordless `acfsutil` encryption operations for ACFS file systems that use the Oracle Key Vault as the encryption key store.

The `acfsutil encr init` command must be run before any other `acfsutil` encryption commands can be run. This command must be run once for each cluster on which

Oracle ACFS encryption is run. This command must be run in either case of using the OCR as the encryption key store or Oracle Key Vault as the encryption key store.

If the `-p` option is not specified, an encryption key store will be created within the OCR using passwordless SSO wallets. If the `-p` option is specified, an encryption key store will be created within the OCR using password-protected PKCS wallets. You must provide a password for the PKCS wallets when prompted. The password must conform to the format:

- The maximum number of characters is 20.
- The minimum number of characters is 8.
- The password must contain at least one digit.
- The password must contain at least one letter.

If OCI Vault Master Encryption Keys will be used to encrypt Volume Encryption Keys (VEKs) for any ACFS file system, the encrypted VEKs will need to be stored in the OCR with passwordless SSO wallets. The `-p` option must not be specified so that the encryption key store within the OCR will be created with passwordless SSO wallets.

If the `-o` option is specified, then in addition to creating an encryption key store within the OCR, an autologin wallet will also be created for the Oracle Key Vault. This Oracle Key Vault autologin wallet will enable ACFS to autologin to the Oracle Key Vault.

The Oracle Key Vault autologin wallet enables the following functionality:

- Creating the Oracle Key Vault autologin wallet enables all forms of automount (e.g. CRS automount, OS-level automount, etc) to correctly mount ACFS file systems that use the Oracle Key Vault as the encryption key store. Without the Oracle Key Vault autologin wallet, those ACFS file systems will not be correctly mounted by automounts, causing the encrypted files within the ACFS file system to be inaccessible.
- Creating the Oracle Key Vault autologin wallet enables `acfsutil` encryption operations to be performed passwordless for ACFS file systems that use the Oracle Key Vault as the encryption key store.

When the `-o` option is used to create the Oracle Key Vault autologin wallet, note that the `ORACLE_BASE`, `ORACLE_HOME`, `ORACLE_SID`, and `OKV_HOME` environment variables must be set appropriately for a login to the Oracle Key Vault endpoint. If ACFS is to be shared by multiple nodes, then each node needs to be registered as a separate endpoint. If ACFS is to be shared by multiple nodes, then each node needs to be registered as a separate endpoint. An Oracle key vault wallet should be created and configured as the default wallet for every endpoint. This allows all the nodes to share the same set of volume encryption keys stored on Oracle key vault. Additionally, if the Oracle Key Vault endpoint requires a password for login, the `-o` option will prompt for the Oracle Key Vault endpoint password. The Oracle Key Vault endpoint password will be saved within the autologin wallet to enable ACFS to autologin to the Oracle Key Vault.

Note that all Oracle Key Vault endpoints within the same cluster must share the same endpoint password to allow ACFS to autologin to the Oracle Kev Vault from all nodes. If any Oracle Key Vault endpoint has a different endpoint password from the password stored in the Oracle Key Vault autologin wallet, ACFS will be unable to autologin to the Oracle Key Vault through that endpoint.

If the `-o` option is specified and an encryption key store within the OCR already exists, then the creation of the encryption key store within the OCR will be skipped and only the creation of the Oracle Key Vault autologin wallet will be performed.

If both the -p and -o options are specified, note that two passwords may be requested, one for the PKCS wallets in the OCR encryption key store, one for the Oracle Key Vault endpoint.

Only a user with root or system administrator privileges can run this command.

Examples

The following is an example of the use of `acfsutil encr init`.

Example 6-42 Using the acfsutil encr init command

```
# /sbin/acfsutil encr init
```

acfsutil encr off

Purpose

Disables encryption for an Oracle ACFS file system, directories, or individual files.

Syntax and Description

```
acfsutil encr off -h
acfsutil encr off -m mount_point [[-r] path [ path ...]]
```

`acfsutil encr off -h` displays help text and exits.

Table 6-50 contains the options available with the `acfsutil encr off` command.

Table 6-49 Options for the acfsutil encr off command

Option	Description
`-m` *mount_point*	Specifies the directory where the file system is mounted.
`-r`	Specifies to disable encryption recursively under an existing directory identified by *path*.
path	Specifies the absolute or relative path of a directory. Multiple path values are allowed.

Only an administrator can run this command on an Oracle ACFS file system (`-m` option without a *path* specified). When the `-m` option is specified without a *path*, all the files under the mount point are decrypted.

The *path* option can specify a path to a file or directory in a read-write snapshot. If the `-r` option is specified with the command on the root directory, the command does not transverse the snapshots under the `.ACFS` directory. If a decryption operation is specified at the file system level, then the operation does not process files and directories of snapshots in the `.ACFS/snaps/` directory.

Only a user with root or system administrator privileges can run this command to disable encryption on a file system. The file owner can also run this command to disable encryption on a directory or file.

Examples

The following are examples of the use of `acfsutil encr off`.

Example 6-43 Using the acfsutil encr off command

```
# /sbin/acfsutil encr off -m /acfsmounts/acfs1

# /sbin/acfsutil encr off -m /acfsmounts/acfs1
                          -r /acfsmounts/acfs1/myfiles
```

acfsutil encr on

Purpose

Encrypts an Oracle ACFS file system, directories, or individual files.

Syntax and Description

```
acfsutil encr on -h
acfsutil encr on -m mount_point
          [-a {AES} -k {128|192|256}] [[-r] path [path...]]
```

`acfsutil encr on -h` displays help text and exits.

Table 6-50 contains the options available with the `acfsutil encr on` command.

Table 6-50 Options for the acfsutil encr on command

Option	Description
`-m` *mount_point*	Specifies the directory where the file system is mounted.
`-a` *algorithm*	Specifies the encryption algorithm type for a directory or file. Advanced Encryption Standard (AES) is the only encryption algorithm supported for this release.
`-k` *key_length*	Specifies the encryption key length for a directory or file.
`-r`	Specifies encryption recursively under existing directory folder identified by *path*.
path	Specifies the absolute or relative path of a directory. Multiple path values are allowed.

The default values for the `-a` and `-k` are determined by the volume parameters specified when `acfsutil encr set` was run. To set the key length at the volume level, use the `acfsutil encr set` command.

Only an administrator can run this command on an Oracle ACFS file system (`-m` option without a *path* specified). When the `-m` option is specified without a *path*, all the files under the mount point are encrypted.

The *path* option can specify a path to a file or directory in a read-write snapshot. If the `-r` option is specified with the command on the root directory, the command does not transverse the snapshots under the `.ACFS` directory. If an encryption operation is specified at the file system level, then the operation does not process files and directories of snapshots in the `.ACFS/snaps/` directory.

When you run `acfsutil encr on` with the `-r` option, the command encrypts the specified directory recursively, but does not enable encryption on the file system level.

Only a user with root or system administrator privileges can run this command to enable encryption on a file system. The file owner can also run this command to enable encryption on a directory or file.

Examples

The following are examples of the use of `acfsutil encr on`.

Example 6-44   Using the acfsutil encr on command

```
# /sbin/acfsutil encr on -m /acfsmounts/acfs1

# /sbin/acfsutil encr on -m /acfsmounts/acfs1
                         -a AES -k 128 -r /acfsmounts/acfs1/myfiles
```

acfsutil encr passwd

Purpose

Changes password for password-protected PKCS wallets in OCR encryption key store, or changes password stored in autologin wallet for Oracle Key Vault endpoint.

Syntax and Description

```
acfsutil encr passwd -h
acfsutil encr passwd [-o]
```

`acfsutil encr passwd -h` displays help text and exits.

Options for the acfsutil encr passwd contains the options available with the `acfsutil encr passwd` command.

Table 6-51   Options for the acfsutil encr passwd command

Option	Description
`-o`	If not specified, change password for password-protected PKCS wallets in OCR encryption key store. If specified, change password stored in Oracle Key Vault autologin wallet for the Oracle Key Vault endpoint.

If the -o option is not specified, this command changes the password for the password-protected PKCS wallets in the OCR encryption key store. The command must be run on an OCR encryption key store that uses password-protected PKCS wallets for key storage. The command cannot be run on an OCR encryption key store that uses passwordless SSO wallets for key storage.

The command will prompt for the existing password of the password-protected PKCS wallets, then prompt for a new password. The new password must conform to the format:

- The maximum number of characters is 20.
- The minimum number of characters is 8.
- The password must contain at least one digit.
- The password must contain at least one letter.

If the -o option is specified, this command changes the password stored in the Oracle Key Vault autologin wallet for the Oracle Key Vault endpoint. The command will prompt for the existing password of the Oracle Key Vault endpoint, then prompt for the new password. The command will verify that the new password can correctly login to the Oracle Key Vault endpoint, then store the new password in the Oracle Key Vault autologin wallet.

Only a user with root or system administrator privileges can run this command.

Examples

The following is an example of the use of `acfsutil encr passwd` command.

Example 6-45 Using the acfsutil encr passwd command

```
# /sbin/acfsutil encr passwd
```

acfsutil encr rekey

Purpose

Generates a new key and re-encrypts volume or file.

Syntax and Description

```
acfsutil encr rekey -h
acfsutil encr rekey -m mount_point
    {-f [-r] path [path…] |-v } [-a {AES} -k {128|192 |256}]
```

`acfsutil encr rekey -h` displays help text and exits.

Table 6-52 contains the options available with the `acfsutil encr rekey` command.

Table 6-52 Options for the acfsutil encr rekey command

Option	Description
-m *mount_point*	Specifies the directory where the file system is mounted.
-f [-r] *path* ...	Generates a new file encryption key for the specified path and then encrypts the data with the new key. If -r is specified, the rekey operation is performed recursively under *path*. *path* specifies the absolute or relative path of a directory. Multiple path values are allowed.
-v	Generates a new volume encryption key (VEK) for the specified mount point and then encrypts all the file encryption keys in file system with the new key. Prompts for the wallet password because the wallet must be accessed to store the new VEK. The generated key is stored in the key store that was previously configured with the `acfsutil encr init` command.
-a *algorithm*	Specifies the algorithm. Advanced Encryption Standard (AES) is the only encryption supported for this release.
-k *key_length*	Specifies the key length for the directory or file specified by *path*.

The default values for the -a and -k are determined by the volume parameters specified when `acfsutil encr set` was run.

The *path* option can specify a path to a file or directory in a read-write snapshot. If the `-r` option is specified with the command on the root directory, the command does not transverse the snapshots under the `.ACFS` directory. If a rekey operation is specified at the file system level, then the operation does not process files and directories of snapshots in the `.ACFS/snaps/` directory.

If Oracle Key Vault is the key store for the file system, then the Oracle Key Vault home environmental variable (`OKV_HOME`) must be set when using the `-v` option to generate a new volume key. If ACFS is to be shared by multiple nodes, then each node needs to be registered as a separate endpoint. An Oracle key vault wallet should be created and configured as the default wallet for every endpoint. This allows all the nodes to share the same set of volume encryption keys stored on Oracle key vault. If the client was configured to use a password with Oracle Key Vault, then the same password must be entered when prompted.

See Also:

Oracle Key Vault Administrator's Guide for information about Oracle Key Vault

If existing VEKs are encrypted by an OCI Vault Master Encryption Key, then when the -v option generates a new VEK, the new VEK will also be encrypted by the OCI Vault Master Encryption Key. The new VEK will automatically be encrypted by the latest version of the OCI Vault Master Encryption Key.

Note that the `-v` option will only rotate the VEK for the ACFS file system, it will not rotate the OCI Vault Master Encryption Key. The OCI Vault Master Encryption Key can be rotated independently via the OCI Console. When the OCI Vault Master Encryption Key is rotated, existing VEKs will continue to be encrypted with older versions of the OCI Vault Master Encryption Key. Only new VEKs generated with the `-v` option will be encrypted with the new version of the OCI Vault Master Encryption Key. Although the VEK and the OCI Vault Master Encryption Key can be rotated independently, it is recommended that both be rotated at the same time. The OCI Vault Master Encryption Key should be rotated before the VEK is rotated so that the new VEK will be encrypted with the new version of the OCI Vault Master Encryption Key.

Only a user with root or system administrator privileges can run this command with the `-v` option. The file owner can also run this command with the `-f` option to rekey encryption on the directory or file.

Examples

The following are examples of the use of `acfsutil encr rekey`.

Example 6-46 Using the acfsutil encr rekey command

```
# /sbin/acfsutil encr rekey -m /acfsmounts/acfs1 -v

# /sbin/acfsutil encr rekey -m /acfsmounts/acfs1 -f 
                            -r /acfsmounts/acfs1/myfiles
```

acfsutil encr set

Purpose

Sets or changes encryption parameters for an Oracle ACFS file system.

Syntax and Description

```
acfsutil encr set -h
acfsutil encr set [-a {AES} -k {128|192|256}] -m mount_point
acfsutil encr set [-a {AES} -k {128|192|256}] -e -m mount_point
acfsutil encr set [-a {AES} -k {128|192|256}] -K master_encryption_key_ocid -C
cryptographic_endpoint
-M management_endpoint -m mount_point
acfsutil encr set -u -m mount_point
```

`acfsutil encr set -h` displays help text and exits.

Table 6-53 contains the options available with the `acfsutil encr set` command.

Table 6-53 Options for the acfsutil encr set command

Option	Description
`-a` *algorithm*	Specifies the algorithm. Advanced Encryption Standard (`AES`) is the default value and the only encryption supported for this release. The algorithm must be specified if `-k` is specified.
`-k` {`128`|`192`|`256`}	Specifies the key length. The key length is set at the volume level. The default is `256`. Must be specified if `-a` is specified.
`-e`	Specifies to use Oracle Key Vault as the key store.
`-K` *master_encryption_key_ocid*	Specifies the OCID of the OCI Vault Master Encryption Key to use to encrypt VEKs for the Oracle ACFS file system.
`-C` *cryptographic_endpoint*	Specifies the URL of the cryptographic endpoint for the OCI Vault Master Encryption Key
`-M` *management_endpoint*	Specifies the URL of the management endpoint for the OCI Vault Master Encryption Key.
`-u`	Backs out encryption. Decrypts all encrypted files in the file system and reverts the file system to the state before `acfsutil encr set` was run on the file system.
`-m` *mount_point*	Specifies the directory where the file system is mounted.

Before running the `acfsutil encr set` command, you must first run the `acfsutil encr init` command.

The `acfsutil encr set` command configures encryption parameters for a file system, transparently generates a Volume Encryption Key (VEK), and stores that generated key in the encryption key store that was previously configured with the `acfsutil encr init` command.

In addition `acfsutil encr set` creates the *mount_point*`/.Security/encryption/logs/` directory that contains the log file (`encr-`*hostname*`_fsid.log`) that collects auditing and diagnostic data.

Password requirements when storing the key are dependent on how the encryption key storage was configured. If `-p` was specified with `acfsutil encr init`, then a password is required to run this command.

Before using the `-e` option to specify Oracle Key Vault as the key store, Oracle Key Vault must be configured first. If you want to choose Oracle Key Vault as the key store for the file system, then the Oracle Key Vault home environmental variable (`OKV_HOME`) must be set when running the command with the `-e` option. If ACFS is to be shared by multiple nodes, then each node needs to be registered as a separate endpoint. An Oracle key vault wallet should be created and configured as the default wallet for every endpoint. This allows all the nodes to share the same set of volume encryption keys stored on Oracle key vault. If the client was configured to use a password with Oracle Key Vault, then the same password must be entered when prompted.

If an OCI Vault Master Encryption Key is specified using `-K/-C/-M`, the OCI Vault Master Encryption Key will be used to encrypt VEKs for the file system. The VEK generated by acfsutil encr set will be encrypted by the OCI Vault Master Encryption Key before being stored in the encryption key store that was previously configured with the `acfsutil encr init` command. The specified information for the OCI Vault Master Encryption Key will also be stored in the encryption key store. All future VEKs generated by `acfstuil encr rekey` will be transparently encrypted by the OCI Vault Master Encryption Key before being stored in the encryption key store.

Some requirements must be met before using an OCI Vault Master Encryption Key to encrypt VEKs for an ACFS file system:

- Each ACFS file system should have its own OCI Vault Master Encryption Key. See Creating a Master Encryption Key
- The OCI instance with the ACFS file system must be granted access to the OCI Vault Master Encryption Key. See Granting OCI Instances access to an OCI Vault Master Encryption Key

> **See Also:**
>
> *Oracle Key Vault Administrator's Guide* for information about configuring Oracle Key Vault

The `acfsutil encr set -u` command is not allowed if any snapshots exist in the file system.

Only a user with root or system administrator privileges can run the `acfsutil encr set` command.

Examples

The following example shows the use of `acfsutil encr set` command.

Example 6-47 Using the acfsutil encr set command

```
# /sbin/acfsutil encr set -a AES -k 256 -m /acfsmounts/acfs1
```

acfsutil keystore migrate

Purpose

Migrates the ACFS encryption key store within the OCR between PKCS and SSO wallets. Can also migrate the encryption keys for a specific ACFS file system from the OCR to OKV.

Syntax and Description

```
acfsutil keystore migrate -h
acfsutil keystore migrate [-p | -o <mountpoint>]
```

`acfsutil keystore migrate -h` displays help text and exits.

Table 6-48 contains the options available with the `acfsutil keystore migrate` command.

Table 6-54 Options for the acfsutil keystore migrate command

Option	Description
`-p`	Converts the ACFS encryption key store within the OCR from passwordless SSO wallets to password-protected PKCS wallets for key storage.
`-o` *`<acfs_mountpoint>`*	Migrates the ACFS encryption keys on the specified ACFS mountpoint from the Oracle Cluster Registry (OCR) to the Oracle Key Vault (OKV).

If the `-p` option is specified, `acfsutil keystore migrate` converts the ACFS encryption key store within the OCR from passwordless SSO wallets to password-protected PKCS wallets for VEK storage. If the `-p` option is not specified, `acfsutil keystore migrate` converts the ACFS encryption key store within the OCR from password-protected PKCS wallets to passwordless SSO wallets for VEK storage. If the `-p` option is specified, you must provide a password when prompted. The password must conform to the format:

- The maximum number of characters is 20.
- The minimum number of characters is 8.
- The password must contain at least one digit.
- The password must contain at least one letter.

Only a user with root or system administrator privileges can run this command.

If the `-o` *`<acfs_mountpoint>`* option is specified, the command can be used to migrate the VEKs for the specified ACFS file system encryption keys from the OCR to the OKV. Once this command is finished running, all the encryption keys stored in the OCR will be migrated to the OKV, and the OCR will no longer contain these keys. If other mountpoints exist, their encryption keys remain unchanged. In the case of command failure, the encryption setting before running the command is preserved. Rerunning this command after the issues are resolved could complete the migration.

> **Note:**
>
> - Once VEKs for an ACFS file system are migrated from the OCR to the OKV, there is no way to reverse the operation.
> - This command is available for file system with compatibility set as 23.0.0.0.0 or the above, and ACFS software versions set as 23.0.0.0.0 or the above.

If there are any ACFS file systems using OCI Vault Master Encryption Keys to encrypt VEKs, the ACFS encryption key store in the OCR cannot be migrated from passwordless SSO wallets to password-protected PKCS wallets. `acfsutil keystore migrate` will prevent this migration.

If an ACFS file system is using an OCI Vault Master Encryption Key to encrypt VEKs, if the VEKs are migrated from the OCR to OKV using the `-o` option, the VEKs will no longer be encrypted by the OCI Vault Master Encryption Key once stored in OKV.

Examples

The following are examples of the use of `acfsutil keystore migrate`.

Example 6-48 Using the `acfsutil keystore migrate` command

```
# /sbin/acfsutil keystore migrate
```

The following is an example of the use of `acfsutil keystore migrate` with the `-o` *`<acfs_mountpoint>`* option.

Example 6-49 Using the `acfsutil keystore migrate -o <acfs_mountpoint>` command

```
# /sbin/acfsutil keystore migrate -o /my_acfs_mnt
```

Oracle ACFS Command-Line Tools for Snapshots

This topic provides a summary of commands to manage Oracle ACFS snapshots.

Table 6-55 lists Oracle ACFS snapshot commands with brief descriptions. For an overview of Oracle ACFS snapshots, refer to About Oracle ACFS Snapshots.

For more information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

Table 6-55 Summary of commands for Oracle ACFS snapshots

Command	Description
acfsutil snap convert	Converts the type of an existing snapshot.
acfsutil snap create	Creates a snapshot of an Oracle ACFS file system or an existing snapshot.
acfsutil snap delete	Deletes a snapshot of an Oracle ACFS file system.
acfsutil snap duplicate apply	Updates a duplicate snapshot or file system.

Table 6-55 (Cont.) Summary of commands for Oracle ACFS snapshots

Command	Description
acfsutil snap duplicate create	Creates a duplicate of an existing snapshot.
acfsutil snap info	Displays information about Oracle ACFS file system snapshots.
acfsutil snap link	Creates or deletes a snapshot link.
acfsutil snap quota	Sets the quota limit for a specified snapshot.
acfsutil snap remaster	Remasters a file system with a specified snapshot.
acfsutil snap rename	Renames a snapshot.

acfsutil snap convert

Purpose

Converts the type of an existing snapshot image from read-only to read-write or read-write to read-only.

Syntax and Description

```
acfsutil snap convert -h
acfsutil snap convert [-r|-w] snap_shot mount_point
```

`acfsutil snap convert -h` displays help text and exits.

The following table contains the options available with the `acfsutil snap convert` command.

Table 6-56 Options for the acfsutil snap convert command

Option	Description
`-r`	Converts snapshot to read-only snapshot.
`-w`	Converts snapshot to read-write snapshot.
`snap_shot`	Specifies the name of the snapshot to be converted.
`mount_point`	Specifies the directory where the file system is mounted.

`acfsutil snap convert` converts the type of an existing snapshot image from a read-only snapshot image to a read-write snapshot image, or from a read-write snapshot image to a read-only snapshot image.

The type option (`-r` or `-w`) is required for the convert operation. Specifying a type parameter that matches the type of the existing snapshot image results in no conversion and no failure. An 11.2 read-only snapshot image can be converted to a read-write snapshot image, but this conversion results in an update of the Oracle ACFS on-disk structure version. After the Oracle ACFS on-disk structure version has been updated, the updated version is not compatible with previous Oracle ACFS 11.2 versions.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-50 shows the use of the `acfsutil snap convert` command.

Example 6-50 Using the acfsutil snap convert command

```
$ acfsutil snap convert -w midday_test1 /acfsmounts/critical_apps

$ acfsutil snap convert -r midday_test2 /acfsmounts/critical_apps
```

acfsutil snap create

Purpose

Creates a read-only or read-write snapshot of an Oracle ACFS file system or an existing snapshot.

Syntax and Description

```
acfsutil snap create -h
acfsutil snap create [-r|-w] [-p parent_snap_shot] snap_shot mount_point
```

`acfsutil snap create -h` displays help text and exits.

The following table contains the options available with the `acfsutil snap create` command.

Table 6-57 Options for the acfsutil snap create command

Option	Description
`-r`	Creates a read-only snapshot. This is the default setting.
`-w`	Creates a read-write snapshot.
`snap_shot`	Specifies a name for the snapshot. The name provided must follow host operating system specific naming rules for directories. The `.ACFS/snaps` directory itself cannot be snapped.
`-p` *`parent_snap_shot`*	Specifies the name of an existing snapshot image within the Oracle ACFS file system.
`mount_point`	Specifies the directory where the file system is mounted.

`acfsutil snap create` creates a read-only or read-write snapshot of the Oracle ACFS file system mounted on *`mount_point`* or creates a snapshot of an existing snapshot if the `-p` option is specified. You can specify `-r` for read-only or `-w` for read-write; read-only is the default if neither `-r` or `-w` are specified.

The read-write snapshot enables the fast creation of an Oracle ACFS snapshot image that can be both read and written without impacting the state of the Oracle ACFS file system hosting the snapshot images. You can use the read-write functionality for testing new versions of application software or running test scenarios on production file data without modifying the original file system.

Snapshots are not separate file systems. The snapshot appears in the `.ACFS/snaps/`*`snapshot`* directory and initially is a complete replica of the file system at the time the snapshot command was given. A read-only snapshot continues to preserve that initial point-in-time view. A read-write snapshot can be modified by updates written directly to the files located within the `.ACFS/snaps/snapshot` hierarchy.

A symbol link (symlink) file is a pointer to a path specified at the time that the symlink is created. For any existing symlink in the source of the `acfsutil snap create` command, a new link is made in the snapshot; however, the stored path to the symlink target is not modified during snap creation time. When an absolute path is used as the target of the link, the link target of the snapshot still refers to the absolute path, outside the newly-created snapshot namespace. Future modifications to the targeted file can occur through the snapshot symlink, not to a new target inside the snapshot namespace. The contents of the targeted file can also be changed outside of the snapshot. Because of this behavior, you should use relative path targets for all symlinks that are within the same file system.

Any user can access the snapshot directory by specifying the path name. However, the `.ACFS` directory itself is hidden from directory listings of the root of the file system. This prevents recursive commands, such as `rm -rf` or `acfsutil tag set -r`, from the root of the file system inadvertently operating on snapshot files.

Snapshots usually use very little storage initially as they share file system blocks with the original file system until a file changes.

Tools such as `du` report the total disk space usage of the snapshotted files, which includes the storage shared with the original versions of the files. To determine the total space used for the snapshots, use the `acfsutil snap info` or `acfsutil info fs` command.

Oracle ACFS snapshots are immediately available for use after they are created. They are always online under the `.ACFS/snaps` directory when the original file system is mounted. No separate command is needed to mount them.

The `acfsutil snap create` operation fails if the cluster is in rolling migration.

Creation from an existing snapshot is not permitted if there are:

- Any snapshots present in the file system that were created with the `ADVM` compatibility set to less than 12.1
- Any snapshots of the file system that were created after `ADVM` compatibility was set to `12.1` but while 11.2 snapshots existed
- Any snapshot deletion cleanup operations still running in the background for the above conditions

  You can display pending snapshot operations with the `acfsutil snap info` command.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-51 shows the use of the `acfsutil snap create` command to create snapshots on a Linux platform.

Example 6-51 Using the acfsutil snap create command on Linux

```
$ /sbin/acfsutil snap create -w midday_test1 /acfsmounts/critical_apps

$ /sbin/acfsutil snap create -w midday_test2 /acfsmounts/critical_apps

$ /sbin/acfsutil snap create payroll_report1 /acfsmounts/critical_apps

$ /sbin/acfsutil snap create payroll_report2 /acfsmounts/critical_apps
```

```
$ ls /acfsmounts/critical_apps/.ACFS/snaps
midday_test1  midday_test2  payroll_report1  payroll_report2
```

See Also:

- About Oracle ACFS Snapshots for information about Oracle ACFS snapshots
- Oracle ACFS Disk Space Usage for information about the number of snapshots supported
- acfsutil snap info for information about displaying pending snapshots operations
- acfsutil info fs for information about displaying the space used for snapshots
- Oracle ACFS Command-Line Tools for Snapshots for information about the `acfsutil snap duplicate apply` and `acfsutil snap duplicate create` commands

acfsutil snap delete

Purpose

Deletes a snapshot of an Oracle ACFS file system.

Syntax and Description

```
acfsutil -h snap delete
acfsutil snap delete snapshot mount_point
```

`acfsutil -h snap delete` displays help text and exits.

Table 6-58 contains the options available with the `acfsutil snap delete` command.

Table 6-58 Options for the acfsutil snap delete command

Option	Description
snapshot	Specifies a name for the snapshot.
mount_point	Specifies the directory where the file system is mounted.

`acfsutil snap delete` deletes the snapshot named *snapshot* in the Oracle ACFS mounted on *mount_point*. After successful completion of the command, the representation of the snapshot in the `.ACFS/snaps` directory is removed. The command succeeds even if there are open files or directory references; however, storage associated with the snapshot is not released until all such references are closed.

The disk space used by the snapshot being deleted is released by a background task after the completion of the `acfsutil snap delete` command or after the last close of a file in the snapshot. If one of these background threads is running to clean up a deleted snapshot, then the `acfsutil snap info` command shows a pending delete operation. For information about the `acfsutil snap info` command, refer to "acfsutil snap info".

The `acfsutil snap delete` operation fails if the cluster is in rolling migration.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-52 shows the use of the `acfsutil snap delete` command.

Example 6-52 Using the acfsutil snap delete command

```
$ /sbin/acfsutil snap delete midday_test1 /acfsmounts/critical_apps
acfsutil snap delete: Snapshot operation is complete.
```

acfsutil snap duplicate apply

Purpose

Applies a previously created duplicated snapshot to an Oracle ACFS snapshot or file system.

Syntax and Description

```
acfsutil snap duplicate apply -h
acfsutil snap duplicate apply [-b | -B snap_shot] [-d trace_level] [snap_shot]
mount_point
```

`acfsutil snap duplicate apply -h` displays help text and exits.

The following table contains the options available with the `acfsutil snap duplicate apply` command.

Table 6-59 Options for the acfsutil snap duplicate apply command

Option	Description
`—b`	Specifies that a backup snapshot is generated after a successful completion of the command. The name of the backup is generated in the format `SDBACKUP_seccnt_sourcesnap_targetsnap`, which is described in the text of this topic.
`—B` *`snap_shot`*	Specifies that a backup snapshot with the name specified is generated after a successful completion of the command.
`-d` *`trace_level`*	Specifies the trace level setting [0..6] for the command execution.
`snap_shot`	Specifies that the named snapshot of that file system is the target. The specified snapshot must exist and must be writable. If the *snap_shot* is not specified, the target for the apply operation is the file system currently mounted at the mount point
`mount_point`	Specifies the directory where the duplicate snapshot is applied.

The `acfsutil snap duplicate apply` command uses the snapshot duplication stream produced by `acfsutil snap duplicate create` to update the specified target file system or snapshot. The stream of data is read from the standard input.

If the target has never been applied to by a `acfsutil snap duplicate apply` operation, this is an initial apply, and the target must be initially empty. If the target has been applied to by a previous apply operation, this is an incremental apply, and the initial contents of the target must match the contents of the older snapshot used to produce the snapshot duplication stream.

If the contents of the target no longer match the older snapshot for the stream, the command returns an error and the stream is not applied. The target must not be updated by other means while the apply operation is in progress. If another update occurs, the command returns an error and stops applying the snapshot duplication stream. If the data stream being applied is identical to the last data stream applied to the specified target, the command exits without writing anything to the target with a status value of `2`.

If the `-b` or `–B` option is specified, then a backup snapshot is taken of the target after the apply operation completes the update. Because the snapshot taken for an apply operation is automatically deleted when the next apply operation completes, the snapshot present is the one from the last-completed apply operation. The backup snapshot provides a known and consistent version of the target for the last successful apply operation and a comparison point to identify the in-progress changes made by a `acfsutil snap duplicate apply` operation.

If a name was not specified with the `–B` option, then a backup snapshot has a name of the form `SDBACKUP_seccnt_sourcesnap` or `SDBACKUP_seccnt_sourcesnap_targetsnap`. In the name string, `seccnt` is a local timestamp (number of seconds since the Unix epoch) indicating the point at which the target was captured by the backup snapshot. `sourcesnap` indicates the snapshot from the source file system that was applied by the last completed `acfsutil snap duplicate apply` operation. `targetsnap`, if present, indicates the snapshot name in the target file system that was updated by the last completed `acfsutil snap duplicate apply` operation.

System administrator (`root`) privileges are required to use this command.

The `acfsutil snap duplicate apply` command produces an exit status of `0` on success, `1` if an error occurs, or `2` if the data stream read matches the last previous data stream applied to the specified target.

For information about displaying snapshot information, refer to "acfsutil snap info".

Examples

Example 6-53 shows the use of the `acfsutil snap duplicate apply` command. Because the command expects to read a binary data stream from its standard input, this command is normally used in a pipeline with the `acfsutil snap duplicate create` command, as shown in the example.

Example 6-53 Using the acfsutil snap duplicate apply command

```
$ /sbin/acfsutil snap duplicate create new_snapshot_name /acfsmounts/
critical_apps | \
      /sbin/acfsutil snap duplicate apply /acfsmounts/ca_backups
```

acfsutil snap duplicate create

Purpose

Creates a duplicate snapshot of an existing snapshot.

Syntax and Description

```
acfsutil snap duplicate create -h
acfsutil snap duplicate create [-r] [-d trace_level] [-p parent_snapshot]
                               [-i old_snapshot] new_snapshot mount_point
acfsutil snap duplicate create -a [-d trace_level] mount_point
```

`acfsutil snap duplicate create -h` displays help text and exits.

Table 6-60 contains the options available with the `acfsutil snap duplicate create` command.

Table 6-60 Options for the acfsutil snap duplicate create command

Option	Description
`-r`	Specifies to mark the snap duplication stream as a restart stream. This option is intended for use on the second or later attempt to create and apply a given stream. This option might be necessary if an earlier attempt terminated prematurely.
`-d` *trace_level*	Specifies the trace level setting [0..6] for the command execution.
`-p` *`parent_snapshot`*	Specifies the name of a parent snapshot image of which both the *`old_snapshot`* and *`new_snapshot`* are children. This option ensures that both snapshots are children (*snap-of-snap*) of the same snapshot. If the `–p` option is not specfied, then either the *`new_snapshot`* must be a child of the *`old_snapshot`*, or the two snapshots must be snapshots of the base file system at the specified mount point.
`–i` *`old_snapshot`*	Specifies the name of an existing snapshot.
`new_snapshot`	Specifies a name for the new snapshot.
`mount_point`	Specifies the directory where the file system is mounted. This is the mount point of the file system of which the snapshots were taken.
`-a`	Specifies to perform a sparse backup of the file system mounted at *`mount_point`*, plus all of its snapshots.

> **Note:**
>
> If too many snapshots exist of the named file system, the `-a` option to create a sparse backup cannot be used. As of the 23ai release, a maximum of 511 snapshots may be present in order for `-a` to be used.

The `acfsutil snap duplicate create` command externalizes snapshot-related data in a form suitable for use by snap duplicate apply to create a different instance of a compatible snapshot. This externalized data is called a snapshot duplication stream.

In the second form of the command:

- If only the new snapshot name is specified, then this command encodes the entire contents of the named snapshot (the difference between an empty snapshot and this snapshot). This is an *initial* snap duplicate operation.
- If both the old snapshot name and new snapshot name are given, this command encodes only the differences between the two snapshots. This is an *incremental snap duplicate* operation.

In the third form of the command:

- This command encodes the entire contents of the named file system, plus all of its snapshots.
- This *sparse backup* of the file system preserves the sparseness of all of the snapshots. That is, any blocks shared between snapshots, or between the file system and snapshots, are still shared in the backup.
- A sparse backup is always a full backup. There is no provision for performing an incremental backup.
- The named file system can be in active, read-write use. However, to capture a consistent version of the set of snapshots, this command will freeze the file system during the first phase of its operation, and will thaw the file system once that phase completes. Updates to the file system may block until that point.
- In restoring the backup:
  - `acfsutil snap duplicate apply` should be run with just the target file system specified. No options should be given. The file system specified must be empty.
  - No special characteristics of the snapshots beyond their writability are preserved.
  - Snapshots that had been defined as replication snapshots are no longer defined that way.
  - Though any of the snapshots may be used as the basis for an initial `acfsutil snap duplicate create` operation, they may not be used as the basis for a non-initial (incremental) `acfsutil snap duplicate create` operation.
  - The file system and snapshots cannot be used as targets for any further `acfsutil snap duplicate apply` commands.
  - The restored backup is not considered read-only. (If the restored backup is to be treated as read-only, it needs to be mounted that way.)

In all cases, this command writes the snapshot duplication stream to its standard output.

System administrator (`root`) privileges are required to use this command.

The `acfsutil snap duplicate create` command produces an exit status of `0` on success or `1` if an error occurs.

For information about displaying snapshot information, refer to "acfsutil snap info".

Examples

The example below shows the use of the `acfsutil snap duplicate create` command for an initial snap duplicate operation, in which the snapshot `new_snapshot_name` is applied to `/acfsmounts/ca_backups`. Because the command writes a binary data stream to its standard output, this command is normally used in a pipeline with the `acfsutil snap duplicate apply` command, as shown in the example.

Example 6-54 Using `acfsutil snap duplicate create` for an initial operation

```
$ /sbin/acfsutil snap duplicate create new_snapshot_name /acfsmounts/critical_apps | \
      /sbin/acfsutil snap duplicate apply /acfsmounts/ca_backups
```

The next example shows the use of the `acfsutil snap duplicate create` command to create a sparse backup of the file system mounted at `/source`, plus all of its snapshots. It then shows the use of `acfsutil snap duplicate apply` to restore the sparse backup to the file system mounted at `/target`.

Example 6-55 Using `acfsutil snap duplicate` commands for a sparse backup

```
# acfsutil snap dup create -a /source > /bkps/source_sparse_bkp
# acfsutil snap dup apply /target < /bkps/source_sparse_bkp
```

acfsutil snap info

Purpose

Displays information about Oracle ACFS file system snapshots.

Syntax and Description

```
acfsutil snap info -h
acfsutil snap info [-t] [snap_shot] mount_point
```

`acfsutil snap info -h` displays help text and exits.

The following table contains the options available with the `acfsutil snap info` command.

Table 6-61 Options for the acfsutil snap info command

Option	Description
`-t`	Displays a tree structure for the specified snapshot and mount point.
`snap_shot`	Specifies a name for the snapshot.
`mount_point`	Specifies the directory where the file system is mounted.

The `acfsutil snap info` command displays information about an individual specified snapshot or all snapshots in the specified Oracle ACFS file system.

The snapshot name, snapshot type, parent name, creation date and time, and space used are displayed. The parent name is either the mount point, or the parent snapshot if the snapshot was created from an existing snapshot.

The type of a snapshot is indicated by one or more of the following strings. Any relevant strings are displayed on the same line.

- `RO` - snapshot is read-only
- `RW` - snapshot is read-write
- `Primary` - snapshot is a primary storage location for replication
- `Standby` - snapshot is a standby storage location for replication

- `REPL` - snapshot is an internal snapshot used by replication
- `DUP` - snapshot or its parent is a target of snap duplicate apply

The appearance of the string `REPL` is not related to whether a snapshot is a replication primary or standby. This string appears only if the snapshot is one that was created by replication for its own internal use.

The string `DUP` appears if either of the following conditions is true:

- The snapshot is in use as the target of an apply operation.
- The parent of the snapshot was in use as the target of an apply operation at the point when the snapshot was created.

The `snapshot space usage` amount includes snapshot metadata. If all the files are deleted from a snapshot, some metadata still remains and that amount is displayed with `acfsutil snap info`.

When using an accelerator volume, the reported `snapshot space usage` amount can include space that is in use on the accelerator volume.

To reclaim the disk space used by snapshots, delete the snapshots. Deleting the files within the snapshots may not reclaim this disk space.

The `acfsutil snap info` command shows the status of a snapshot that is in the process of being deleted as follows:

- When waiting on files to be closed:

  snapshot_name (delete waiting for last close)

- When freeing the storage for the snapshot in the background:

  snapshot_name (delete in progress)

Oracle ACFS caches and asynchronously updates snapshot storage usage statistics. If an Oracle ASM instance is terminated, power is lost to the machine, or the Oracle ACFS file system goes offline for any reason, one of these cached updates may be lost. This loss can result in the display of a message similar to `Reported snapshot space usage is inaccurate` when `acfsutil snap info` is run.

Examples

Example 6-56 shows the use of the `acfsutil snap info` command on a Linux platform. The command output contains information about a read-only (RO) snapshot, a read-write (RW) snapshot with quota information, a snapshot in `delete waiting for last close` status, and a snapshot in `delete in progress` status. After a delete operation completes on a snapshot and the background process finishes the clean up of a deleted snapshot, messages about the delete operation do not appear in the output.

Example 6-56 Using the acfsutil snap info command on Linux

```
$ /sbin/acfsutil snap info /acfsmounts/critical_apps/
snapshot name:               midday_test (delete waiting for last close)
snapshot creation time:      Tue Apr  5 10:52:55 2016


snapshot name:               payroll_report1
snapshot location:           /acfsmounts/critical_apps/.ACFS/snaps/
payroll_report1
```

```
RO snapshot or RW snapshot:  RW
parent name:                 /acfsmounts/critical_apps/
snapshot creation time:      Tue Apr  5 10:52:59 2016
storage added to snapshot:   95514624   (  91.09 MB )

quota:                       104857600  ( 100.00 MB )
quota usage:                 91 %


snapshot name:               midday_test2 (delete in progress)
snapshot creation time:      Mon Apr 11 11:40:13 2016


snapshot name:               payroll_report2
snapshot location:           /acfsmounts/critical_apps/.ACFS/snaps/payroll_report2
RO snapshot or RW snapshot:  RO
parent name:                 midday_test2
snapshot creation time:      Mon Apr 11 12:05:42 2016
storage added to snapshot:   1081344    (   1.03 MB )


    number of snapshots:  3 (active)
                          1 (delete in progress)
        snapshot space usage: 102084608  (  97.36 MB )


$ /sbin/acfsutil snap info -t /acfsmounts/critical_apps
/acfsmounts/critical_apps RW
midday_test               RW    /acfsmounts/critical_apps
payroll_report1           RW    /acfsmounts/critical_apps
midday_test2              RW    /acfsmounts/critical_apps
payroll_report2           RO    midday_test2


$ /sbin/acfsutil snap info -t midday_test2 /acfsmounts/critical_apps
midday_test2              RW
payroll_report2           RO    midday_test2
```

acfsutil snap link

Purpose

Creates or deletes a snapshot link.

Syntax and Description

```
acfsutil snap link -h
acfsutil snap link -s snap_shot path_to_link
acfsutil snap link -s snap_shot -d path_to_link
```

`acfsutil snap link -h` displays help text and exits.

The following table contains the options available with the `acfsutil snap link` command.

Table 6-62 Options for the acfsutil snap link command

Option	Description
`—s` *`snap_shot`*	Specifies the name of an Oracle ACFS snapshot in the Oracle ACFS file system.
`-d`	Deletes the specified snapshot link.
`path_to_link`	Specifies a directory in the Oracle ACFS file system.

You can create or delete snapshot links with the `acfsutil snap link` command. A snapshot link provides access to the contents of a snapshot using an alternate path which does not include the `.ACFS/snaps` directory, the default location where snapshot are created.

A snapshot link is a specialized hard link that requires that only one snapshot link may exist for a snapshot. Snapshot links can only be created in the primary file system, not within another snapshot. Snapshot links can be recreated to change the name or location of the snapshot link. The name that you specify for a snapshot link does not have to be the same as the name or the snapshot to which it links. A snapshot link does require on disk changes and the `COMPATIBLE.ADVM` disk group attribute must be set to `18.0` or higher. Snapshot links can only be created and deleted with the `acfsutil snap link` command. The links cannot be moved or removed with normal operating system commands.

You can display information about snapshot links with the `acfsutil snap info` command. You can use `acfsutil snap info` *`snap_shot`* to determine whether the specified snapshot has a link.

Examples

The following example shows the use of the `acfsutil snap link` command. After a new snapshot is created, a link is created to the new snapshot.

Example 6-57 Using the acfsutil snap link command

```
$ /sbin/acfsutil snap create -w snap1 /usmmnt
acfsutil snap create: Snapshot operation is complete.

$ /sbin/acfsutil snap link -s snap1 /usmmnt/dir1/link
acfsutil snap link: Snapshot operation is complete.

$ /sbin/acfsutil snap info /usmmnt
snapshot name:              snap1
snapshot location:          /usmmnt/.ACFS/snaps/snap1
snapshot link:              /usmmnt/dir1/link
RO snapshot or RW snapshot:  RW
parent name:                /usmmnt
...
```

acfsutil snap quota

Purpose

Sets the quota limit for a specified snapshot.

Syntax and Description

```
acfsutil -h snap quota
```

```
acfsutil snap quota [+|-]n[K|M|G|T|P] snapshot mount_point
```

`acfsutil –h snap quota` displays help text and exits.

The following table contains the options available with the `acfsutil snap quota` command.

Table 6-63 Options for the acfsutil snap quota command

Option	Description
[+|-]*n*	Specifies the quota limit size for the Oracle ACFS snapshot where *n* is a valid positive whole number greater than zero. The number can be preceded by a + or - to indicate the amount to add or decrease. If no operand exists, the new size is the absolute size.
K|M|G|T|P	Specifies that the integer supplied for size is in the units of K (Kilobytes), M (Megabytes), G (Gigabytes), T (Terabytes), or P (Petabytes). If the unit indicator is specified, then it must be appended to the integer. If omitted, the default unit is bytes.
snapshot	Specifies a name of the snapshot.
mount_point	Specifies the directory where the file system is mounted.

The acfsutil snap quota command sets the quota limit for a specified snapshot on the specified Oracle ACFS file system.

The quota can be larger than the size of the file system because a file system can be resized or automatically resized.

The quota limit specified does not reserve space. The quota only establishes an upper limit on the amount of storage that can be added to the snapshot.

Only snapshots created after ADVM compatibility (`COMPATIBLE.ADVM`) has been set to `12.2` or greater have storage accounting tracking and the ability to have quotas set.

Snapshots can be converted from read-write (RW) to read-only (RO) and vice versa. Both RW and RO snapshots can have a quota.

There is a grace period and advanced warnings for quota usage. A warning message is written to both the alert log and the console if the requested storage allocation causes the local storage usage to be near or exceeding more than `90%` of the quota. If an allocation exceeds the quota, then the write or other modifying operation fails as `EDQUOT`, or it may fail as `ENOSPC`, depending whether the quota limit is set larger than the size of the file system.

System administrator (`root`) privileges are required to use this command.

You can display snapshot information, including the quota limit and quota usage percentage, with the `acfsutil snap info` command. For information about `acfsutil snap info`, refer to "acfsutil snap info".

Examples

The following example shows the use of the `acfsutil snap quota` command.

Example 6-58 Using the acfsutil snap quota command

```
$ /sbin/acfsutil snap quota +500M snap_test /acfsmounts/acfs1
```

acfsutil snap remaster

Purpose

Remasters an Oracle ACFS file system with a snapshot.

Syntax and Description

```
acfsutil snap remaster -h
acfsutil snap remaster { snapshot | -c } -f volume
```

`acfsutil snap remaster -h` displays help text and exits.

Table 6-64 contains the options available with the `acfsutil snap remaster` command.

Table 6-64 Options for the acfsutil snap remaster command

Option	Description
`–c`	Continues the interrupted remaster operation. After the command has started, the remastering must run to completion.
`–f`	Forces the operation and does not prompt for questions.
`snapshot`	Specifies the name of an existing snapshot.
`volume`	Specifies specifies the primary Oracle ADVM volume device path for the file system.

This command accepts a snapshot name or the `–c` option.

The specified snapshot is used for the base of the file system, replacing the current base of the file system. Using an existing read-only or read-write snapshot, the command reintegrates the snapshot extents into the base of the current file system creating a new remastered file system with the point in time state of the snapshot. The remastered file system is left with no snapshots. This operation must be done on an unmounted file system and it cannot be reversed.

Use the `-c` option, instead of the snapshot name, to complete an interrupted snapshot remastering.

You cannot remaster a file system on which replication, security, or encryption is running. You must terminate replication, security, or encryption before running the `acfsutil snap remaster` command. In addition, you cannot remaster a file system with plugins enabled.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-59 shows the use of the `acfsutil snap remaster` command.

Example 6-59 Using the acfsutil snap remaster command

```
$ /sbin/acfsutil snap remaster snapshot volume
```

acfsutil snap rename

Purpose

Renames an Oracle ACFS snapshot.

Syntax and Description

```
acfsutil snap rename -h
acfsutil snap rename old_snapshot new_snapshot mount_point
```

`acfsutil snap rename -h` displays help text and exits.

Table 6-65 contains the options available with the `acfsutil snap duplicate create` command.

Table 6-65 Options for the acfsutil snap rename command

Option	Description
old_snapshot	Specifies the name of an existing snapshot.
new_snapshot	Specifies a name for the new snapshot.
mount_point	Specifies the directory where the file system is mounted.

The `acfsutil snap rename`command is used to safely rename an existing snapshot in the filesystem.

While a snapshot with open files can be renamed, applications accessing files in that snapshot experience the same behavior as operating in a non-snap directory tree in which one of the parent directories was renamed.

Renaming snapshots used in conjunction with pluggable databases (PDBs) is not supported at this time. Such a snapshot should not be renamed.

Replication-specific snapshots cannot be renamed as this may cause complications and inconsistencies with replication.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

For information about displaying snapshot information, refer to "acfsutil snap info".

Examples

Example 6-60 shows the use of the `acfsutil snap rename` command.

Example 6-60   Using the acfsutil snap rename command

```
$ /sbin/acfsutil snap rename old_snapshot_name new_snapshot_name /acfsmounts/
critical_apps
```

Oracle ACFS Command-Line Tools for Compression

This topic provides a summary of commands to manage Oracle ACFS compression.

Table 6-66 lists Oracle ACFS compression commands with brief descriptions. For an overview of Oracle ACFS compression, refer to Oracle ACFS Compression.

For more information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

Table 6-66   Summary of commands for Oracle ACFS compression

Command	Description
acfsutil compress copy	Copies and compresses a file or files in a directory.
acfsutil compress info	Displays compression information about a file.
acfsutil compress off	Disables Oracle ACFS file compression.
acfsutil compress on	Enables Oracle ACFS file compression.

acfsutil compress copy

Purpose

Compresses and copies a file.

Syntax and Description

```
acfsutil compress copy -h
acfsutil compress copy [-v] [-f] {-c size source | [-n] {-r source | source
[...]}
                       target_dir
```

The following table contains the options available with the `acfsutil compress copy` command.

Table 6-67   Options for the acfsutil compress copy command

Option	Description
`-v`	Specifies verbose mode.
`-f`	Specifies to overwrite any existing copies.
`—c` *`size source`*	Copies a single source file and uses compression unit (CU) *size* for the copy. The format of *size* is *n*`K`. The CU sizes supported are: `4K`, `8K`, `16K`, `32K`, `64K` and `128K` bytes.
`—n` *`source`*	Specifies that copies created are not compressed.
`—r` *`source`*	Specifies to recursively copy files and directories starting from a single source directory.

Table 6-67 (Cont.) Options for the acfsutil compress copy command

Option	Description
source [...]	Specifies source files to be copy.
target_dir	Specifies the target directory to copy into.

The `acfsutil compress copy` command does not copy over an existing files unless the `-f` option is used. If `-f` is not used, the copy operation terminates when the first existing target is found.

The command does not copy to an Oracle ACFS file system which is capable of supporting compression if compression is disabled unless the `-n` option is specified.

Example 6-61 Using the acfsutil compress copy command

```
$ /sbin/acfsutil compress copy -v -f my_file1 my_file2 /my_target_directory
```

acfsutil compress info

Purpose

Displays information about compressed files.

Syntax and Description

```
acfsutil compress info -h
acfsutil compress info pathname
```

The following table contains the options available with the `acfsutil compress info` command.

Table 6-68 Options for the acfsutil compress info command

Option	Description
pathname	Specifies the pathname to a file.

Example 6-62 Using the acfsutil compress info command

```
$ /sbin/acfsutil compress info /my_mount_point/testfile1

Compression Unit size: 8192
Disk storage used: ( 189.89 MB )
Disk storage saved: ( 810.20 MB )
Storage used is 18% of what the uncompressed file would use
```

acfsutil compress off

Purpose

Disables Oracle ACFS compression.

Syntax and Description

```
acfsutil compress off -h
acfsutil compress off mount_point
```

The following table contains the options available with the `acfsutil compress off` command.

Table 6-69 Options for the acfsutil compress off command

Option	Description
mount_point	Specifies the directory where the file system is mounted.

When running `acfsutil compress off`, the following apply:

- Files that are created after compression is disabled are created uncompressed.
- Any compressed files in the file system remain compressed.

You must have root or system administrator privileges to run this command.

Example 6-63 Using the acfsutil compress off command

```
$ /sbin/acfsutil compress off /my_mount_point
```

acfsutil compress on

Purpose

Enables Oracle ACFS compression.

Syntax and Description

```
acfsutil compress on -h
acfsutil compress on [-a algorithm] mount_point
```

The following table contains the options available with the `acfsutil compress on` command.

Table 6-70 Options for the acfsutil compress on command

Option	Description
—a *algorithm*	Specifies the default compression algorithm. Currently only `lzo` is available and that is the default algorithm used.
mount_point	Specifies the directory where the file system is mounted.

> **Note:**
>
> Oracle ACFS compression is not supported on Oracle Database files. Oracle ACFS file systems which are intended to hold database files should not be compressed with Oracle ACFS compression.

When running `acfsutil compress on`, the following apply:

- Files created after compression is enabled are compressed by default.
- Any uncompressed files in the file system remain uncompressed.

You must have root or system administrator privileges to run this command.

Example 6-64 Using the acfsutil compress on command

```
$ /sbin/acfsutil compress on /my_mount_point
```

Oracle ACFS Command-Line Utilities

This topic provides a summary of the Oracle ACFS command-line utilities.

Table 6-71 lists the Oracle ACFS command-line utilities with brief descriptions.

For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

Table 6-71 Summary of Oracle ACFS command-line utilities

Command	Description
acfsutil accel replace	Replaces an existing accelerator volume with a new one.
acfsutil cluster info	Displays the information on the type of cluster and internal information for the running cluster.
acfsutil compat get	Displays the compatibility level of an Oracle ACFS file system.
acfsutil compat set	Changes the compatibility level of an Oracle ACFS file system.
acfsutil defrag dir	Defragments all the files in a specified directory.
acfsutil defrag file	Defragments the specified files.
acfsutil freeze	Freezes modification activity temporarily in the file system.
acfsutil fshare create	Create a file in the namespace which is shared with the source file.
acfsutil info file	Displays information for a file in an Oracle ACFS file system.
acfsutil info fs	Displays detailed Oracle ACFS file system information.
acfsutil info id	Displays Oracle ACFS file system information for a specified file identifier and mount point.
acfsutil info storage	Displays space usage for the Oracle ASM, Oracle ACFS, and Oracle ADVM components in Oracle ASM disk groups.
acfsutil plugin disable	Disables the Oracle ACFS plug-in infrastructure.
acfsutil plugin enable	Enables the Oracle ACFS plug-in infrastructure.
acfsutil plugin info	Displays information about the Oracle ACFS plug-in infrastructure.
acfsutil registry	Adds, deletes, or displays entries in the Oracle ACFS mount registry.
acfsutil rmfs	Removes an Oracle ACFS file system.
acfsutil scrub	Checks for and reports any inconsistencies in the metadata or file data.
acfsutil size	Resizes an Oracle ACFS file system.
acfsutil thaw	Resumes activity after an acfsutil freeze command has been issued on a file system.
acfsutil version	Displays Oracle ACFS version information.

Table 6-71 (Cont.) Summary of Oracle ACFS command-line utilities

Command	Description
advmutil canonical	Displays the canonical name of an Oracle ADVM volume.
advmutil volinfo	Displays information about Oracle ADVM volumes.

acfsutil accel replace

Purpose

Replaces an existing accelerator volume with a new one.

Syntax and Description

```
acfsutil accel replace -h
acfsutil accel replace -a new_accel_volume primary_volume
```

`acfsutil accel replace -h` displays help text and exits.

The following table contains the options available with the `acfsutil accel replace` command.

Table 6-72 Options for the acfsutil accel replace command

Option	Description
`—a new_accel_volume`	Specifies the name of the new accelerator volume.
`primary_volume`	Specifies the name of the primary volume with which the Oracle ACFS file system is mounted.

This command allows an administrator to replace an existing accelerator volume with a new accelerator volume. The command is useful if the current accelerator volume is full and cannot be grown, or to migrate an accelerator to a faster volume.

The file system must be unmounted on all nodes to use this command. The new accelerator must be at least as large as the existing accelerator. If new accelerator volume is larger than the existing accelerator volume, then the administrator should run `acfsutil resize -d new_accel_volume` after the file system is remounted following the `acfsutil accel replace` operation to enable Oracle ACFS to utilize the entire space of the new accelerator volume.

You must be the `root` user or an Oracle ASM administrator user to run this command.

The following example shows the use of the `acfsutil accel replace` command.

Example 6-65 Using the acfsutil accel replace command

```
$ /sbin/acfsutil accel replace -a new_accelerator_volume my_primary_volume
```

acfsutil cluster info

Purpose

Displays information about the type of cluster and the internal information for the running cluster.

Syntax and Description

```
acfsutil cluster info -h
acfsutil cluster info
```

`acfsutil cluster info` displays information about the type of cluster and the internal information for the running cluster. The command can be used to ensure that the kernel Oracle ACFS cluster is in sync with Oracle ASM. This command is applicable on all cluster types.

Example 6-66 Using acfsutil cluster info

```
# /sbin/acfsutil cluster info
Node Count: 3
Rebuild Manager: 3
Local Node: 3
Current Incarnation: 96
ACFS DLM Interfaces: ENABLED
OKS DLM Interfaces: ENABLED
Cluster State: [ NORMAL_OPERATION ]
Oracle Appliance: [NONAPP]
Nodes:
    Node 1 - IP 169.254.10.251 (port x) - nodex
    Node 2 - IP 169.254.15.46 (port y) - nodey
    Node 3 - IP 169.254.12.235 (port z) - nodez
```

The "port" number listed is the port that acfs listens on for incoming tcp connections on the private interconnect.

- Node Count: Number of nodes in the cluster.

  In this example, this is a 3–node cluster.
- Rebuild Manager: Node that manages the cluster recovery.

  In this example, node 3 is in charge of rebuilding the cluster.
- Current Incarnation: An incarnation is any significant membership change in the cluster.

  In this example, there have been 96 incarnations of the cluster.
- Cluster State: NORMAL_OPERATION, MIGRATION, UNKNOWN

  In this example, the cluster is operating normally.

acfsutil compat get

Purpose

Displays the Oracle ACFS compatibility level for the file system and the `COMPATIBLE.ADVM` level for the disk group.

Syntax and Description

```
acfsutil compat get -h
acfsutil compat get mount_point
```

`acfsutil compat get -h` displays help text and exits.

The following table contains the options available with the `acfsutil compat get` command.

Table 6-73 Options for the acfsutil compat get command

Option	Description
mount_point	Specifies the directory where the file system is mounted.

The `acfsutil compat get` command displays the Oracle ACFS compatibility level for the file system that may have been set by the `mkfs` command or the `acfsutil compat set` command. `acfsutil compat get` also displays the `COMPATIBLE.ADVM` setting for the disk group, which is default compatibility level for the file system if not otherwise explicitly set. For information about Oracle ASM disk group compatibility attributes, see Oracle ACFS Features Enabled by Compatibility Attribute Settings.

Examples

The following example shows the use of the `acfsutil compat get` command.

Example 6-67 Using the acfsutil compat get command

```
$ /sbin/acfsutil compat get /acfsmounts/acfs1
```

acfsutil compat set

Purpose

Change the compatibility level of an Oracle ACFS file system.

Syntax and Description

```
acfsutil compat set -h
acfsutil compat set -c release_version mount_point
```

`acfsutil compat set -h` displays help text and exits.

The following table contains the options available with the `acfsutil compat set` command.

Table 6-74 Options for the acfsutil compat set command

Option	Description
`–c` *release_version*	Specifies the release version for the Oracle ACFS file system determined by *mount_point*.
mount_point	Specifies the directory where the file system is mounted.

The value specified with the `-c release_version` option must be greater than or equal to the value of `COMPATIBLE.ADVM` for the disk group, and must be less than or equal to the running Oracle Grid infrastructure release version. If `–c release_version` is not specified, the value of `COMPATIBLE.ADVM` is used. After the compatibility is set, it cannot be downgraded. The `–c release_version` option can be used in situations where it is not possible, or desirable, to update `COMPATIBLE.ADVM` and `COMPATIBLE.ASM` for the disk group, but you want to use an Oracle ACFS feature that requires a compatibility increase. After the Oracle ACFS compatibility is updated, it is no longer possible to mount the file system with an older Oracle Grid infrastructure release. For information about Oracle ASM disk group compatibility attributes, see Oracle ACFS Features Enabled by Compatibility Attribute Settings.

The following example shows the use of the `acfsutil compat set` command.

Example 6-68 Using the acfsutil compat set command

```
$ /sbin/acfsutil compat set -c 12.2 /acfsmounts/acfs1
```

acfsutil defrag dir

Purpose

Defragments all the files in a specified directory.

Syntax and Description

```
acfsutil defrag dir -h
acfsutil defrag dir [-r] dir_path
```

`acfsutil defrag dir -h` displays help text and exits.

Table 6-75 contains the options available with the `acfsutil defrag dir` command.

Table 6-75 Options for the acfsutil defrag dir command

Option	Description
`–r`	Specifies recursive defragment operation.
dir_path	Specifies the directory where the files to be defragmented are located.

The `acfsutil defrag dir` command is used to defragment files in the specified subdirectory. The `–r` option is used to recursively defragment files in the subdirectories of the specified directory.

You must have write access to the files to be defragmented.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-69 shows the use of the `acfsutil defrag dir` command.

Example 6-69 Using the acfsutil defrag dir command

```
$ /sbin/acfsutil defrag dir /acfsmounts/critical_apps/01jan1015
```

acfsutil defrag file

Purpose

Defragments the specified files.

Syntax and Description

```
acfsutil defrag file -h
acfsutil defrag file [-i] file_path [file_path ... ]
```

`acfsutil defrag file -h` displays help text and exits.

Table 6-76 contains the options available with the `acfsutil defrag file` command.

Table 6-76 Options for the acfsutil defrag file command

Option	Description
`–i`	Estimates the reduction in number of extents possible. Requires read access to the file.
`file_path`	Specifies the file to be defragmented.

The `acfsutil defrag file` command defragments files. The `–i` option displays the estimated reduction in the number of extents for the files to be defragmented.

You must have write access to the files to be defragmented.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-70 shows the use of the `acfsutil defrag file` command.

Example 6-70 Using the acfsutil defrag file command

```
$ /sbin/acfsutil defrag file /acfsmounts/critical_apps/testfile
```

acfsutil freeze

Purpose

Freezes modification activity temporarily in the file system.

Syntax and Description

```
acfsutil freeze -h
acfsutil freeze [-f] mount_point
```

`acfsutil freeze -h` displays help text and exits.

The following table contains the options available with the `acfsutil freeze` command.

Table 6-77 Options for the acfsutil freeze command

Option	Description
`-f`	Specifies the command to flush data to disk before freezing the file system.
`mount_point`	Specifies the directory where the file system is mounted.

The `acfsutil freeze` command temporarily halts modification activity on a file system. You can use `acfsutil freeze` to create point-in-time images across different snapshots without stopping applications. To resume activity after the `acfsutil freeze` command has been issued on a file system, run the `acfsutil thaw` command. For information about `acfsutil thaw`, refer to acfsutil thaw.

Use the `acfsutil freeze` command with caution to prevent application timeouts and possible instance evictions. If a database or database diagnostic area is running on an Oracle ACFS file system and the file system is frozen for more than three minutes, the database instance can be terminated due to processes that are hung while waiting to update files on the file system.

Using the `acfsutil freeze` command on a file system where Oracle ACFS replication is active may block replication operations if the operations modify on-disk replication configuration information or if the operations are updates to a frozen standby file system. Replication operations that are likely to be affected are:

- `acfsutil repl sync`
- `acfsutil repl terminate primary`
- `acfsutil repl terminate standby`

Examples

The following example shows the use of the `acfsutil freeze` command.

Example 6-71 Using the acfsutil freeze command

```
$ /sbin/acfsutil freeze -f /acfsmounts/acfs1
```

acfsutil fshare create

Purpose

Creates a file in the namespace which is shared with the source file.

Syntax and Description

```
acfsutil fshare create -h
```

```
acfsutil fshare create source_file_path destination_file_path
```

`acfsutil fshare create -h` displays help text and exits.

The following table contains the options available with the `acfsutil fshare create` command.

Table 6-78 Options for the acfsutil fshare create command

Option	Description
source_file_path	Specifies the path to the source file.
destination_*file_path*	Specifies the path to the destination file.

The `acfsutil fshare create` command can be used to create a file in the namespace (*fshare*) which is shared with the source file. This fshare is an exact replica of the source file and functions as a file based snapshot. This file sharing capability consumes no storage until it is modified and then the storage being modified is copied on write (COW). The fshare is similar to a file system snapshot except it is created on an individual file basis.

`acfsutil fshare create` can create a replica of a single file. The file shares all its storage with the original file until its storage is modified. Similarly the original file also allocates new storage if it is modified (COW).

An fshare of an fshare can be created; however, at any time there can only be 32K shared files in a single block of storage. If the creation of an fshare causes any block in the file to exceed 32K sharers, then the command fails.

Fshares coexist with file system snapshots. An fshare can be created in a snapshot and when a snapshot is created, all fshares that exist now also exist in the snapshot.

The creation of a fshare may require some storage so the command can fail with an out of space error. An fshare consumes space when either the fshare or the original file is modified.

An fshare behaves like regular POSIX files in that all commands that operate on regular files also operate on fshares. An fshare can be deleted like any other file. There is no acfsutil command for deleting an fshare.

If no path to the file is given, then the command creates the fshare in the current working directory. An fshare can have hard links. Symlinks are followed. An fshare can be created on a frozen file system.

Write permission is required on the directory in which the new file share is created. Read permission is required on the base file to be shared.

The following are restrictions on the file sharing capability.

- The first creation of an fshare in a namespace (the primary file system or a snapshot) requires Oracle ACFS to create metadata requiring space proportional to the size of the file system. Subsequent creations of fshares in the namespace do not need to allocate new storage in this way.
- Creating a fshare of a directory is not be allowed. Only user files can be shared.
- Fshares cannot be created across file systems or namespaces, such as from within a snapshot to another snapshot.
- If the target file of the fshare create operation exists, then the command fails.
- An fshare of a compressed file is not allowed and the compression of an fshare is not allowed.
- An fshare cannot be created on a file system that has Oracle ACFS replication version 1 enabled.

- An fshare cannot be created on a read-only snapshot.
- The file share capability requires that all nodes in the cluster must have the `ADVM.COMPATIBLE` attribute set to `20.0` or higher.

Examples

The following are examples of the use of `acfsutil fshare create`.

Example 6-72 Using acfsutil fshare create

The following command successfully creates a file share in the current working directory.

```
# acfsutil fshare create file1 file2
acfsutil fshare create: Fshare operation is complete.
```

The following command attempts, and fails, to create a file share of a directory.

```
# acfsutil fshare create dir1 file2
acfsutil fshare create: Fshare operation did not complete.
acfsutil fshare create: Cannot share a directory.
```

See Also:

- About Oracle ACFS Snapshots for information about snapshots of an Oracle ACFS file system
- Documentation about the Linux/UNIX `cp --reflink` command for information about creating a reference link to a base file.

acfsutil info file

Purpose

Displays information for a file in an Oracle ACFS file system.

Syntax and Description

```
acfsutil info file -h

acfsutil info file [ [-d] [-o acfs_extent_offset] [-m] [-u] | [-c] ] path [path ... ]

acfsutil info file -f file_name [ [-d] [-o acfs_extent_offset]
```

`acfsutil info file -h` displays help text and exits.

Table 6-79 contains the options available with the `acfsutil info file` command.

Table 6-79 Options for the acfsutil info file command

Option	Description
`-d`	Displays detailed extent information, from Oracle ACFS to the Oracle ASM devices in the disk group.

Table 6-79 (Cont.) Options for the acfsutil info file command

Option	Description
`-o` *`acfs_extent_offset`*	`acfs_extent_offset` — Filters the extent display to the Oracle ACFS file offset specified.
`-c`	Displays extent counts only.
`-f` *`file_name`*	Reports the amount of storage that the file is currently shared with shared files (fshares) and the amount of storage that is exclusive to the file. This option also displays whether each extent is shared or not. If the `-o` offset is specified, then only the extent containing the specified offset is displayed. The `-d` option displays the Oracle ASM extent information.
`-m`	Displays details about file metadata extents. The `-m` option is compatible with `-d`, `-o`, and `-u`. The `-m` option is not compatible with `-c`.
`-u`	Displays extent usage information.
`path`	Specifies the full path name of a file.

The `acfsutil info file` command displays information about a file in an Oracle ACFS file system. The information includes the extent map which details the locations of the blocks comprising the file and the used versus allocated storage for a file. Tagging information is also displayed for a file.

The `–u` option adds an additional column to the standard `acfsutil info file` output indicating whether or not the extent is inherited between a snapshot parent and child. The `-c` option displays the total extent count and the number of primary and accelerator volume extent nodes, while omitting the details of each individual extent. The output from the `–c` option is useful when you want to display the number of extents, and when using an accelerator, to ensure that the metadata is situated on the accelerator volume.

You must have read access to the specified file to run `acfsutil info file`.

Examples

The following example shows the use of the `acfsutil info file` command with various options.

Example 6-73 Using the acfsutil info file command

```
$ /sbin/acfsutil info file /usmmnt/.ACFS/snaps/s1/afile
  /usmmnt/.ACFS/snaps/s1/afile
    flags:        File
    inode:        18014398509482029
    owner:        root
    group:        root
    size:         13013  (  12.71 KB )
    allocated:    24576  (  24.00 KB )
    hardlinks:    1
    device index: 1
    major, minor: 202,64
    access time:  Thu Jan  7 14:13:24 2016
```

```
    modify time:  Thu Jan  7 14:13:24 2016
    change time:  Thu Jan  7 14:13:24 2016
    extents:
        -offset ----length | -dev --------offset
              0      24576 |    1      142188544
         extent count: 1
         granularity level: -1


$ /sbin/acfsutil info file -c /usmmnt/.ACFS/snaps/s1/afile
  /usmmnt/.ACFS/snaps/s1/afile
    flags:        File
    inode:        18014398509482029
    owner:        root
    group:        root
    size:         13013  (  12.71 KB )
    allocated:    24576  (  24.00 KB )
    hardlinks:    1
    device index: 1
    major, minor: 202,64
    access time:  Thu Jan  7 14:13:24 2016
    modify time:  Thu Jan  7 14:13:24 2016
    change time:  Thu Jan  7 14:13:24 2016
    extents:
         extent count: 1
         primary volume extent nodes:     0
         accelerator volume extent nodes: 0
         granularity level: -1


$ /sbin/acfsutil info file -u /usmmnt/.ACFS/snaps/s1/afile
  /usmmnt/.ACFS/snaps/s1/afile
    flags:        File
    inode:        18014398509482029
    owner:        root
    group:        root
    size:         13013  (  12.71 KB )
    allocated:    24576  (  24.00 KB )
    hardlinks:    1
    device index: 1
    major, minor: 202,64
    access time:  Thu Jan  7 14:13:24 2016
    modify time:  Thu Jan  7 14:13:24 2016
    change time:  Thu Jan  7 14:13:24 2016
    extents:
        -offset ----length | -dev --------offset | inherited
              0      24576 |    1      142188544 | Yes
         extent count: 1
         added extent count: 0
         inherited extent count: 1
         granularity level: -1
```

These are examples for `acfsutil info file -f` *file_name* to display information about shared files.

```
The file is an fshare:
# /usmmnt/file2
    flags:        File
    inode:        42
```

```
    owner:         root
    group:         root
    size:          557056  ( 544.00 KB )
    allocated:     552960  ( 540.00 KB )
    hardlinks:     1
    device index: 1
    major, minor: 202,16
    create time:  Wed Jun  6 12:40:39 2018
    access time:  Wed Jun  6 12:40:39 2018
    modify time:  Wed Jun  6 12:40:48 2018
    change time:  Wed Jun  6 12:40:48 2018
    fshare:        Yes
    extents:
        -offset ----length | -dev --------offset -shared
              0       4096 | <gap>
           4096      65536 |    1 1386610688          No
          69632       4096 |    1 1386737664          No
          73728      61440 |    1 1386676224          No
         135168       4096 |    1 1386213376         Yes
         139264       4096 |    1 1386864640          No
         143360      65536 |    1 1386741760          No
         208896       4096 |    1 1386868736          No
         212992      57344 |    1 1386807296          No
         270336       4096 |    1 1386340352         Yes
         274432       4096 |    1 1386872832          No
         278528       4096 |    1 1386995712          No
         282624      65536 |    1 1386876928          No
         348160       4096 |    1 1386999808          No
         352256      53248 |    1 1386942464          No
         405504       4096 |    1 1386467328         Yes
         409600       8192 |    1 1387003904          No
         417792       4096 |    1 1387126784          No
         421888      65536 |    1 1387012096          No
         487424       4096 |    1 1387130880          No
        -offset ----length | -dev --------offset -shared
         491520      49152 |    1 1387077632          No
         540672      16384 |    1 1386594304         Yes
         extent count: 21
         owned extent count: 17
         shared extent count: 4

Specifying the -o offset option:
# ./acfsutil info file /usmmnt/file2 -f -o 405504
     # /usmmnt/file2
    flags:         File
    inode:         42
    owner:         root
    group:         root
    size:          557056  ( 544.00 KB )
    allocated:     552960  ( 540.00 KB )
    hardlinks:     1
    device index: 1
    major, minor: 202,16
    create time:  Wed Jun  6 12:40:39 2018
    access time:  Wed Jun  6 12:40:39 2018
    modify time:  Wed Jun  6 12:40:48 2018
    change time:  Wed Jun  6 12:40:48 2018
    fshare:        Yes
    extents:
        -offset ----length | -dev --------offset -shared
         405504       4096 |    1 1386467328         Yes
```

```
The file is not an fshare:
/usmmnt/dir1
    flags:        Directory,Range-locked
    inode:        44
    owner:        root
    group:        root
    size:         20480  (  20.00 KB )
    allocated:    53248  (  52.00 KB )
    hardlinks:    2
    device index: 1
    major, minor: 202,16
    create time:  Wed Jun  6 12:37:16 2018
    access time:  Wed Jun  6 12:37:16 2018
    modify time:  Wed Jun  6 12:37:16 2018
    change time:  Wed Jun  6 12:37:16 2018
    fshare:       No
    extents:
        -offset ----length | -dev --------offset -shared -remap
              0      53248 |    1 389099520          No     No
         extent count: 1
         owned extent count: 1
         shared extent count: 0
```

acfsutil info fs

Purpose

Displays detailed Oracle ACFS file system information.

Syntax and Description

```
acfsutil info fs -h
acfsutil info fs [-l] [{-o option_item | -s [interval[count][-d]] | -f [-v]}]
                 [mount_point]
```

`acfsutil info fs -h` displays help text and exits.

Table 6-80 contains the options available with the `acfsutil info fs` command.

Table 6-80 Options for the acfsutil info fs command

Option	Description
`-l`	Displays symbolic links for an Oracle ACFS file system specified by the mount point or displays symbolic links for all mounted Oracle ACFS file systems if a mount point is not specified.

Table 6-80 (Cont.) Options for the acfsutil info fs command

Option	Description
`-o`	Displays the specific file system option item from the following list: • `allvolumes [`*`mount_point`*` | `*`primary_volume_device`*` | `*`auxiliary_volume_device`*`]` — Lists all volumes that are associated with a file system. • `autoresize` — 1 if automatic resize is enabled, 0 if not. • `autoresizeincrement` — Lists the automatic resize increment for the file system. • `autoresizemax` — Lists the automatic resize maximum. • `auxvolume {`*`mount_point`*` | `*`primary_volume_device`*`}` — Lists the auxiliary volumes associated with the file system. Can be run against a mount point or device. • `available` - Length of time the file system has been available. • `compatacfs` — Lists the ACFS compatibility setting • `compatadvm` — Lists the ADVM compatibility setting. • `diskgroup` - Name of the disk group associated with *`path`*. • `fetasize` - File entry table size which is the amount of storage consumed by the on disk metadata for inodes. • `freespace` - Amount of free space available in the file system. • `isadvm` - 1 if an Oracle ADVM volume is present, 0 if not. • `isauxvolume {`*`volume_device`*`}` — 1 if a volume is an auxiliary volume or 0 if not. Must be run against a volume device. • `isavailable` - 1 if the file system is available, 0 if file system is not available. • `iscompression` — State of compression on the file system. 1 for enabled, 0 for disabled. • `iscorrupt` - 1 if the file system is corrupt, 0 if file system is not corrupt. • `isencryption` — State of encryption on the file system, 0 for disabled, 1 for enabled.

Table 6-80 (Cont.) Options for the acfsutil info fs command

Option	Description
`-o` (continued)	• `ismountpoint` - 1 if *`path`* is the mount point, 0 if *`path`* is not a mount point. • `isreplprimary` - 1 if primary file system, 0 if not. • `isreplstandby` - 1 if standby file system, 0 if not. • `issecurity` — State of security on the file system, 0 for disabled, 1 for enabled. • `metablocksize` — Size of metadata blocks on the file system. • `mountpoints` - Mount points associated with file systems. • `primaryvolume` - Path name of the device associated with file system. • `redundancy` - Type of redundancy used by the Oracle ADVM volume. • `replication` - 1 if file system has been initialized for replication as the primary or standby file system, 0 if not initialized for replication. • `resizeincrement` - Allocation unit size in bytes of the Oracle ADVM volume used by *`path`*. • `stripewidth` - Stripe width for the Oracle ADVM volume used by *`path`*. • `stripecolumns` - Stripe set columns for the Oracle ADVM volume used by *`path`*. • `totalspace` - Size of the file system. • `volumes` - Number of volumes associated with the file system.
`-s` [*`interval`* [*`count`*] [`-d`]]	Displays file system statistics. The optional *`interval`* parameter specifies the amount of time in seconds between each report. The first report contains statistics for the time since the file system mount. Each subsequent report contains statistics collected during the interval since the previous report. The optional *`count`* parameter can be specified with the *`interval`* parameter. If the *`count`* parameter is specified, the value of count determines the number of reports generated at interval seconds apart. If the *`interval`* parameter is specified without the *`count`* parameter, the command generates reports continuously until you break out of the command with `Ctrl+c`. The optional `-d` option displays more detailed statistical information.
`-f` [`-v`]	Displays file system fragmentation. The `-v` option displays additional fragmentation information.
`mount_point`	Specifies the directory where the file system is mounted. Not a valid option for replication data.

`acfsutil info fs` displays information and statistics about Oracle ACFS file systems. If compression, encryption, security, or replication has been initiated on a file system, additional flags and status information are displayed.

The `-o` *`option_item`* option displays file system information for the *`option_item`* specified.

The `-s` option displays the amount and rate of change currently on a file system for the node that the command is run on.

With no option specified, the command displays file system information that includes the volume device name, the size of the file system, the amount of space available on the volume device, the file system mount time on this node, the state of the file system, the user specified block size, the number of Oracle ACFS snapshots, the space consumed by snapshots in the file system, and the optional name or volume label associated with the file system. The possible file system states are displayed in the `flags` line. These states include:

- `Offline` indicates that the underlying devices are not accessible, possibly due to an Oracle ASM instance failure, disk group forced dismount, or an irrecoverable I/O error. The file system on this node can only be dismounted. All other attempts at access result in errors.
- `Available` indicates that the file system is on line and operational.
- `Corrupt` indicates that the file system should be checked and repaired at the earliest possible convenience to correct a detected inconsistency. For example, run the `fsck` command on Linux to check and repair the file system. Ensure that you run the command in repair mode to correct the problem. For more information, refer to fsck (offline mode).

Any user can run `acfsutil info fs`.

Examples

The following are examples of the use of `acfsutil info fs`.

Example 6-74 displays information about an Oracle ACFS file system in a Linux environment.

Example 6-74 Using the acfsutil info fs command on Linux

```
$ /sbin/acfsutil info fs
/primary
    ACFS Version: 11.2.0.2.0
    flags:        MountPoint,Available,Replication
    mount time:   Mon Oct 25 12:11:03 2010
    volumes:      1
    total size:   5368709120
    total free:   4144230400
    primary volume: /dev/asm/pvol-74
        label:
        flags:                 Primary,Available,ADVM
        on-disk version:       40.0
        allocation unit:       4096
        major, minor:          252, 37889
        size:                  5368709120
        free:                  4144230400
        ADVM diskgroup         REPLDG
        ADVM resize increment: 268435456
        ADVM redundancy:       unprotected
        ADVM stripe columns:   4
        ADVM stripe width:     131072
    number of snapshots:  0
    snapshot space usage: 0
    replication status: primary

/standby
```

```
    ACFS Version: 11.2.0.2.0
    flags:        MountPoint,Available,Replication
    mount time:   Mon Oct 25 12:11:03 2010
    volumes:      1
    total size:   5368709120
    total free:   5263945728
    primary volume: /dev/asm/svol-74
        label:
        flags:                 Primary,Available,ADVM
        on-disk version:       40.0
        allocation unit:       4096
        major, minor:          252, 37890
        size:                  5368709120
        free:                  5263945728
        ADVM diskgroup         REPLDG
        ADVM resize increment: 268435456
        ADVM redundancy:       unprotected
        ADVM stripe columns:   4
        ADVM stripe width:     131072
    number of snapshots:  0
    snapshot space usage: 0
    replication status: standby

$ /sbin/acfsutil info fs -o mountpoints,replication
/primary
1
/standby
1

$ /sbin/acfsutil info fs -o mountpoints,isreplprimary
/primary
1
/standby
0

$ /sbin/acfsutil info fs -o mountpoints,isreplstandby
/primary
0
/standby
1
```

Example 6-75 illustrates the use of `acfsutil info fs` with `-s` option to display the current amount and rate of change on a file system.

Example 6-75   Using the acfsutil info fs command with the -s option

```
$ /sbin/acfsutil info fs -s /acfsmounts/acfs1
    amount of change since mount:       359.22 MB
    average rate of change since mount: 3 KB

$ /sbin/acfsutil info fs -s 60 4 /acfsmounts/acfs1
    amount of change since mount:       359.22 MB
    average rate of change since mount: 3 KB/s

    amount of change: 15.02 MB   rate of change: 256 KB/s
    amount of change: 9.46 MB   rate of change: 161 KB/s
    amount of change: 7.32 MB   rate of change: 125 KB/s
    amount of change: 6.89 MB   rate of change: 117 KB/s
...
```

acfsutil info id

Purpose

Displays Oracle ACFS file system information for a specified file identifier number and mount point.

Syntax and Description

```
acfsutil info id -h
acfsutil info id file_identifier mount_point
```

`acfsutil info id -h` displays help text and exits.

Table 6-81 contains the options available with the `acfsutil info id` command.

Table 6-81 Options for the acfsutil info id command

Option	Description
mount_point	Specifies the directory where the file system is mounted.
file_identifier	Specifies the Oracle ACFS file identifier number reported by the Oracle ACFS driver. The file identifier number should be specified in decimal format.

You can use `acfsutil info id` to translate an internal numeric Oracle ACFS file identifier to a path name in the file system. This is useful when the Oracle ACFS driver reports I/O errors to the system event logger associated with a particular file in an Oracle ACFS and identifies it by its internal identifier. You must have administrator privileges or you must be a member of the Oracle ASM administrator group to run `acfsutil info id`.

Examples

The following is an example of `acfsutil info id`.

Example 6-76 Using the acfsutil info id command

```
$ /sbin/acfsutil info id 117 /acfsmounts/acfs1
```

acfsutil info storage

Purpose

Displays space usage for the Oracle ASM, Oracle ACFS, and Oracle ADVM components in Oracle ASM disk groups.

Syntax and Description

```
acfsutil info storage -h
acfsutil info storage [-u {B|MB|GB|TB}] [-l] [disk_group,...]
```

`acfsutil info storage -h` displays help text and exits.

The following table contains the options available with the `acfsutil info storage` command.

Table 6-82 Options for the acfsutil info storage command

Option	Description
`-u`	Specifies the unit of measurement.
{`B`|`MB`|`GB`|`TB`}	Specifies that the space usage values are in units of `B` (kilobytes), `MB` (megabytes), `GB` (gigabytes), or `TB` (terabytes).
`-l`	Displays more detailed space information for the Oracle ASM disk group, the Oracle ADVM volume, and the Oracle ACFS file system.
disk_group	Specifies the disk group or disk groups for which space usage is displayed. If a disk group is not specified, then information for all disk groups is displayed.

`acfsutil info storage` provides a platform independent way to view how the space in an Oracle ASM disk group is being consumed by components of Oracle ASM, Oracle ACFS, and Oracle ADVM.

This command gathers the space information from multiple sources, including `V$ASM_DISKGROUP`, `V$ASM_FILE`, `acfsutil snap info`, and `acfsutil info fs`. Each line in the summary output displays the primary Oracle ACFS consumers of storage in a disk group.

For each disk group, all information for that disk group is grouped together. The `% Free` column is calculated using the `Space` and the `Usable Free` columns. For `NORMAL` and `HIGH` redundancy disk groups, the `Usable Free` column displays the unused portion of the disk group after accounting for mirroring. For `FLEX` or `EXTENDED` disk group, the `Usable Free` column reports the raw total free space as the redundancy is flexible and can vary.

The `Consumer` and `Path` columns are always empty on the line that displays information for the disk group. The `Consumer` column is populated on the lines for the consumers of storage in the disk group.

If a volume is enabled, then the volume name is displayed in the `Consumer` column. The `Space` column displays the current size of the volume. The `Usable Free` column displays the amount of free space in the file system if the volume is part of an Oracle ACFS file system or part of a mounted non-Oracle ACFS file system. The `Path` column displays the mount point if the volume is part of a mounted file system. The `Path` column displays `*ACFS`if the volume is part of an unmounted Oracle ACFS file system.

If a snapshot exists in the Oracle ACFS file system associated with the previously listed volume, then the snapshot name is displayed on the next line in the `Consumer` column. The `Space` column associated with a snapshot is always marked with an asterisk (`*`) because the space used by a snapshot in the file system has been accounted for in the volume line. The `Size With Mirroring` column associated with a snapshot represents the space currently being used by that snapshot in the file system. If a snapshot quota has been established, then the `Usable Free` column displays the amount of free space left before for that snapshot reaches the quota. If a snapshot quota has not been set, then this column is empty. The `Path` column for a snapshot is a full path to the root of the snapshot. If a snapshot link has been established, then the snapshot link is displayed in the `Path` column.

`root` or users who are members of the `asmadmin` group can run this command.

Examples

The following example shows the use of `acfsutil info storage` to display space usage information in both summary and long format.

Example 6-77 Using the acfsutil info storage command

```
$ /sbin/acfsutil info storage -u MB
Diskgroup      Consumer      Space     Size With Mirroring  Usable Free  %Free   Path
DATA                       4608.00            2304.00        895.00       38%
                   VOL1    2048.00            1024.00        831.45       81%    /acfsmnt
                 052317          *               0.25                            /acfsmnt/latest
                 052217          *             131.26        893.00       87%    /acfsmnt/.ACFS/
snaps/052217
FLASH                      2048.00            1024.00        147.00       14%
              ACCELVOL1    1024.00             512.00        260.92       50%    ACCEL for VOL1
in DATA
TST                        2048.00            1024.00        143.00       13%
                   VOL1    1024.00             512.00        512.00      100%
----
unit of measurement: MB
```

```
$ /sbin/acfsutil info storage -l

Diskgroup: DATA (38% free)
  total disk space:           4.50
  ASM file space:             2.05
  total free space:           2.25
  net free with mirroring:    1.12
  usable after reservation:   0.87
  redundancy type:          NORMAL

    Total space used by ASM non-volume files:
      used:                      0.00
      mirror used:               0.00

    volume: /dev/asm/vol1-447
      total:                     1.00
      free:                      0.81
      redundancy type:         mirror
      file system:             /acfsmnt
        snapshot: 052317 (/acfsmnt/latest)
          used:          0.00
          quota limit:   none
        snapshot: 052217 (/acfsmnt/.ACFS/snaps/052217)
          used:          0.13
          quota limit:   1.00

...

----
unit of measurement: GB
```

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges

acfsutil plugin disable

Purpose

Disables the Oracle ACFS plug-in infrastructure for an Oracle ACFS file system.

Syntax and Description

```
acfsutil plugin disable -h
acfsutil plugin disable mount_point
```

`acfsutil plugin disable -h` displays help text and exits.

Table 6-83 contains the options available with the `acfsutil plugin disable` command.

Table 6-83 Options for the acfsutil plugin disable command

Option	Description
mount_point	Specifies the directory where the file system is mounted.

`acfsutil plugin disable` disables the collection of Oracle ACFS plug-in metrics and the associated communication with a plug-in application.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-78 shows the use of the `acfsutil plugin disable` command. After monitoring has completed for the Oracle ACFS file metric data for the file system mounted on the `/humanresources`, the command in Example 6-78 disables the subsequent collection of summary data in the Oracle ACFS driver.

Example 6-78 Disabling the Oracle ACFS driver

```
# /sbin/acfsutil plugin disable /humanresource
```

See Also:

- About Using Oracle ACFS Command-Line Tools for information about running Oracle ACFS `acfsutil` commands
- Oracle ACFS Plugins for an overview of Oracle ACFS plug-ins
- Oracle ACFS Plug-in Generic Application Programming Interface for information about the Oracle ACFS plug-in application programming interface

acfsutil plugin enable

Purpose

Enables the Oracle ACFS plug-in infrastructure for an Oracle ACFS file system.

Syntax and Description

```
acfsutil plugin enable -h
acfsutil plugin enable -m metrictype [-t tag, ...]
                       [-i interval[s|m]] mount_point
```

`acfsutil plugin enable -h` displays help text and exits.

Table 6-84 contains the options available with the `acfsutil plugin enable` command.

Table 6-84 Options for the acfsutil plugin enable command

Option	Description
`-t` *tag*, ...	An optional list of tag names selecting Oracle ACFS files that you want to be monitored. The maximum number of tags is 12. The names are separated by commas. The default is to monitor all files in the Oracle ACFS file system. The tag names are managed by the Oracle ACFS tagging APIs.
`-m` *metrictype*	Selects the message payload to be transmitted from Oracle ACFS to the plug-in module with each call-out message. Oracle ACFS provides the `acfsmetric1` and `acfsmetric2` pre-defined metric types.
`-i` *interval* [s|m]	Selects posting message delivery and the posting interval. The Oracle ACFS plug-in mechanism posts a metrics message during each specified message interval. An interval is the maximum amount of time which can pass before metrics are posted. The application blocks in the API call to retrieve metrics until a posting occurs. The interval can be specified as an integer value in `s` seconds or `m` minutes. The default is minutes. The maximum value allowed for an interval is 60 minutes or 3600 seconds. If an interval is specified, then Oracle ACFS is requested to post metrics messages to the plug-in application on an interval basis. If an interval is not specified, then the application polls for metrics updates from the Oracle ACFS file system. This is the default action.
mount_point	Specifies the directory where the file system is mounted.

`acfsutil plugin enable` enables an Oracle ACFS file system for application plug-in service. Oracle ACFS plug-ins are enabled for single nodes only, either a standalone host or one or more individual nodes of a Grid Cluster. After enabled for plug-ins, the Oracle ACFS file system begins collecting metric information for either the set of specified tagged files or all files in the Oracle ACFS file system, and transmits callout messages to the plug-in application based upon the selected message payload and delivery parameters. Metrics are reset after every delivery to the application (whether through polling or posting).

In general, metrics are collected from the point of view of a user application. I/O for metadata or internal Oracle ACFS activities; such as snapshot copy-on-writing, encrypting, and replicating files; are not included. Metrics are collected for user I/O to both files in the original file system and snapshot files. Memory mapping of files that results in I/O operations is recorded in the metrics. Replication related files under the `.ACFS/repl` directory are not included in the metrics.

Oracle ACFS provides the acfsmetric1 and acfsmetric2 pre-defined metric types.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-79 shows the use of the acfsutil plugin enable command to provide additional storage usage metrics to a monitoring application.

Example 6-79 Enabling the Oracle ACFS driver for storage visibility: poll model

```
# /sbin/acfsutil plugin enable -m acfsmetric1 -t HRDATA /humanresource
```

An Oracle ACFS file system mounted on /humanresources is to be enabled for plug-in service and configured with the objective of providing additional storage usage metrics to a monitoring application. The message payload type is acfsmetric1, the files to be monitored are human resource files that have been tagged with the Oracle ACFS file tag HRDATA, and the message delivery type is poll which means that the application plug-in polls to request metric data update messages. No interval is specified with polled message delivery.

When the command in Example 6-79 completes, the Oracle ACFS file system mounted on the /humanresources mount point is enabled for plug-in communication and begins collecting Oracle ACFS file access metrics for files tagged with HRDATA. Oracle ACFS maintains metrics collecting a summary of read and write activity. Each time the associated application plug-in module polls for the metrics, Oracle ACFS sends a message with the data as defined in the ACFS_METRIC1 structure that is a summary of the selected Oracle ACFS activity since either the Oracle ACFS driver plug-in functionality was enabled or since the last delivery of metric data.

Example 6-80 shows the use of the acfsutil plugin enable command to monitor movie file updates.

Example 6-80 Enabling the Oracle ACFS driver for storage visibility: post on interval model

```
# /sbin/acfsutil plugin enable -m acfsmetric1 -t FILECONTENT -i 120s /moviemods
```

An Oracle ACFS file system mounted on /moviemods is enabled for plug-in service and is configured to record and deliver storage usage messages to a monitoring application. On an interval basis, the Oracle ACFS file system sends a wakeup on a driver event on which the application API call is waiting. The application then collects a summary of read and write activity on the files.

The message payload type is acfsmetric1 and the files to be monitored are movie files that have been tagged with the FILECONTENT tag. The message delivery type is post, which means that Oracle ACFS posts messages containing the metrics to the plug-in application for the specified interval of 120 seconds.

When the command in Example 6-80 completes, the Oracle ACFS file system mounted on the /moviemods mount point is enabled for plug-in communication and begins collecting Oracle ACFS storage visibility metrics for files tagged with FILECONTENT. On each interval, Oracle ACFS posts one or more messages to the application plug-in containing the storage visibility information.

See Also:

- About Using Oracle ACFS Command-Line Tools for information about running Oracle ACFS `acfsutil` commands
- Oracle ACFS Plugins for an overview of Oracle ACFS plug-ins
- Oracle ACFS Pre-defined Metric Types for information about the pre-defined metric types
- Oracle ACFS Plug-in Generic Application Programming Interface for information about the Oracle ACFS plug-in application programming interface
- Oracle ACFS Tagging Generic Application Programming Interface for information about the Oracle ACFS tagging application programming interface

acfsutil plugin info

Purpose

Displays information about the Oracle ACFS plug-in infrastructure for an Oracle ACFS file system.

Syntax and Description

```
acfsutil plugin info -h
acfsutil plugin info mount_point
```

`acfsutil plugin info -h` displays help text and exits.

Table 6-85 contains the options available with the `acfsutil plugin info` command.

Table 6-85 Options for the acfsutil plugin info command

Option	Description
mount_point	Specifies the directory where the file system is mounted.

`acfsutil plugin info` displays information about the state of the Oracle ACFS plug-in mechanism.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Examples

Example 6-81 shows the use of the `acfsutil plugin info` command.

Example 6-81 Displaying information about the Oracle ACFS driver

```
# /sbin/acfsutil plugin info /humanresource

Plug-in status: ENABLED
Metric type: acfsmetric1
```

```
Enabled tags: HRDATA
Delivery method: Poll
Post interval (seconds):

# /sbin/acfsutil plugin info /moviemods

Plug-in status: ENABLED
Metric type: acfsmetric1
Enabled tags: FILECONTENT
Delivery method: Post
Post interval (seconds): 120
```

See Also:

- About Using Oracle ACFS Command-Line Tools for information about running Oracle ACFS `acfsutil` commands
- Oracle ACFS Plugins for an overview of Oracle ACFS plug-ins
- Oracle ACFS Plug-in Generic Application Programming Interface for information about the Oracle ACFS plug-in application programming interface

acfsutil registry

Purpose

Adds, deletes, or displays entries in the Oracle ACFS mount registry.

Syntax and Description

```
acfsutil registry -h
acfsutil registry
acfsutil registry -a [-f] [-o moptions] [-n { nodes|all } ]
           [-u user] [-t description] device mount_point
acfsutil registry -c  {device | mount_point} [-o moptions]
           [-n { nodes|all } ] [-u user] [-t description]
acfsutil registry -d {device | mount_point}
acfsutil registry -l [device | mount_point]
acfsutil registry -m deviceacfsutil registry -r
```

`acfsutil registry -h` displays help text and exits.

If no options are specified, the command displays all the Oracle ACFS file systems that are configured for automatic start.

Table 6-86 contains the options available with the `acfsutil registry` command.

Table 6-86 Options for the acfsutil registry command

Option	Description
`-a`	Add the device, mount point, and associated *`moptions`* to the Oracle ACFS mount registry. The Oracle ADVM volume device specified must exist on the local node to add the information to the mount registry. The arguments represent all the information needed to mount the file system. At Oracle ACFS startup time these file systems are automatically mounted. Duplicate device entries are not allowed. Duplicate mount points are allowed but must be include the `-n` option for disjoint node-specific mounts.
`-c`	Changes an existing registry entry for the specified device or mount point.
`-d`	Deletes the specified device or mount point from the Oracle ACFS mount registry. If a mount point is specified and it is not unique in the Oracle ACFS mount registry, the command fails and you must run the command again specifying the device.
`-f`	This option is used in combination with `-a` when the specified device might exist in the registry and the administrator wants to replace the registration.
`-l` [*`device`* | *`mount_point`*]	Without specifying a device or mount point, the `-l` option lists all the file systems currently in the registry on a single line, with fields separated by a colon (`:`). With a device specified, lists information about the device in the Oracle ACFS registry. With a mount point specified, lists information about the mount point in the Oracle ACFS registry.
`-m` *`device`*	Lists the registered mount point, if one exists, associated with the specified device. The mount point is only returned if the Oracle ACFS file system has been registered or has been previously mounted.
`-n` { *`nodes`* | `all`}	This option, used in combination with `-a` and `-c`, specifies an optional comma-delimited list of nodes, or the `all` keyword. This specifies which nodes should attempt to mount this device on the specified mount point. Host names should be specified. The `all` keyword is functionally equivalent to not specifying a list of nodes and indicates that the device should be mounted on all nodes. Mounting on all nodes is the default behavior.
`-o` *`moptions`*	Specifies the mount options for use when mounting the file system. Valid for Linux, Solaris, and AIX. Used in combination with `-a` and `-c`. For specific `-o` *`moptions`*, refer to the `mount` command for each operating system. All the options for each operating system are available except the `all` option.
`-r`	Displays all registered file systems, not just file systems with `auto_start=always`.
`-t` *`description`*	Adds a description of the mount. Used in combination with `-a` and `-c`. For example: `-t "HR Exports share"`. This description can be seen later when looking at configured resources.

Table 6-86 (Cont.) Options for the acfsutil registry command

Option	Description
`-u` *`user`*	Specifies a user that is allowed to mount or umount (start and stop) the file system. This option, used in combination with `-a` and `-c`, is useful for creating a registered file system that can be started or stopped by someone other than root.
`mount_point`	Specifies the directory where the file system is mounted.
`device`	Specifies an Oracle ACFS device file that has been formatted.

`acfsutil registry` adds, deletes, or displays a file system from the Oracle ACFS persistent mount registry. The mount registry is a global registry that is used at Oracle ACFS startup on each node to mount all file systems specified in it. `root` or `asmadmin` privileges are required to modify the registry.

Any user is allowed to display the contents of the registry. To mount all the file systems in the Oracle ACFS mount registry, use the platform specific mount command with the `all` option. This is done automatically at Oracle ACFS startup on each node.

> **Note:**
>
> Oracle ACFS registration (`acfsutil registry`) is not supported in an Oracle Restart (standalone) configuration, which is a single-instance (non-clustered) environment.

Examples

The following examples show the use of `acfsutil registry`. The first example shows how to add the volume device file and file system mount point to the registry. The second example shows how to list the registered mount point associated with the specified volume device file. The third example shows how to delete the specified volume device file from the registry.

Example 6-82 Using the acfsutil registry command

```
$ /sbin/acfsutil registry -a /dev/asm/volume1-123 /acfsmounts/acfs1

$ /sbin/acfsutil registry -m /dev/asm/volume1-123

$ /sbin/acfsutil registry -d /dev/asm/volume1-123
```

> **See Also:**
>
> - About the Oracle ACFS Mount Registry for information about the mount registry
> - *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges

acfsutil rmfs

Purpose

Removes an Oracle ACFS file system.

Syntax and Description

```
acfsutil rmfs -h
acfsutil rmfs device
acfsutil rmfs device -b
```

`acfsutil rmfs -h` displays help text and exits.

Table 6-87 contains the options available with the `acfsutil rmfs` command.

Table 6-87 Options for the acfsutil rmfs command

Option	Description
device	Specifies an Oracle ACFS device file that has been formatted.
-b	Specifies batch mode. No user interaction is required.

You can use `acfsutil rmfs` to remove an Oracle ACFS that is dismounted. When the command is run, the superblock of the file system is disabled. `root` or `asmadmin` privileges are required to run this command.

After `acfsutil rmfs` runs successfully, the `MOUNTPATH` and `USAGE` columns in the `V$ASM_VOLUME` view are cleared for the device. The removed Oracle ACFS can be restored using `fsck`. The device can be reformatted with a new Oracle ACFS using the `mkfs` command.

Examples

The following example shows the use of `acfsutil rmfs` to remove the specified volume device file and associated file system.

Example 6-83 Using the acfsutil rmfs command

```
$ /sbin/acfsutil rmfs /dev/asm/volume1-123
```

> **See Also:**
>
> - *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges

acfsutil scrub

Purpose

Checks for and reports any inconsistencies in the metadata or file data.

Syntax and Description

```
acfsutil scrub [-h] [-n] [-m] [-q] [-e number] [-b ext1[,...]] [-z] [-a] [-p level]
               [-t level] {file | directory}
```

`acfsutil -h scrub` displays help text and exits.

The following table contains the options available with the `acfsutil scrub` command.

Table 6-88 Options for the acfsutil scrub command

Option	Description
`-n`	Specifies not to scrub the directory recursively.
`-m`	Specifies to scrub the metadata only.
`-q`	Specifies quiet mode. Only report inconsistencies to STDOUT.
`-e` *`number`*	Specifies the number of inconsistencies encountered before program terminates.
`-b` *`ext1,ext2, ...`*	Specifies not to scrub files with the specified extensions.
`-z`	Specifies not to scrub the Oracle ACFS snapshot directory.
`-a`	Specifies to scrub extensions skipped by default (`.dbf`, `.arc`).
`-p` *`level`*	Specifies I/O load level , from `0` (lowest) to `6` (highest).
`-t` *`level`*	Specifies trace level verbosity, from `0` (terse) to `6` (verbose).
`file`	Specifies the full path name of a file.
`directory`	Specifies the full path name of a directory.

On Linux and AIX systems, the `acfsutil scrub` command checks the mirror consistency of Oracle ACFS directory metadata and file user data and metadata.

For paths with metadata inconsistencies, the scrubber displays the inconsistent path.

For file paths with user data inconsistencies, the scrubber provides the following information:

- Pathname
- Byte offset of inconsistency
- Byte length of inconsistency
- ADVM volume byte offset of inconsistency

You must have `root` privileges to run `acfsutil scrub` .

Examples

The following example illustrates the use of the `acfsutil scrub` command. The first command scrubs only the metadata of the specified file. The second command scrubs the specified directory and all nested files, except for snapshots, `txt` files, and `pdf` files. The third command scrubs the specified directory and all nested files with the power level set to `5` and trace level set to `3`.

Example 6-84 Using the Oracle ACFS acfsutil scrub command

```
$ /sbin/acfsutil scrub -m /acfsmnt/textdoc1.txt
```

```
$ /sbin/acfsutil scrub -b txt,pdf -z /acfsmnt/dir1

$ /sbin/acfsutil scrub -p 5 -t 3 /acfsmnt/dir1
```

acfsutil size

Purpose

Resizes an Oracle ACFS file system.

Syntax and Description

```
acfsutil size -h
acfsutil size {[[+|-]n[K|M|G|T|P]] [-d volume_device] [-q]} |
              {[-a n[K|M|G|T] ] [-f n[K|M|G|T] ] [-x n[K|M|G|T] ]} mount_point

acfsutil size -s mount_point
acfsutil size -n mount_point
```

`acfsutil size -h` displays help text and exits.

The following table contains the options available with the `acfsutil size` command.

Table 6-89 Options for the acfsutil size command

Option	Description
[+|-]*n* K|M|G|T|P	Specifies the new size for the Oracle ACFS file system where *n* is a valid positive whole number greater than zero. The number can be preceded by a + or - to indicate the amount to add or decrease. If no operand exists, the new size is the absolute size.
–q	Provides an estimate of how much data may need to be moved to complete the command.
-a*n*	Specifies the increment by which an automatic resize increases the file system when the amount of free space falls below the amount specified by the increment. A nonnegative numeric value must be provided, and may optionally be followed by `K` (Kilobytes), `M` (Megabytes), `G` (Gigabytes), or `T` (Terabytes) to specify a value in kilobytes, megabytes, gigabytes, or terabytes. If the unit is omitted, the default is bytes. A value of 0 disables automatic resize. The minimum allowed increment is 1G.
-f*n*	Specifies the percentage free space threshold for automatic resize. If not set, the Oracle ACFS file system uses 10% as the threshold. Whole number values of 0 and 5-50 are allowed. Specifying 0 restores the Oracle ACFS file system to using the default 10%.

Table 6-89 (Cont.) Options for the acfsutil size command

Option	Description
`-x`*n*	Specifies the maximum size to which a file system automatically resizes. A nonnegative numeric value must be provided, and may optionally be followed by`K` (Kilobytes), `M` (Megabytes), `G` (Gigabytes), or `T` (Terabytes) to specify a value in kilobytes, megabytes, gigabytes, or terabytes. If the unit is omitted, the default is bytes. If this value is less than the current size of the file system then no error is returned, but automatic resize does not take place until the file system is shrunk (by manual action of the system administrator) below this value. A value of 0 indicates that no maximum size is configured, so the file system may continue to automatic resize until the underlying volume is out of space.
`–d` *volume_device*	Resizes only the specified device.
mount_point	Specifies the directory where the file system is mounted
`–s`	Enables automatic shrinking functionality.
`–n`	Disables automatic shrinking functionality.

At least one of the `+` or `–`, `–a`, or `–x` options must be specified.

The size value is rounded up based on the block size of the file system and the allocation unit of the Oracle ADVM volume device file. To determine the Oracle ADVM volume device resize increment, examine the `RESIZE_UNIT_MB` field in the `V$ASM_VOLUME` view, `Resize Unit` in the output of `asmcmd volinfo`, or `ADVM resize increment` in the output of `acfsutil info fs`.

Before issuing an `acfsutil size` command to grow or shrink a file system, you can issue `acfsutil size -q` to report how much data may be needed to be moved to complete the command. This estimation can be helpful in predicting how long the command may take to complete. Using the `-q` option does not cause any resizing of the file system.

There is no limit to the number of times a file system can be expanded for a disk group with the ADVM compatibility attribute set to `11.2.0.4` or higher.

For a disk group with the ADVM compatibility attribute set to less than `11.2.0.4`, there is a limit of 5 extents for the file system's internal storage bitmap. This causes any attempts to increase the file system to fail after it has been increased four or more times. However, if after increasing the file system four times or more times the file system size is decreased, then you may be able to increase the file system size again if the size of the increase is less than the size of the decrease. When the limit on a file system expansion has been reached, running `fsck` with the `-a` option may consolidate the internal storage bitmap, allowing future file system expansion.

`root` or users who are members of the `asmadmin` group can run this command.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges

Automatic Shrinking Option

The automatic shrinking option (`-s`) shrinks a file system by 25% if it is 50% full and it was at least 80% full since the last expansion, or since `mkfs` was run (if this is a new file system). Oracle ACFS checks once an hour to determine whether the file system meets the automatic shrink criteria. When the criteria has been detected, the automatic shrinking process begins within an hour if automatic shrinking has not already occurred that day.

At most one automatic shrink action occurs daily. The operation is run on the files system in the background.

If automatic shrinking is enabled, then `acfsutil info fs` displays `AutoShrinkEnabled` in the **flags** output.

Oracle ACFS automatic shrinking is supported on Linux.

Manual Resize Option

The `+` or `-` option grows or shrinks the mounted Oracle ACFS and its underlying Oracle ADVM storage to match the new size specified. This operation also resizes the underlying Oracle ADVM volume file to match the new length that is specified. However, the disk group must have enough free storage to accommodate any requested increase to the file system size.

You can use `acfsutil size` to extend or shrink an entire file system or only a specified device.

Reducing a file system size returns unused storage space located at the end of the file system to the disk group.

Prior to Oracle ASM release 18c, the shrinking operation would only succeed if the operation did not require existing files in the file system to be moved. For Oracle ASM release 18c and later releases, the shrink process moves files as necessary when shrinking the file system. This process of moving files can be time consuming. In addition, a subsequent command to grow the file system may also need to move data, which may be time-consuming.

> **Note:**
>
> - An attempt to extend or shrink the file system may partially succeed, resulting in the file system having a larger or smaller size than before the `acfsutil size` command was issued. However, because of errors or environmental changes, the new size may not be the requested size. If a grow or shrink operation fails on a node for any reason, an error message is written to the `USM` event log and the `OKS` log.
> - A shrink of the accelerator volume does not cause the migration of metadata from the accelerator volume to the primary volume. The process fails if there is too much data on the accelerator.
> - A shrink command moves files out of the area of the volume that is to be removed and into a different part of the volume. This moving of files could change the fragmentation of these files. It could result in their becoming more fragmented than they were prior to the file system shrink operation. It could also result in their becoming less fragmented.

Automatic Resize Option

The automatic resize option (`-a` option) specifies an increment by which an Oracle ACFS file system grows automatically if the amount of available free space in the file system falls below the amount specified by the increment. You can also specify the maximum size (`-x` option) allowed when using the automatic resize option. The output of the `acfsutil info fs` command displays the automatic resize increment and maximum amounts. If only the `-x` option is provided without the `-a` option (and no automatic resize increment has previously been specified) then a warning is issued if the increment is still zero. If the increment had previously been set by an earlier invocation of the command, then the command succeeds. Automatic resize requires ADVM compatibility attribute set to `12.2` or higher.

If an automatic resize fails on a node for any reason, automatic resizing on the node is temporarily disabled. An error message is written to the `USM` event log and the `OKS` log. Automatic resizing may still take place on another node; however, if the failure is due to the underlying volume being out of free space then all nodes quickly disable automatic resize. The file system does continue to periodically try to automatic resize, at increasingly longer intervals. After a resize succeeds, such as when more space is added to the underlying Oracle ASM disk group, then automatic resize becomes re-enabled on the node. Note that other nodes may remain in the error state even after one node successfully auto-resizes the file system. Automatic resize can be manually re-enabled by either re-mounting the file system or running the acfsutil size command again with the `-a` or `-x` option to update automatic resize settings. When a file system has grown to the maximum size specified with `acfsutil size -x`, automatic resize remains enabled but does not grow the file system any further until the maximum is increased.

The `-d` argument is not compatible with the `-a` and `-x` automatic resize arguments. Unless a volume device is specified with `-d`, it is assumed that the caller is specifying an amount of storage that applies to the primary volume, which represents the size of the file system. When growing a file system with an accelerator volume, the `acfsutil size` command determines if the accelerator size needs to be increased. If the accelerator must be larger and cannot be increased, the command fails and the primary volume remains at the same size. When shrinking a file system, only the primary volume is shrunk. If a volume device is specified with the `-d` option, then the size specified applies to either the primary volume or the accelerator volume, whichever is specified, and only that volume device is changed. The command warns the user if the resulting accelerator volume size is not large enough to accommodate the primary volume size.

When using an accelerator volume and automatic resize, the increment specified with the automatic resize `-a` argument applies to the primary volume; however, both the primary volume and the accelerator volume are monitored. Automatic resize grows the primary volume by the specified increment, or greater if needed. Automatic resize monitors and grows the accelerator volume as needed as well by smaller increments. When automatic resize needs to grow the primary volume, it first attempts to grow the accelerator if insufficient space remains on the accelerator to accommodate the new growth, and if that fails the primary volume remains at the same size.

In addition to being triggered because the free space falls below the configured automatic resize increment, an automatic resize also takes place under other conditions, such as:

- If the free space in the file system falls below the free space threshold of the current file system size or 5 GB, whichever is smaller. The free space threshold is 10% if the percentage has not been configured for the Oracle ACFS filesystem.
- If a copy-on-write cannot find enough contiguous space in the file system to allocate new storage for the write.
- If the defragger cannot find enough contiguous space to allocate in order to defragment 8 MB worth of extents in a file.

These conditions protect against a large file system having enough total free space so that an automatic resize appears unnecessary, but the space is so fragmented that applications cannot use it.

When a non-sparse file is resized so that it grows by more than the configured auto-resize interval, the file system automatic resizes by a multiple of the automatic resize increment so that the resize request can be satisfied.

Examples

The following example shows the use of `acfsutil size`. This example increases the primary device file of `/acfsmounts/acfs1` file system by 500 MB.

Example 6-85 Using the acfsutil size command

```
$ /sbin/acfsutil size +500M /acfsmounts/acfs1
```

acfsutil thaw

Purpose

Resumes activity after an `acfsutil freeze` command has been issued on a file system.

Syntax and Description

```
acfsutil thaw -h
acfsutil thaw mount_point
```

`acfsutil thaw -h` displays help text and exits.

The following table contains the options available with the `acfsutil thaw` command.

Table 6-90 Options for the acfsutil thaw command

Option	Description
mount_point	Specifies the directory where the file system is mounted.

You can use the `acfsutil thaw` command to resume activity after the `acfsutil freeze` command has been issued to temporarily halt modification activity on a file system. For information about `acfsutil freeze`, refer to acfsutil freeze.

Examples

The following example shows the use of the `acfsutil thaw` command.

Example 6-86 Using the acfsutil thaw command

```
$ /sbin/acfsutil thaw /acfsmounts/acfs1
```

acfsutil version

Purpose

Displays Oracle ACFS version information.

Syntax and Description

```
acfsutil version -h
acfsutil version -v
```

`acfsutil version -h` displays help text and exits.

The following table contains the options available with the `acfsutil version` command.

Table 6-91 Options for the acfsutil version command

Option	Description
`—v`	Displays verbose version information about Oracle ACFS.

Administrator privileges are required to use this command or you must be a member of the Oracle ASM administrator group.

Example 6-87 Using acfsutil version

In the first example, basic information is displayed about the Oracle ACFS release version. In the second example, detailed version information is displayed about the installed Oracle ACFS components.

```
$ /sbin/acfsutil version
acfsutil version: 19.0.0.0.0

$ /sbin/acfsutil version -v
Kernel:
    Build version: 19.0.0.0.0
    Build full version: 19.2.0.0.0
```

```
    Build hash:    9728583571
    Bug numbers:   26947218
Commands:
    Build version: 19.0.0.0.0
    Build full version: 19.2.0.0.0
    Build hash:    9728583571
    Bug numbers:   26947218
```

advmutil canonical

Purpose

`advmutil canonical` displays the canonical name of the specified Oracle ADVM device name.

Syntax and Description

```
advmutil -h
advmutil canonical volume_device
```

`advmutil -h` displays help text and exits.

Table 6-92 contains the options available with the `advmutil canonical` command.

Table 6-92 Options for the advmutil canonical command

Option	Description
volume_device	Specifies a string identifying an Oracle ADVM volume device.

There are several different formats that can identify an Oracle ADVM volume device, but a normalized, unambiguous (canonical) name should be used when the volume device name is used with other commands such as `SRVCTL`.

The `advmutil canonical` command would return the canonical name that another utility would recognize without having to strip off extra characters. The command would most likely be used in a script.

advmutil volinfo

Purpose

`advmutil volinfo` displays information about Oracle ADVM volume devices.

Syntax and Description

```
advmutil -h
advmutil volinfo [-l][-L] [volume_device]
```

`advmutil -h` displays help text and exits.

Table 6-93 contains the options available with the `advmutil volinfo` command.

Table 6-93 Options for the advmutil volinfo command

Option	Description
volume_device	Specifies an optional volume device name.
-l	Separates the Oracle ADVM volume device information by field descriptions and colons on one line.
-L	Separates the Oracle ADVM volume device information by spaces on one line.

advmutil volinfo displays information about Oracle ADVM volume devices in a list format by default. The -l option on Linux formats the display into a colon-separated string with field descriptions. The -L option on Linux formats the display into a space-separated string in a format suitable for scripting.

Examples

The first example displays information about an Oracle ADVM volume device, using the advmutil volinfo command with the volume device name. The second example displays information about the volume device using the -l option. The third example displays information about the volume device using the -L option.

Example 6-88 Using advmutil volinfo

```
$ /sbin/advmutil volinfo /dev/asm/volume1-123
Device : /dev/asm/volume1-228
Interface Version: 1
Size (MB): 256
Resize Increment (MB): 32
Redundancy: mirror
Stripe Columns: 4
Stripe Width (KB): 128
Disk Group: DATA
Volume: VOLUME1
Compatible.advm : 11.2.0.0.0

$ /sbin/advmutil volinfo -l /dev/asm/volume1-228
Device : /dev/asm/volume1-228 : Interface Version : 1 : Size (MB) : 256 :
Resize Increment (MB) : 32 : Redundancy : mirror : Stripe Columns : 4 :
Stripe Width (KB) : 128 : Disk Group : DATA : Volume : VOLUME1 :
Compatible.advm : 11.2.0.0.0

$ /sbin/advmutil volinfo -L /dev/asm/volume1-228
/dev/asm/volume1-228  1  256 32  mirror  4  128  DATA  VOLUME1  11.2.0.0.0
```

Part II
Advanced Topics

Advanced topics contains the following chapter:

- Understanding Oracle ACFS Advanced Topics

7 Understanding Oracle ACFS Advanced Topics

Oracle ACFS advanced topics include discussions about more complex administrative issues.

This appendix discusses Oracle Advanced Cluster File System (Oracle ACFS) advanced topics, including limits, advanced administration, troubleshooting, and patching.

> **See Also:**
>
> Articles available at My Oracle Support (`https://support.oracle.com`) for information about Oracle ACFS and Oracle ADVM.

This appendix contains the following topics:

- Limits of Oracle ACFS
- Limits of Oracle ADVM
- How to Clone a Full Database (non-CDB or CDB) with ACFS Snapshots
- Steps to perform an RMAN sparse backup and restore of a PDB using ACFS fshares
- Steps to perform an RMAN sparse backup and restore of a PDB using an ACFS snapshot
- Oracle ACFS Loopback Support
- Oracle ACFS Drivers Resource Management
- Oracle ACFS Registry Resource Management
- Oracle ACFS File System Resource Management
- Oracle ACFS and Oracle Restart
- Oracle ACFS Driver Commands
- Oracle ACFS Plug-in Generic Application Programming Interface
- Oracle ACFS Tagging Generic Application Programming Interface
- Understanding Oracle ACFS I/O Failure Console Messages
- Configuring Oracle ACFS Snapshot-Based Replication
- Oracle Patching and Oracle ACFS

For an overview of Oracle ACFS, see Introducing Oracle ACFS and Oracle ADVM .

Limits of Oracle ACFS

The limits of Oracle ACFS are discussed in this section.

The topics contained in this section are:

- Oracle ACFS Disk Space Usage
- Oracle ACFS Error Handling
- Oracle ACFS and NFS

Note:

Oracle ACFS does not support hard links on directories.

Oracle ACFS Disk Space Usage

Oracle ACFS supports 256 mounts on 64-bit systems. However, more file systems can be mounted if there is adequate memory.

Oracle ACFS supports 2^40 (1 trillion) files in a file system. More than 4 billion files have been tested. There is no absolute limit to the number of directories in a file system; the limit is based on hardware resources.

Oracle ACFS preallocates large user files to improve performance when writing data. This storage is not returned when the file is closed, but it is returned when the file is deleted. Oracle ACFS also allocates local metadata files as nodes mount the file system for the first time. This storage is approximately 64-128 megabytes per node.

Oracle ACFS also keeps local bitmaps available to reduce contention on the global storage bitmap when searching for free space. This disk space is reported as `in use` by tools such as the Linux `df` command even though some space may not actually be allocated yet. This local storage pool can be as large as 128 megabytes per node and can allow space allocations to succeed, even though commands, such as `df`, report less space available than what is being allocated.

The maximum sizes that can be allocated to an Oracle ACFS file system are shown in Table 7-1. The storage limits for Oracle ACFS and Oracle ASM are dependent on disk group compatibility attributes.

Table 7-1 Maximum file sizes for Oracle ACFS file systems/Oracle ADVM volumes

Redundancy	Disk Group with COMPATIBLE.ASM < 12.2.0.1	Disk Group with COMPATIBLE.ASM >= 12.2.0.1
External	128 TB	128 TB
Normal	64 TB	128 TB
High	42.6 TB	128 TB

> **Note:**
>
> Customers with `Compatible.ASM >= 19.0` wanting a 1PB ADVM volume can set the following diskgroup attribute to '`64`'. This parameter only effects newly created volumes:
>
> ```
> SQL> alter diskgroup my_dg set attribute 'advm_extent_size_mb'='64';
> ```

> **See Also:**
>
> - *Oracle Automatic Storage Management Administrator's Guide* for information about file size limits and disk group compatibility settings
> - *Oracle Automatic Storage Management Administrator's Guide* for information about storage limits for Oracle ASM files and disk groups

Oracle ACFS Error Handling

Oracle ASM instance failure or forced shutdown while Oracle ACFS or another file system is using an Oracle ADVM volume results in I/O failures. The volumes must be closed and re-opened to access the volume again. This requires dismounting any file systems that were mounted when the local Oracle ASM instance failed. After the instance is restarted, the corresponding disk group must be mounted with the volume enabled followed by a remount of the file system. See "Deregistering, Dismounting, and Disabling Volumes and Oracle ACFS File Systems".

If any file systems are currently mounted on Oracle ADVM volume files, the `SHUTDOWN ABORT` command should not be used to terminate the Oracle ASM instance without first dismounting those file systems. Otherwise, applications encounter I/O errors and Oracle ACFS user data and metadata being written at the time of the termination may not be flushed to storage before the Oracle ASM storage is fenced. If there is not time to permit the file system to dismount, then you should run two `sync` (1) commands to flush cached file system data and metadata to persistent storage before issuing the `SHUTDOWN ABORT` operation.

Oracle ACFS does not interrupt the operating system environment when a metadata write fails, whether due to Oracle ASM instance failure or storage failure. Instead, Oracle ACFS isolates errors to a specific file system, putting it in an offline error state. The only operation that succeeds on that node for that file system from that point forward is a dismount operation. Another node recovers any outstanding metadata transactions, assuming it can write the metadata out to the storage. It is possible to remount the file system on the offlined node after the I/O condition is resolved.

It might not be possible for an administrator to dismount a file system while it is in the offline error state if there are processes referencing the file system, such as a directory of the file system being the current working directory for a process. To dismount the file system in this case it would be necessary to identify all processes on that node with references to files and directories on the file system and cause them to exit. The Linux `fuser` or `lsof` commands list information about processes and open files.

If Oracle ACFS detects inconsistent file metadata returned from a read operation, based on checksum or expected type comparisons, Oracle ACFS takes the appropriate action to isolate the affected file system components and generate a notification that `fsck` should be run as soon as possible. Each time the file system is mounted a notification is generated with a system event logger message until `fsck` is run.

Oracle ACFS and NFS

When exporting file systems through NFS on Linux, use the `-fsid=num` exports option. This option forces the file system identification portion of the file handle used to communicate with NFS clients to be the specified number instead of a number derived from the major and minor number of the block device on which the file system is mounted. You can use any 32-bit number for `num`, but it must be unique among all the exported file systems. In addition, `num` must be unique among members of the cluster and must be the same `num` on each member of the cluster for a given file system. This is needed because Oracle ADVM block device major numbers are not guaranteed to be the same across restarts of the same node or across different nodes in the cluster.

> **Note:**
>
> Oracle ASM Dynamic Volume Manager (Oracle ADVM) volumes and Oracle Advanced Cluster File System (Oracle ACFS) file systems are currently not supported on disk groups that have been created from NFS or Common Internet File System (CIFS) files. However, Oracle ACFS file systems may be exported as NFS or CIFS file systems to network clients in some cases. Samba/CIFS clients on Windows cannot use ACLs when interfacing with Oracle ACFS Linux, Solaris, or AIX servers.

When using High Availability NFS for Grid Home Clusters (HANFS), HANFS automatically handles the situation described in the previous paragraph. For information about HANFS, refer to "High Availability Network File Storage for Oracle Grid Infrastructure".

Limits of Oracle ADVM

The limits of Oracle ADVM are discussed in this topic.

The default configuration for an Oracle ADVM volume is 8 columns and a 1 MB stripe width. The default volume extent size (64 MB).

Setting the number of columns on an Oracle ADVM dynamic volume to `1` effectively turns off striping for the Oracle ADVM volume. Setting the columns to 8 (the default) is recommended to achieve optimal performance with database data files and other files.

On Linux platforms Oracle ASM Dynamic Volume Manager (Oracle ADVM) volume devices are created as block devices regardless of the configuration of the underlying storage in the Oracle ASM disk group. Do not use `raw (8)` to map Oracle ADVM volume block devices into raw volume devices.

For information about ASMCMD commands to manage Oracle ADVM volumes, refer to Managing Oracle ADVM with ASMCMD.

How to Clone a Full Database (non-CDB or CDB) with ACFS Snapshots

ACFS snapshots are sparse, point-in-time copies of the filesystem and this can be used to create full DB clones as well as clones of PDBs using PDB snapshot cloning when the DB is on ACFS (User Interface for PDB Cloning). ACFS snapshots can be used in test and development environments to create quick and space efficient clones of a test master. This section explains the steps required to create a full DB clone using ACFS snaps with an example.

Test setup: We have a single test master CDB called `SOURCE` that will be cloned. The CDB has ten PDBs by name `sourcepdb[1-10]` and each of them is loaded with a OLTP schema. This is a Real Application Cluster database (RAC) and the instances are running on both the nodes. The datafiles, redo logs and controlfiles are stored in ACFS mounted at "`/mnt/dbvol`". This filesystem is created on `DATA` diskgroup. Recovery logs and archive logs are stored on a filesystem mounted at "`/mnt/rvol`" that is created on top of the `RECO` diskgroup. Note that ACFS snaps will be contained within the filesystem and can be accessed through the same mountpoint.

Oracle highly recommends periodically creating backups of the test master database to provide a recovery method in case of issues.

For more detailed information regarding the configuration of ACFS Spanpshots on Exadata, see Setting up Oracle Exadata Storage Snapshots

For more information about different ACFS Snapshot use cases, please see My Oracle Support (MOS) note Oracle ACFS Snapshot Use Cases on Exadata (Doc ID 2761360.1).

Steps to Create and Use the Clone

1. Make sure that the database is in archive log mode:

```
SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            /mnt/rvol/archive
Oldest online log sequence     488
Next log sequence to archive   489
Current log sequence           489
SQL>
```

2. Take a SQL trace backup of the control file for the test master database, the script generated will be used as the basis for creating controlfiles for each snapshot of this test master. This backup is created in the location specified in the AS clause. Specifying the RESETLOGS argument ensures only the RESETLOGS version of the create controlfile statement is generated in the trace file.

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS '/tmp/
source1_ctlfile_bkup.sql' RESETLOGS;
Database altered.

[oracle@machine ~]$ ls -lrt /tmp/source1_ctlfile_bkup.sql
```

```
-rw-r----- 1 oracle oinstall 32874 Jun 16 04:01 /tmp/
source1_ctlfile_bkup.sql
```

3. Create a pfile from the spfile if the DB instance currently uses an spfile:

```
SQL> CREATE PFILE='/tmp/clone_pfile.ora' FROM SPFILE;

File created.

SQL>
```

The pfile is saved to the location specified by the `PFILE=` clause.

4. Stop the test master database and create RW snap of the datafile filesystem:

```
[oracle@machine ~]$ srvctl stop database -db source
[oracle@machine ~]$ /sbin/acfsutil  snap create -w clone /mnt/dbvol/
acfsutil snap create: Snapshot operation is complete.
[oracle@machine ~]$ /sbin/acfsutil snap info clone /mnt/dbvol/
snapshot name:               clone
snapshot location:           /mnt/dbvol/.ACFS/snaps/clone
RO snapshot or RW snapshot:  RW
parent name:                 /mnt/dbvol/
snapshot creation time:      Tue Jun 16 04:09:10 2020
file entry table allocation: 17170432   (  16.38 MB )
storage added to snapshot:   17170432   (  16.38 MB )
$
```

5. Modify the trace file generated in step 2.

The following block in the backed up trace file:

```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "SOURCE" RESETLOGS  ARCHIVELOG
    MAXLOGFILES 192
    MAXLOGMEMBERS 3
    MAXDATAFILES 1024
    MAXINSTANCES 32
    MAXLOGHISTORY 2254
LOGFILE
  GROUP 5 (
    '/mnt/dbvol/oradata/SOURCE/onlinelog/o1_mf_5_hfh27gq1_.log',
    '/mnt/rvol/fast_recovery_area/SOURCE/onlinelog/
o1_mf_5_hfh27jgv_.log'
  ) SIZE 2048M BLOCKSIZE 512,
  GROUP 6 (
    '/mnt/dbvol/oradata/SOURCE/onlinelog/o1_mf_6_hfh27ymp_.log',
    '/mnt/rvol/fast_recovery_area/SOURCE/onlinelog/
o1_mf_6_hfh280cb_.log'
  ) SIZE 2048M BLOCKSIZE 512
```

Needs to be changed to:

```
CREATE CONTROLFILE  DATABASE "SOURCE" RESETLOGS ARCHIVELOG
    MAXLOGFILES 192
    MAXLOGMEMBERS 3
    MAXDATAFILES 1024
    MAXINSTANCES 32
    MAXLOGHISTORY 2254
LOGFILE
  GROUP 5  SIZE 2048M BLOCKSIZE 512,
  GROUP 6  SIZE 2048M BLOCKSIZE 512
```

Note that you will be creating new redo log files and hence the `db_create_online_log_dest_1` parameter will be picked by the DB instance to create the target files. No need to name them in the above statements, let Oracle use Oracle Managed Files (OMF) for naming.

The `DATABASE` name specified in the `CREATE CONTROLFILE` statement must match the database name of the test master. The `DB_UNIQUE_NAME` parameter will be used to make this into a distinct database from the test master.

The file names in the `DATAFILE` block need their directory structures changed to point to the files created in the snapshot.

NOTE: Failure to change these file names could allow the snapshot to start using the parent files directly there by corrupting them. Ensure all the datafile names are modified to point to the snapshot location.

A file named

`/mnt/dbvol/oradata/source/datafile/o1_mf_system_hkp8q4l0_.dbf`
would need to be renamed to

`/mnt/dbvol/.ACFS/snaps/clone/oradata/source/datafile/o1_mf_system_hkp8q4l0_.dbf`
Ensure all snapshot datafiles exist prior to attempting to create the controlfile.

Since the TEMP files are all going to be created new, the corresponding statements need to change.

- Old TEMP file clauses:

  ```
  ALTER TABLESPACE TEMP ADD TEMPFILE '/mnt/dbvol/oradata/SOURCE/
  datafile/o1_mf_temp_hf9ck61d_.tmp'
       SIZE 1377M REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE 32767M;
  ALTER SESSION SET CONTAINER = "PDB$SEED";
  ALTER TABLESPACE TEMP ADD TEMPFILE '/mnt/dbvol/oradata/SOURCE/
  datafile/temp012020-06-01_03-40-57-522-AM.dbf'
       SIZE 37748736  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE 32767M;
  ALTER SESSION SET CONTAINER = "SOURCEPDB1";
  ALTER TABLESPACE TEMP ADD TEMPFILE '/mnt/dbvol/oradata/SOURCE/
  A70228A28B4EB481E053DFB2980A90DF/datafile/o1_mf_temp_hf9f8yxz_.dbf'
       SIZE 37748736  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE 32767M;
  ALTER SESSION SET CONTAINER = "SOURCEPDB2";
  ALTER TABLESPACE TEMP ADD TEMPFILE '/mnt/dbvol/oradata/SOURCE/
  A70229825B448EF3E053E0B2980AA4D4/datafile/o1_mf_temp_hf9f9goq_.dbf'
       SIZE 37748736  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE 32767M;
  ...........................................
  ```

```
..........................................
..........................................
```

- New TEMP file clauses:

```
ALTER TABLESPACE TEMP ADD TEMPFILE SIZE 1377M AUTOEXTEND ON NEXT
655360 MAXSIZE 32767M;
ALTER SESSION SET CONTAINER = "PDB$SEED";
ALTER TABLESPACE TEMP ADD TEMPFILE SIZE 37748736   AUTOEXTEND ON
NEXT 655360  MAXSIZE 32767M;
ALTER SESSION SET CONTAINER = "SOURCEPDB1";
ALTER TABLESPACE TEMP ADD TEMPFILE SIZE 37748736   AUTOEXTEND ON
NEXT 655360  MAXSIZE 32767M;
ALTER SESSION SET CONTAINER = "SOURCEPDB2";
ALTER TABLESPACE TEMP ADD TEMPFILE SIZE 37748736   AUTOEXTEND ON
NEXT 655360  MAXSIZE 32767M;
..........................................
..........................................
..........................................
```

> **Note:**
>
> OMF is being used for naming the temp files, there is no need to specify a file name.

- Other changes:
  – Remove the `REUSE` statement throughout.
  – Remove the `RECOVER DATABASE` command.
  – Remove commands to enable block change tracking.

Following is an example of a full create controlfile statement:

```
STARTUP NOMOUNT
CREATE CONTROLFILE  DATABASE "SOURCE" RESETLOGS  ARCHIVELOG
    MAXLOGFILES 1024
    MAXLOGMEMBERS 5
    MAXDATAFILES 32767
    MAXINSTANCES 32
    MAXLOGHISTORY 33012
LOGFILE
GROUP 1 SIZE 4096M BLOCKSIZE 512,
GROUP 2 SIZE 4096M BLOCKSIZE 512,
GROUP 5 SIZE 4096M BLOCKSIZE 512,
GROUP 6 SIZE 4096M BLOCKSIZE 512
'/mnt/dbvol/.ACFS/snaps/clone/oradata/datafile/
o1_mf_system_hkpjm4lh_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/
AAFDB0C85C03151AE053FE5E1F0AA61D/datafile/
o1_mf_system_hkpjm4rf_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/datafile/
o1_mf_sysaux_hkpjm56t_.dbf',
```

```
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AAFDB0C85C03151AE053FE5E1F0AA61D/
datafile/o1_mf_sysaux_hkpjm6x4_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/datafile/
o1_mf_undotbs1_hkpjm6x0_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AAFDB0C85C03151AE053FE5E1F0AA61D/
datafile/o1_mf_undotbs1_hkpjm6w5_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/datafile/
o1_mf_undotbs2_hkpjm6tx_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/datafile/
o1_mf_users_hkpjm722_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AAFE1E09C8245CC2E053FE5E1F0A5632/
datafile/o1_mf_system_hkpjmqds_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AAFE1E09C8245CC2E053FE5E1F0A5632/
datafile/o1_mf_sysaux_hkpjmqoc_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AAFE1E09C8245CC2E053FE5E1F0A5632/
datafile/o1_mf_undotbs1_hkpjmtl7_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AAFE1E09C8245CC2E053FE5E1F0A5632/
datafile/o1_mf_undo_2_hkpjn03d_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AAFE1E09C8245CC2E053FE5E1F0A5632/
datafile/o1_mf_users_hkpjn16b_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AAFE1E09C8245CC2E053FE5E1F0A5632/
datafile/o1_mf_soe_hkpjn1ft_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB1122BDE19EF370E053FE5E1F0A4E77/
datafile/o1_mf_system_hkpjn8h4_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB1122BDE19EF370E053FE5E1F0A4E77/
datafile/o1_mf_sysaux_hkpjn8nt_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB1122BDE19EF370E053FE5E1F0A4E77/
datafile/o1_mf_undotbs1_hkpjn8qq_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB1122BDE19EF370E053FE5E1F0A4E77/
datafile/o1_mf_undo_2_hkpjnf2p_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB1122BDE19EF370E053FE5E1F0A4E77/
datafile/o1_mf_users_hkpjnglw_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB1122BDE19EF370E053FE5E1F0A4E77/
datafile/o1_mf_soe_hkpjnkm6_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB112685643A0B77E053FE5E1F0A7BDD/
datafile/o1_mf_system_hkpjnl8v_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB112685643A0B77E053FE5E1F0A7BDD/
datafile/o1_mf_sysaux_hkpjnmfm_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB112685643A0B77E053FE5E1F0A7BDD/
datafile/o1_mf_undotbs1_hkpjnqjz_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB112685643A0B77E053FE5E1F0A7BDD/
datafile/o1_mf_undo_2_hkpjnqk7_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB112685643A0B77E053FE5E1F0A7BDD/
datafile/o1_mf_users_hkpjnygh_.dbf',
'/mnt/dbvol/.ACFS/snaps/clone/oradata/AB112685643A0B77E053FE5E1F0A7BDD/
datafile/o1_mf_soe_hkpjnyv6_.dbf'
CHARACTER SET AL32UTF8;

--
--
-- Create log files for threads other than thread one.
ALTER DATABASE ADD LOGFILE THREAD 2
GROUP 3 SIZE 4096M BLOCKSIZE 512 ,
GROUP 4 SIZE 4096M BLOCKSIZE 512 ,
GROUP 7 SIZE 4096M BLOCKSIZE 512 ,
```

```
GROUP 8 SIZE 4096M BLOCKSIZE 512 ;
-- Database can now be opened zeroing the online logs.
ALTER DATABASE OPEN RESETLOGS;
-- Open all the PDBs.
ALTER PLUGGABLE DATABASE ALL OPEN;
-- Commands to add tempfiles to temporary tablespaces.
-- Online tempfiles have complete space information.
-- Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE
     SIZE 32768M  AUTOEXTEND ON NEXT 16384M MAXSIZE 524288M;
ALTER SESSION SET CONTAINER = PDB$SEED;
ALTER TABLESPACE TEMP ADD TEMPFILE
     SIZE 32767M  AUTOEXTEND OFF;
ALTER SESSION SET CONTAINER = PDB201;
ALTER TABLESPACE TEMP ADD TEMPFILE
     SIZE 32768M  AUTOEXTEND ON NEXT 16384M MAXSIZE 524288M;
ALTER SESSION SET CONTAINER = PDB202;
ALTER TABLESPACE TEMP ADD TEMPFILE
     SIZE 32768M  AUTOEXTEND ON NEXT 16384M MAXSIZE 524288M;
ALTER SESSION SET CONTAINER = PDB203;
ALTER TABLESPACE TEMP ADD TEMPFILE
     SIZE 32768M  AUTOEXTEND ON NEXT 16384M MAXSIZE 524288M;
ALTER SESSION SET CONTAINER = CDB$ROOT;
-- End of tempfile additions.
shutdown immediate
exit
```

6. Edit the PFILE backed up in step 3 above. Note that the `db_name` will remain the same for the database clone as well but `db_unique_name` needs to be different. The RAC instances too should derive their names from `db_unique_name`. Make sure that the file paths are pointing the corresponding locations inside the snaps. The required changes are summarized below:
   - New parameter to be added:

     ```
     *.db_unique_name='clone'
     ```

   - Change existing instance names to new ones:
     – Old:

       ```
       source1.instance_number=1
       source2.instance_number=2
       source1.thread=1
       source2.thread=2
       source1.undo_tablespace='UNDOTBS1'
       source2.undo_tablespace='UNDOTBS2'
       ```

     – New:

       ```
       clone1.instance_number=1
       clone2.instance_number=2
       clone1.thread=1
       clone2.thread=2
       ```

```
clone1.undo_tablespace='UNDOTBS1'
clone2.undo_tablespace='UNDOTBS2'
```

Make similar changes to all occurrences of instance names.

Note that in the case of RAC databases, the clone instance should first be started in exclusive mode and then restarted in clustered mode after the clone has been created. In such cases, comment out the `CLUSTER_DATABASE` parameter in the PFILE:

```
*.cluster_database=true
```

Change parameters in the pfile to account for new datafile, archive log and controlfile locations in the snapshot:

- Old:

  ```
      *.db_create_file_dest='/mnt/dbvol/oradata'
      *.log_archive_dest_1='LOCATION=/mnt/rvol/archive1'
      *.control_files='/mnt/dbvol/oradata/SOURCE/controlfile/
  o1_mf_hf9cjtv0_.ctl','/mnt/rvol/fast_recovery_area/SOURCE/controlfile/
  o1_mf_hf9cjtxr_.ctl'
  ```

- New:

  ```
      *.db_create_file_dest='/mnt/dbvol/.ACFS/snaps/clone/oradata'
      *.log_archive_dest_1='LOCATION=/mnt/rvol/archive1'
      *.control_files='/mnt/dbvol/.ACFS/snaps/clone/oradata1/
  ctrl_1_.ctl'
  ```

Give a new location for the audit trace files for the clone:

```
    *.audit_file_dest='/u01/app/oracle/admin/clone/adump'
```

Create the above directory if not present already.

```
$ mkdir -p /u01/app/oracle/admin/clone/adump
```

7. Copy the modified PFILE from the `$ORACLE_HOME/dbs` directory and rename it as an `init.ora` file to match the local clone database instance name. For example: `$ cp /tmp/clone_pfile.ora $ORACLE_HOME/dbs/initclone1.ora`
8. Copy the password file from the source database to a new file name for the clone:

   ```
   $ cp <Directory where password file is stored>/orapwsource /mnt/
   dbvol/.ACFS/snaps/clone/orapwclone
   ```

9. Set `ORACLE_SID` to point to the new SID and run the SQL script that you generated in step 5.

   ```
   [oracle@machine dbclone]$ export ORACLE_SID=clone1
   [oracle@machine dbclone]$ sqlplus / as sysdba

   SQL*Plus: Release 19.0.0.0.0 - Production on Tue Jun 16 04:50:21 2020
   ```

```
Version 19.7.0.0.0

Copyright (c) 1982, 2020, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> spool startup.log
SQL> @ctlfile
ORACLE instance started.

Total System Global Area 8.1068E+10 bytes
Fixed Size                  30383424 bytes
Variable Size             1.1006E+10 bytes
Database Buffers          6.9793E+10 bytes
Redo Buffers               238051328 bytes

Control file created.
System altered.
Database altered.
Pluggable database altered.
Tablespace altered.
Session altered.
Tablespace altered.
Session altered.
Tablespace altered.
Session altered.
Tablespace altered.
.........................
.........................
.........................
.........................
```

10. The DB is now mounted in exclusive mode on the first node. If the test master database is a RAC database and you wish to enable RAC on the snapshot, shutdown the database and restart the instance with "`cluster_database=true`" parameter set in the pfile. Create an spfile and store it in the the ACFS database file system. Once this is done, we can start both the RAC instances and have the DB running on all the nodes.

```
SQL> create spfile='/mnt/dbvol/.ACFS/snaps/clone/oradata/
spfileclone.ora' from pfile="<ORACLE_HOME>/dbs/initclone1.ora'
```

Add the database and its properties to Cluster Ready Services (CRS)

```
$ srvctl add database -db clone -oraclehome $ORACLE_HOME -dbtype
RAC -spfile /mnt/dbvol/.ACFS/snaps/clone/oradata/spfileclone.ora -
pwfile /mnt/dbvol/.ACFS/snaps/clone/orapwclone -dbname source -
acfspath "/mnt/dbvol,/mnt/rvol" -policy AUTOMATIC -role PRIMARY
```

Add the instances for each node/instance of the cluster:

```
$ srvctl add instance -db clone -instance clone -node <node
hostname>
```

11. Check the archive logs are using the new destination:

```
SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            /mnt/rvol/archive1
Oldest online log sequence     489
Next log sequence to archive   490
Current log sequence           490
SQL>
```

Check if clustered database mode is used:

```
SQL> SHOW PARAMETER cluster_database

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
cluster_database                     boolean     TRUE
cluster_database_instances           integer     2
SQL>
```

For RAC environments, be sure to use the srvctl start database and stop database commands as appropriate to help ensure the database is configured correctly.

12. Make sure that all PDBs in the clone can be opened and used.

```
SQL> SELECT open_mode FROM v$pdbs;

OPEN_MODE
----------
READ ONLY
READ WRITE
READ WRITE
READ WRITE
READ WRITE
READ WRITE
READ WRITE
READ WRITE
READ WRITE
READ WRITE
READ WRITE

11 rows selected.


SQL> ALTER SESSION SET CONTAINER=SOURCEPDB1;

Session altered.

SQL> SELECT name FROM v$tablespace;

NAME
------------------------------
```

```
SYSTEM
SYSAUX
UNDOTBS1
UNDO_2
USERS
USER_TBS
LOB_TBS
FLASHBACK_TBS
TEMP
USER_TEMP

10 rows selected.

SQL>
```

The DB clone is now fully ready and can be used for running tests. While the clone is up and running, additional clones can be created from this or from the test master. This way, any number of DB clones can be easily created using ACFS snapshot technology.
It is also recommended to add services for these new DB clones similar to what is used for the parent (or as needed by your applications).

Steps to perform an RMAN sparse backup and restore of a PDB using ACFS fshares

1. Assume there are two PDBs SOURCE and CLONE. CLONE is a snapshot copy of SOURCE. The snapshot copy is a thin clone using ACFS fshares.

```
SQL> create pluggable database SOURCE admin user foo identified by
bar;

Pluggable database created.

SQL> select d.con_id, p.name pdb_name, p.open_mode, d.file#, d.name
file_name from v$dbfile d, v$pdbs p where d.con_id=p.con_id order
by d.con_id,p.name,d.file#,d.name;

    CON_ID PDB_NAME             OPEN_MODE       FILE# FILE_NAME
---------- -------------------- ---------- ----------
--------------------------------------------------------------------
------------
         3 SOURCE               MOUNTED            23 /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_system_k8qtdbnj_.dbf
         3 SOURCE               MOUNTED            24 /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_sysaux_k8qtdbny_.dbf
         3 SOURCE               MOUNTED            25 /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_sysext_k8qtdbo3_.dbf
         3 SOURCE               MOUNTED            26 /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/
o1_mf_cdb1_pdb_k8qtdbo7_.dbf

SQL> alter session set container = SOURCE;
```

```
Session altered.

SQL> alter pluggable database open;

Pluggable database altered.

SQL> create table tab1 (x number, y number, z number);

Table created.

SQL> insert into tab1 values (1, 2, 3);

1 row created.

SQL> commit;

Commit complete.

SQL> alter session set container = cdb$root;

Session altered.

SQL> alter pluggable database SOURCE close;

Pluggable database altered.

SQL> alter pluggable database SOURCE open read only;

Pluggable database altered.

SQL> create pluggable database CLONE from SOURCE snapshot copy;

Pluggable database created.

SQL> select d.con_id, p.name pdb_name, p.open_mode, d.file#, d.name
file_name from v$dbfile d, v$pdbs p where d.con_id=p.con_id order by
d.con_id,p.name,d.file#,d.name;

    CON_ID PDB_NAME             OPEN_MODE       FILE# FILE_NAME
---------- -------------------- ---------- ----------
--------------------------------------------------------------------------
---------
         3 SOURCE               READ ONLY          23 /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_system_k8qtdbnj_.dbf
         3 SOURCE               READ ONLY          24 /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_sysaux_k8qtdbny_.dbf
         3 SOURCE               READ ONLY          25 /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_sysext_k8qtdbo3_.dbf
         3 SOURCE               READ ONLY          26 /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_cdb1_pdb_k8qtdbo7_.dbf
         4 CLONE                MOUNTED            35 /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8stg6b9_.dbf
         4 CLONE                MOUNTED            36 /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysaux_k8stg6c5_.dbf
         4 CLONE                MOUNTED            37 /acfs/DB/
```

```
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysext_k8stg6cc_.dbf
         4 CLONE                MOUNTED            38 /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_cdb1_pdb_k8stg6cj_.dbf

$ ls -l /acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_system_k8stg6b9_.dbf
-rw-rw---- 1 oracle dba 259M May 24 10:22 /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8stg6b9_.dbf

$ /sbin/acfsutil info file -f /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8stg6b9_.dbf
...
    fshare:       Yes
    fshare parent identifier:       49
    fshare parent _genum:       1
...
         extent count: 79
         owned extent count: 41
         shared extent count: 38

$ /sbin/acfsutil info file -f /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysaux_k8stg6c5_.dbf
...
    fshare:       Yes
    fshare parent identifier:       48
    fshare parent _genum:       1
...
         extent count: 17
         owned extent count: 12
         shared extent count: 5

$ /sbin/acfsutil info file -f /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysext_k8stg6cc_.dbf
...
    fshare:       Yes
    fshare parent identifier:       45
    fshare parent _genum:       0
...
         extent count: 3
         owned extent count: 1
         shared extent count: 2


$ /sbin/acfsutil info file -f /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_cdb1_pdb_k8stg6cj_.dbf
/acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_cdb1_pdb_k8stg6cj_.dbf
    fshare:       Yes
    fshare parent identifier:       47
    fshare parent _genum:       1
...
         extent count: 7
         owned extent count: 7
```

```
         shared extent count: 0

SQL> alter pluggable database CLONE open;

Pluggable database altered.

SQL> alter session set container = CLONE;

Session altered.

SQL> select x,y,z from tab1;

         X          Y          Z
---------- ---------- ----------
         1          2          3

SQL> update tab1 set x = 2*x, y = 10 * y, z = 100 * z;

1 row updated.

SQL> insert into tab1 values (97, 98, 99);

1 row created.

SQL> select x,y,z from tab1;

         X          Y          Z
---------- ---------- ----------
         2         20        300
        97         98         99

SQL> commit;

Commit complete.

SQL> alter pluggable database CLONE close;

Pluggable database altered.

SQL>
```

2. Use RMAN to do a full backup of SOURCE and a sparse backup of CLONE.

```
RMAN> backup as copy pluggable database 'SOURCE'to destination '/backup';

Starting backup at 24-MAY-22
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile file number=00023 name=/acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_system_k8qtdbnj_.dbf
output file name=/backup/DB/DFB3F96AEBAD910AE0538E5246642001/datafile/
o1_mf_system_k8sv37z7_.dbf
tag=TAG20220524T103207 RECID=5 STAMP=1105525929
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting datafile copy
```

```
input datafile file number=00024 name=/acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_sysaux_k8qtdbny_.dbf
output file name=/backup/DB/DFB3F96AEBAD910AE0538E5246642001/
datafile/o1_mf_sysaux_k8sv3c49_.dbf
tag=TAG20220524T103207 RECID=6 STAMP=1105525931
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00026 name=/acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/
o1_mf_cdb1_pdb_k8qtdbo7_.dbf
output file name=/backup/DB/DFB3F96AEBAD910AE0538E5246642001/
datafile/o1_mf_cdb1_pdb_k8sv3d8y_.dbf
tag=TAG20220524T103207 RECID=7 STAMP=1105525932
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00025 name=/acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/o1_mf_sysext_k8qtdbo3_.dbf
output file name=/backup/DB/DFB3F96AEBAD910AE0538E5246642001/
datafile/o1_mf_sysext_k8sv3ff4_.dbf
tag=TAG20220524T103207 RECID=8 STAMP=1105525933 c
hannel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01

Finished backup at 24-MAY-22
Starting Control File Autobackup at 24-MAY-22
piece handle=/ade/b/1779015158/oracle/dbs/c-2109129204-20220524-02
comment=NONE
Finished Control File Autobackup at 24-MAY-22

RMAN> backup as sparse copy pluggable database 'CLONE' to
destination '/backup';

Starting backup at 24-MAY-22 using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy input datafile file
number=00035
name=/acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_system_k8stg6b9_.dbf
output file name=/backup/DB/DFC3A937C9E449D9E0538E52466425A4/
datafile/o1_mf_system_k8sv578m_.dbf
tag=TAG20220524T103310 RECID=9 STAMP=1105525991
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00036 name=/acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysaux_k8stg6c5_.dbf
output file name=/backup/DB/DFC3A937C9E449D9E0538E52466425A4/
datafile/o1_mf_sysaux_k8sv58dm_.dbf
tag=TAG20220524T103310 RECID=10 STAMP=1105525992
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00038 name=/acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_cdb1_pdb_k8stg6cj_.dbf
output file name=/backup/DB/DFC3A937C9E449D9E0538E52466425A4/
datafile/o1_mf_cdb1_pdb_k8sv59jc_.dbf
tag=TAG20220524T103310 RECID=11 STAMP=1105525993
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
```

```
channel ORA_DISK_1: starting datafile copy
input datafile file number=00037 name=/acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysext_k8stg6cc_.dbf
output file name=/backup/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_sysext_k8sv5bnf_.dbf
tag=TAG20220524T103310 RECID=12 STAMP=1105525994
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01

Finished backup at 24-MAY-22
Starting Control File Autobackup at 24-MAY-22
piece handle=/ade/b/1779015158/oracle/dbs/c-2109129204-20220524-03
comment=NONE
Finished Control File Autobackup at 24-MAY-22
```

3. Confirm that the sparse backup is sparse using 'du'. Note that the allocated size of the backup is less than the allocated size of the pluggable database's files.

```
$ ls -l /backup/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
total 191M
-rw-rw---- 1 oracle dba  96M May 24 10:33 o1_mf_cdb1_pdb_k8sv59jc_.dbf
-rw-rw---- 1 oracle dba 211M May 24 10:33 o1_mf_sysaux_k8sv58dm_.dbf
-rw-rw---- 1 oracle dba  40M May 24 10:33 o1_mf_sysext_k8sv5bnf_.dbf
-rw-rw---- 1 oracle dba 259M May 24 10:33 o1_mf_system_k8sv578m_.dbf

$ du -sch /backup/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/*
96M  /backup/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_cdb1_pdb_k8sv59jc_.dbf
88M  /backup/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_sysaux_k8sv58dm_.dbf
1.0M  /backup/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_sysext_k8sv5bnf_.dbf
7.3M  /backup/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_system_k8sv578m_.dbf
191M    total

$ du -sch /acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/*
96M  /acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_cdb1_pdb_k8stg6cj_.dbf
213M  /acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_sysaux_k8stg6c5_.dbf
40M  /acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_sysext_k8stg6cc_.dbf
264M  /acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_system_k8stg6b9_.dbf
41M  /acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_temp_k8stg6c9_.dbf
651M    total

SQL>
```

4. Remove the files for the CLONE pluggable database, simulating a media failure.

```
$ rm -rf /acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/*

$
```

5. Without recreating the fshares, the restore will fail.

```
RMAN> restore from sparse pluggable database 'CLONE';

Starting restore at 24-MAY-22
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=204 device type=DISK

channel ORA_DISK_1: restoring datafile 00035
input datafile copy RECID=9 STAMP=1105525991 file name=/backup/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8sv578m_.dbf
destination for restore of datafile 00035: /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8stg6b9_.dbf
RMAN-00571:
===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
===============
RMAN-00571:
===========================================================
RMAN-03002: failure of restore command at 05/24/2022 10:39:23
ORA-19566: exceeded limit of 0 corrupt blocks for file /backup/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8sv578m_.dbf
ORA-19600: input file is datafile-copy 9 (/backup/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_system_k8sv578m_.dbf)
ORA-19601: output file is datafile 35 (/acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_%u_.dbf)
```

6. Recreate the fshares for the files in the CLONE pluggable database. Note: This step is only necessary if these files do not exist. Depending on the reason for restoring the CLONE PDB, it may not be necessary to recreate the fshares.

```
$ /sbin/acfsutil fshare create /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/
o1_mf_system_k8qtdbnj_.dbf /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8stg6b9_.dbf
acfsutil fshare create: File share operation is complete.

$ /sbin/acfsutil fshare create /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/
o1_mf_sysaux_k8qtdbny_.dbf /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysaux_k8stg6c5_.dbf
acfsutil fshare create: File share operation is complete.

$ /sbin/acfsutil fshare create /acfs/DB/
DFB3F96AEBAD910AE0538E5246642001/datafile/
o1_mf_sysext_k8qtdbo3_.dbf /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysext_k8stg6cc_.dbf
```

```
acfsutil fshare create: File share operation is complete.

$ /sbin/acfsutil fshare create /acfs/DB/DFB3F96AEBAD910AE0538E5246642001/
datafile/o1_mf_cdb1_pdb_k8qtdbo7_.dbf /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_cdb1_pdb_k8stg6cj_.dbf
acfsutil fshare create: File share operation is complete.
```

7. Restore the CLONE pluggable database from the sparse backup.

```
RMAN> restore from sparse pluggable database 'CLONE';

Starting restore at 24-MAY-22
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=57 device type=DISK

channel ORA_DISK_1: restoring datafile 00035
input datafile copy RECID=9 STAMP=1105525991 file name=/backup/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8sv578m_.dbf
destination for restore of datafile 00035: /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_system_k8stg6b9_.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00035, elapsed time:
00:00:03
output file name=/acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_system_k8stg6b9_.dbf RECID=0 STAMP=0
channel ORA_DISK_1: restoring datafile 00036
input datafile copy RECID=10 STAMP=1105525992 file name=/backup/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysaux_k8sv58dm_.dbf
destination for restore of datafile 00036: /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysaux_k8stg6c5_.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00036, elapsed time:
00:00:03
output file name=/acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_sysaux_k8stg6c5_.dbf RECID=0 STAMP=0
channel ORA_DISK_1: restoring datafile 00037
input datafile copy RECID=12 STAMP=1105525994 file name=/backup/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysext_k8sv5bnf_.dbf
destination for restore of datafile 00037: /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_sysext_k8stg6cc_.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00037, elapsed time:
00:00:01
output file name=/acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_sysext_k8stg6cc_.dbf RECID=0 STAMP=0
channel ORA_DISK_1: restoring datafile 00038
input datafile copy RECID=11 STAMP=1105525993 file name=/backup/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_cdb1_pdb_k8sv59jc_.dbf
destination for restore of datafile 00038: /acfs/DB/
DFC3A937C9E449D9E0538E52466425A4/datafile/o1_mf_cdb1_pdb_k8stg6cj_.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00038, elapsed time:
00:00:03
output file name=/acfs/DB/DFC3A937C9E449D9E0538E52466425A4/datafile/
o1_mf_cdb1_pdb_k8stg6cj_.dbf RECID=0 STAMP=0
Finished restore at 24-MAY-22
```

8. Recover the CLONE pluggable database.

```
RMAN> recover pluggable database 'CLONE';

Starting recover at 24-MAY-22
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 24-MAY-22
```

9. Verify that the CLONE pluggable database is usable. Note that the table contains the expected contents after recovery.

```
SQL> alter pluggable database CLONE open;

Pluggable database altered.

SQL> alter session set container = CLONE;

Session altered.

SQL> select x,y,z from tab1;

     X         Y            Z
---------- ---------- ----------
     2        20          300
    97        98           99
SQL>
```

Steps to perform an RMAN sparse backup and restore of a PDB using an ACFS snapshot

1. Assume there are two PDBs SOURCE and CLONE. CLONE is a snapshot copy of SOURCE. The snapshot copy is a thin clone using an ACFS snapshot because the ACFS compatibility is 19.0.0.0. Note that the files in the CLONE PDB are symlinks pointing into the ACFS snapshot.

```
$ /sbin/acfsutil info fs
    on-disk version:       49.0
    ACFS compatibility:    19.0.0.0.0

SQL> create pluggable database SOURCE admin user foo identified by
bar;

Pluggable database created.

SQL> select d.con_id, p.name pdb_name, p.open_mode, d.file#, d.name
file_name from v$dbfile d, v$pdbs p where d.con_id=p.con_id order
by d.con_id,p.name,d.file#,d.name;
```

```
    CON_ID PDB_NAME             OPEN_MODE       FILE# FILE_NAME
---------- -------------------- ---------- ----------
--------------------------------------------------------------------------
----------
         3 SOURCE               MOUNTED            39 /acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_system_k8swj6ms_.dbf
         3 SOURCE               MOUNTED            40 /acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysaux_k8swj6my_.dbf
         3 SOURCE               MOUNTED            41 /acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysext_k8swj6n4_.dbf
         3 SOURCE               MOUNTED            42 /acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_cdb1_pdb_k8swj6n7_.dbf

SQL> alter session set container = SOURCE;

Session altered.

SQL> alter pluggable database open;

Pluggable database altered.

SQL> create table tab1 (x number, y number, z number);

Table created.

SQL> insert into tab1 values (1, 2, 3);

1 row created.

SQL> commit;

Commit complete.

SQL> alter session set container = cdb$root;

Session altered.

SQL> alter pluggable database SOURCE close;

Pluggable database altered.

SQL> alter pluggable database SOURCE open read only;

Pluggable database altered.

SQL> create pluggable database CLONE from SOURCE snapshot copy;

Pluggable database created.

    CON_ID PDB_NAME             OPEN_MODE       FILE# FILE_NAME
---------- -------------------- ---------- ----------
--------------------------------------------------------------------------
----------
         3 SOURCE               READ ONLY          39 /acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_system_k8swj6ms_.dbf
```

```
         3 SOURCE               READ ONLY          40 /acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysaux_k8swj6my_.dbf
         3 SOURCE               READ ONLY          41 /acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysext_k8swj6n4_.dbf
         3 SOURCE               READ ONLY          42 /acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swj6n7_.dbf
         4 CLONE                MOUNTED            43 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_k8swptmp_.dbf
         4 CLONE                MOUNTED            44 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysaux_k8swptmz_.dbf
         4 CLONE                MOUNTED            45 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysext_k8swptn5_.dbf
         4 CLONE                MOUNTED            46 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swptn8_.dbf

$ ls -l /acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/*
lrwxrwxrwx 1 oracle dba 124 May 24 10:59 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swptn8_.dbf -> /acfs/.ACFS/snaps/
DFC427F513EA6F24E0538E5246647E42/DB/
DFC427F513E56F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swj6n7_.dbf
lrwxrwxrwx 1 oracle dba 122 May 24 10:59 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysaux_k8swptmz_.dbf -> /acfs/.ACFS/snaps/
DFC427F513EA6F24E0538E5246647E42/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysaux_k8swj6my_.dbf
lrwxrwxrwx 1 oracle dba 122 May 24 10:59 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysext_k8swptn5_.dbf -> /acfs/.ACFS/snaps/
DFC427F513EA6F24E0538E5246647E42/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysext_k8swj6n4_.dbf
lrwxrwxrwx 1 oracle dba 122 May 24 10:59 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_system_k8swptmp_.dbf -> /acfs/.ACFS/snaps/
DFC427F513EA6F24E0538E5246647E42/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_system_k8swj6ms_.dbf
-rw-rw---- 1 oracle dba 41M May 24 10:59 /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_temp_k8swptn2_.dbf

$ /sbin/acfsutil info file -f /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swptn8_.dbf
    fshare:       No

$ /sbin/acfsutil snap info /acfs
snapshot name:               DFC427F513EA6F24E0538E5246647E42
snapshot location:           /acfs/.ACFS/snaps/
DFC427F513EA6F24E0538E5246647E42
RO snapshot or RW snapshot:  RW
storage interest tracking:   Disabled
parent name:                 /acfs
snapshot creation time:      Tue May 24 10:59:38 2022
```

```
file entry table allocation: 262144   ( 256.00 KB ) ( 64 entries )
storage added to snapshot:   38981632   (  37.18 MB )


    number of snapshots:  1
    kilosnap state:       Enabled
    storage interest tracking: Disabled
    snapshot space usage: 39034880  (  37.23 MB )


SQL> alter pluggable database CLONE open;

Pluggable database altered.

SQL> alter session set container = CLONE;

Session altered.

SQL> select x,y,z from tab1;

         X          Y          Z
---------- ---------- ----------
         1          2          3

SQL> update tab1 set x = 2*x, y = 10 * y, z = 100 * z;

1 row updated.

SQL> insert into tab1 values (97, 98, 99);

1 row created.

SQL> select x,y,z from tab1;

         X          Y          Z
---------- ---------- ----------
         2         20        300
        97         98         99

SQL> commit;

Commit complete.

SQL> alter pluggable database CLONE close;

Pluggable database altered.
SQL>
```

2. Use RMAN to do a full backup of SOURCE and a sparse backup of CLONE.

```
RMAN> backup as copy pluggable database 'SOURCE' to destination '/backup';

Starting backup at 24-MAY-22
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=51 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00039 name=/acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_system_k8swj6ms_.dbf
output file name=/backup/DB/DFC427F513E56F24E0538E5246647E42/
datafile/o1_mf_system_k8sxgx4x_.dbf tag=TAG20220524T111228 RECID=13
STAMP=1105528350
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting datafile copy
input datafile file number=00040 name=/acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysaux_k8swj6my_.dbf
output file name=/backup/DB/DFC427F513E56F24E0538E5246647E42/
datafile/o1_mf_sysaux_k8sxh06x_.dbf tag=TAG20220524T111228 RECID=14
STAMP=1105528352
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00042 name=/acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swj6n7_.dbf
output file name=/backup/DB/DFC427F513E56F24E0538E5246647E42/
datafile/o1_mf_cdb1_pdb_k8sxh1c9_.dbf tag=TAG20220524T111228
RECID=15 STAMP=1105528353
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00041 name=/acfs/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysext_k8swj6n4_.dbf
output file name=/backup/DB/DFC427F513E56F24E0538E5246647E42/
datafile/o1_mf_sysext_k8sxh2j0_.dbf tag=TAG20220524T111228 RECID=16
STAMP=1105528354
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 24-MAY-22

Starting Control File Autobackup at 24-MAY-22
piece handle=/ade/b/1779015158/oracle/dbs/c-2109129204-20220524-05
comment=NONE
Finished Control File Autobackup at 24-MAY-22

RMAN> backup as sparse copy pluggable database 'CLONE' to
destination '/backup';

Starting backup at 24-MAY-22
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile file number=00043 name=/acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_k8swptmp_.dbf
output file name=/backup/DB/DFC427F513EA6F24E0538E5246647E42/
datafile/o1_mf_system_k8sxj9x5_.dbf tag=TAG20220524T111313 RECID=17
STAMP=1105528394
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00044 name=/acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysaux_k8swptmz_.dbf
output file name=/backup/DB/DFC427F513EA6F24E0538E5246647E42/
datafile/o1_mf_sysaux_k8sxjc2h_.dbf tag=TAG20220524T111313 RECID=18
STAMP=1105528395
```

```
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00046 name=/acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_cdb1_pdb_k8swptn8_.dbf
output file name=/backup/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8sxjd70_.dbf tag=TAG20220524T111313 RECID=19
STAMP=1105528396
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00045 name=/acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysext_k8swptn5_.dbf
output file name=/backup/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysext_k8sxjfcp_.dbf tag=TAG20220524T111313 RECID=20
STAMP=1105528397
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 24-MAY-22

Starting Control File Autobackup at 24-MAY-22
piece handle=/ade/b/1779015158/oracle/dbs/c-2109129204-20220524-06
comment=NONE
Finished Control File Autobackup at 24-MAY-22
```

3. Confirm that the sparse backup is sparse using 'du'. Note that the allocated size of the backup is less than the allocated size of the pluggable database's files.

```
$ ls -l /backup/DB/DFC427F513EA6F24E0538E5246647E42/datafile/*
-rw-rw---- 1 oracle dba  96M May 24 11:13 /backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_cdb1_pdb_k8sxjd70_.dbf
-rw-rw---- 1 oracle dba 211M May 24 11:13 /backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysaux_k8sxjc2h_.dbf
-rw-rw---- 1 oracle dba  40M May 24 11:13 /backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysext_k8sxjfcp_.dbf
-rw-rw---- 1 oracle dba 259M May 24 11:13 /backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_k8sxj9x5_.dbf

$ du -sch /backup/DB/DFC427F513EA6F24E0538E5246647E42/datafile/*
4.3M    /backup/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8sxjd70_.dbf
19M    /backup/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysaux_k8sxjc2h_.dbf
1.0M    /backup/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysext_k8sxjfcp_.dbf
8.3M    /backup/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_system_k8sxj9x5_.dbf
32M    total

$ du -schH /acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/*
96M    /acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swptn8_.dbf
213M    /acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysaux_k8swptmz_.dbf
40M    /acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysext_k8swptn5_.dbf
264M    /acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_system_k8swptmp_.dbf
```

```
41M    /acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_temp_k8swptn2_.dbf
651M    total

SQL>
```

The pfile would be saved to the default location at `$ORACLE_HOME/dbs` directory.

4. Remove the files and ACFS snapshot for the CLONE pluggable database, simulating a media failure.

```
$ rm -rf /acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/*

$ /sbin/acfsutil snap delete DFC427F513EA6F24E0538E5246647E42 /acfs
acfsutil snap delete: Snapshot operation is complete.
$
```

5. Without recreating the snapshot, the restore will fail.

```
RMAN> restore from sparse pluggable database 'CLONE';

Starting restore at 24-MAY-22
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=51 device type=DISK

channel ORA_DISK_1: restoring datafile 00043
input datafile copy RECID=17 STAMP=1105528394 file name=/backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_k8sxj9x5_.dbf
destination for restore of datafile 00043: /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_k8swptmp_.dbf
RMAN-00571:
===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
===============
RMAN-00571:
===========================================================
RMAN-03002: failure of restore command at 05/24/2022 11:19:05
ORA-19566: exceeded limit of 0 corrupt blocks for file /backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_k8sxj9x5_.dbf
ORA-19600: input file is datafile-copy 17 (/backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_system_k8sxj9x5_.dbf)
ORA-19601: output file is datafile 43 (/acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_%u_.dbf)
```

6. Recreate the ACFS snapshot for the CLONE pluggable database. Recreate the symlinks pointing into the snapshot. Note: This step is only necessary if this snapshot and/or files do not exist. Depending on the reason for restoring the CLONE PDB, it may not be necessary to recreate the snapshot and/or symlinks.

```
$ /sbin/acfsutil snap create -w DFC427F513EA6F24E0538E5246647E42 /
acfs
acfsutil snap create: Snapshot operation is complete.
```

```
$ ln -s /acfs/.ACFS/snaps/DFC427F513EA6F24E0538E5246647E42/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_cdb1_pdb_k8swj6n7_.dbf /
acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swptn8_.dbf

$ ln -s /acfs/.ACFS/snaps/DFC427F513EA6F24E0538E5246647E42/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysaux_k8swj6my_.dbf /
acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysaux_k8swptmz_.dbf

$ ln -s /acfs/.ACFS/snaps/DFC427F513EA6F24E0538E5246647E42/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_sysext_k8swj6n4_.dbf /
acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysext_k8swptn5_.dbf

$ ln -s /acfs/.ACFS/snaps/DFC427F513EA6F24E0538E5246647E42/DB/
DFC427F513E56F24E0538E5246647E42/datafile/o1_mf_system_k8swj6ms_.dbf /
acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_system_k8swptmp_.dbf
```

7. Restore the CLONE pluggable database from the sparse backup.

```
RMAN> restore from sparse pluggable database 'CLONE';

Starting restore at 24-MAY-22
using channel ORA_DISK_1

channel ORA_DISK_1: restoring datafile 00043
input datafile copy RECID=17 STAMP=1105528394 file name=/backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_k8sxj9x5_.dbf
destination for restore of datafile 00043: /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_system_k8swptmp_.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00043, elapsed time:
00:00:01
output file name=/acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_system_k8swptmp_.dbf RECID=0 STAMP=0
channel ORA_DISK_1: restoring datafile 00044
input datafile copy RECID=18 STAMP=1105528395 file name=/backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysaux_k8sxjc2h_.dbf
destination for restore of datafile 00044: /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysaux_k8swptmz_.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00044, elapsed time:
00:00:01
output file name=/acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysaux_k8swptmz_.dbf RECID=0 STAMP=0
channel ORA_DISK_1: restoring datafile 00045
input datafile copy RECID=20 STAMP=1105528397 file name=/backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysext_k8sxjfcp_.dbf
destination for restore of datafile 00045: /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/o1_mf_sysext_k8swptn5_.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00045, elapsed time:
00:00:01
output file name=/acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_sysext_k8swptn5_.dbf RECID=0 STAMP=0
```

```
channel ORA_DISK_1: restoring datafile 00046
input datafile copy RECID=19 STAMP=1105528396 file name=/backup/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8sxjd70_.dbf
destination for restore of datafile 00046: /acfs/DB/
DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swptn8_.dbf
channel ORA_DISK_1: copied datafile copy of datafile 00046, elapsed
time: 00:00:01
output file name=/acfs/DB/DFC427F513EA6F24E0538E5246647E42/datafile/
o1_mf_cdb1_pdb_k8swptn8_.dbf RECID=0 STAMP=0
Finished restore at 24-MAY-22
```

8. Recover the CLONE pluggable database.

```
RMAN> recover pluggable database 'CLONE';

Starting recover at 24-MAY-22
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 24-MAY-22
```

9. Verify that the CLONE pluggable database is usable. Note that the table contains the expected contents after recovery.

```
SQL> alter pluggable database CLONE open;

Pluggable database altered.

SQL> alter session set container = CLONE;

Session altered.

SQL> select x,y,z from tab1;

     X         Y           Z
---------- ---------- ----------
     2        20         300
    97        98          99
SQL>
```

Oracle ACFS Loopback Support

Oracle ACFS supports loopback functionality on the Linux operating system, enabling Oracle ACFS files to be accessed as devices.

An Oracle ACFS loopback device is an operating system pseudo-device that enables an Oracle ACFS file to be accessed as a block device. This functionality can be used with Oracle Virtual Machines (OVM) in support of OVM images, templates, and virtual disks (vdisks) created in Oracle ACFS file systems and presented through Oracle ACFS loopback devices.

Oracle ACFS loopback functionality provides performance gains over NFS. Files can be sparse or non-sparse.

In addition to general loopback support, Oracle ACFS also provides support for loopback direct I/O (DIO) on sparse images.

Oracle ACFS Drivers Resource Management

Oracle ACFS, Oracle ADVM, and OKS drivers are loaded during the start of the Oracle Grid Infrastructure stack, except in an Oracle Restart configuration. The drivers remain loaded until the system is rebooted, at which point, they are loaded again when the Oracle Grid Infrastructure stack restarts.

For information about commands to manage Oracle ACFS, Oracle ADVM, and OKS drivers, refer to "Oracle ACFS Driver Commands".

Oracle ACFS Registry Resource Management

The Oracle ACFS registry resource is supported only for Oracle Grid Infrastructure cluster configurations; it is not supported for Oracle Restart configurations. See "Oracle ACFS and Oracle Restart".

With Oracle ASM 12*c* Release 1 (12.1), the Oracle ACFS registry uses the standard single file system resource available through the SRVCTL file system interface. For more information, refer to "Oracle ACFS File System Resource Management". Using SRVCTL enables applications to depend on registered file systems, such as for management of the registered file systems using `srvctl filesystem`. By default, `acfsutil registry` shows only file systems that are set to be always mounted, with the `AUTO_START` attribute set to `always`.

The Oracle ACFS registry requires root privileges to register and delete file systems, however, other users can be entitled to start and stop (mount and unmount) the file systems by use of the `user` option.

Oracle ACFS File System Resource Management

The Oracle ACFS file system resource is supported only for Oracle Grid Infrastructure cluster configurations; it is not supported for Oracle Restart configurations. See "Oracle ACFS and Oracle Restart".

Oracle ASM Configuration Assistant (ASMCA) facilitates the creation of Oracle ACFS file system resources (`ora.`*`diskgroup`*`.`*`volume`*`.acfs`). During database creation with Database Configuration Assistant (DBCA), the Oracle ACFS file system resource is included in the dependency list of its associated disk group so that stopping the disk group also attempts to stop any dependent Oracle ACFS file systems.

An Oracle ACFS file system resource is typically created for use with application resource dependency lists. For example, if an Oracle ACFS file system is configured for use as an Oracle Database home, then a resource created for the file system can be included in the resource dependency list of the Oracle Database application. This dependency causes the file system and stack to be automatically mounted due to the start action of the database application.

The start action for an Oracle ACFS file system resource is to mount the file system. This Oracle ACFS file system resource action includes confirming that the associated file system storage stack is active and mounting the disk group, enabling the volume file, and creating the mount point if necessary to complete the mount operation. If the file system is successfully mounted, the state of the resource is set to `online`; otherwise, it is set to `offline`.

The check action for an Oracle ACFS file system resource verifies that the file system is mounted. It sets the state of the resource to `online` status if mounted, otherwise the status is set to `offline`.

The stop action for an Oracle ACFS file system resource attempts to dismount the file system. If the file system cannot be dismounted due to open references, the stop action displays and logs the process identifiers for any processes holding a reference.

Use of the `srvctl start` and `stop` actions to manage the Oracle ACFS file system resources maintains their correct resource state.

Oracle ACFS and Oracle Restart

Oracle Restart does not support root-based Oracle ACFS resources for this release. Consequently, the following operations are not automatically performed:

- Loading Oracle ACFS drivers

  On Linux, drivers are automatically loaded and unloaded at system boot time and system shutdown time. If an action is required while the system is running or the system is running on other operating system (OS) versions, you can load or unload the drivers manually with the `acfsload` command. However, if the drivers are loaded manually,then the Oracle ACFS drivers must be loaded before the Oracle Restart stack is started.

  For more information, refer to acfsload.

- Mounting Oracle ACFS file systems listed in the Oracle ACFS mount registry

  The Oracle ACFS mount registry is not supported in Oracle Restart. However, Linux entries in the `/etc/fstab` file with a valid Oracle ASM device do have the associated volume enabled and are automatically mounted on system startup and unmounted on system shutdown. Note that high availability (HA) recovery is not applied after the file system is mounted; that functionality is a one time action.

  A valid `fstab` entry has the following format:

  ```
  device mount_point acfs noauto 0 0
  ```

  For example:

  ```
  /dev/asm/dev1-123 /mntpoint acfs noauto 0 0
  ```

  The last three fields in the previous example prevent Linux from attempting to automatically mount the device and from attempting to run other system tools on the device. This action prevents errors when the Oracle ASM instance is not available at times during the system startup. Additional standard `fstab` syntax options may be added for the file system mount.

  Should a mount or unmount operation be required on other OS versions, or after the system is started, you can mount Oracle ACFS file systems manually with the `mount` command. For information, refer to Managing Oracle ACFS with Command-Line Tools.

- Mounting resource-based Oracle ACFS database home file systems

  The Oracle ACFS resources associated with these actions are not created for Oracle Restart configurations. While Oracle ACFS resource management is fully supported for Oracle Grid Infrastructure configurations, the Oracle ACFS resource-based management actions must be replaced with alternative, sometimes manual, operations in Oracle Restart configurations. During an attempt to use commands, such as `srvctl`, that register a root-based resource in Oracle Restart configurations, an appropriate error is displayed.

Oracle ACFS Driver Commands

This section describes the Oracle ACFS driver commands that are used during installation to manage Oracle ACFS, Oracle ADVM, and Oracle Kernel Services Driver (OKS) drivers. These commands are located in the `/bin` directory of the Oracle Grid Infrastructure home.

- acfsload
- acfsdriverstate

acfsload

Purpose

`acfsload` loads or unloads Oracle ACFS, Oracle ADVM, and Oracle Kernel Services Driver (OKS) drivers.

Syntax

```
acfsload { start | stop  } [ -s ]
```

`acfsload –h` displays help text and exits.

Table 7-2 contains the options available with the `acfsload` command.

Table 7-2 Options for the acfsload command

Option	Description
`start`	Loads the Oracle ACFS, Oracle ADVM, and OKS drivers.
`stop`	Unloads the Oracle ACFS, Oracle ADVM, and OKS drivers.
`-s`	Operate in silent mode.

Description

You can use `acfsload` to manually load or unload the Oracle ACFS, Oracle ADVM, and OKS drivers.

Before unloading drivers with the `stop` option, you must dismount Oracle ACFS file systems and shut down Oracle ASM. For information about dismounting Oracle ACFS file systems, refer to Deregistering, Dismounting, and Disabling Volumes and Oracle ACFS File Systems.

root or administrator privilege is required to run acfsload.

Examples

The following is an example of the use of acfsload to stop (unload) all drivers.

```
# acfsload stop
```

acfsdriverstate

Purpose

acfsdriverstate provides information on the current state of the Oracle ACFS, Oracle ADVM, and Oracle Kernel Services Driver (OKS) drivers.

Syntax

```
acfsdriverstate [-orahome ORACLE_HOME ]
    { installed | loaded | version [-v] |
    supported [-v] [-k <kernel_rpm_file_path> |
<installed_kernel_rpm_package>} [-s]
```

acfsdriverstate –h displays help text and exits.

Table 7-3 contains the options available with the acfsdriverstate command.

Table 7-3 Options for the acfsdriverstate command

Option	Description
-orahome *ORACLE_HOME*	Specifies the Oracle Grid Infrastructure home in which the user has permission to run the acfsdriverstate command.
installed	Determines whether Oracle ACFS is installed on the system.
loaded	Determines whether the Oracle ADVM, Oracle ACFS, and OKS drivers are loaded in memory.
version	Reports the currently installed version of the Oracle ACFS system software.
supported	Reports whether the system is a supported kernel for Oracle ACFS.
supported -k	This command option accepts an arbitrary kernel rpm – either a file or one already installed on the system – and determine if the current ACFS install is supported on the specified kernel. To specify a installed kernel rpm package, verify the file "/boot/vmlinuz-<*kernel_version*>" has read permissions, to gather the installed rpm that provides the appropriate symbols gather the rpm package name with: rpm -qf /boot/vmlinuz-<*kernel_version*>. To specify a kernel rpm file, verify the file has read permissions and that the rpm provides the appropriate symbols, of the form kernel(*symbolfoo*) = *0xvaluefoo*. The symbols can be verified using: rpm -qip --provides <*path_to_rpm_file*>.
-v	Specifies verbose mode for additional details.

Description

You can use `acfsdriverstate` to display detailed information on the current state of the Oracle ACFS, Oracle ADVM, and OKS drivers.

Examples

The following is an example of the use of `acfsdriverstate`.

```
$ acfsdriverstate version
ACFS-9325:     Driver OS kernel version = 3.8.13-13.el6uek.x86_64.
ACFS-9326:     Driver build number = 171126.
ACFS-9212:     Driver build version = 18.1.0.0 ()..
ACFS-9547:     Driver available build number = 171126.
ACFS-9548:     Driver available build version = 18.1.0.0 ()..
```

Oracle ACFS Plug-in Generic Application Programming Interface

Oracle ACFS plug-in operations are supported through a common, operating system (OS) independent file plug-in (C library) application programming interface (API).

The topics contained in this section are:

- Oracle ACFS Pre-defined Metric Types
- Oracle ACFS Plug-in APIs

For more information about Oracle ACFS plug-ins, refer to "Oracle ACFS Plugins".

Oracle ACFS Pre-defined Metric Types

Oracle ACFS provides the `ACFSMETRIC1_T` and `ACFSMETRIC2_T` pre-defined metric types.

The `ACFSMETRIC1_T` metric set is defined for the storage virtualization model. The metrics are maintained as a summary record for either a selected set of tagged files or all files in the file system. Oracle ACFS file metrics include: number of reads, number of writes, average read size, average write size, minimum and maximum read size, minimum and maximum write size, and read cache (VM page cache) hits and misses.

Example:

```
typedef struct _ACFS_METRIC1 {
    ub2      acfs_version;
    ub2      acfs_type;
    ub4      acfs_seqno;
    ub8      acfs_nreads;
    ub8      acfs_nwrites;
    ub8      acfs_rcachehits;
    ub4      acfs_avgrsize;
    ub4      acfs_avgwsize;
    ub4      acfs_minrsize;
    ub4      acfs_maxrsize;
    ub4      acfs_minwsize;
    ub4      acfs_maxwsize;
    ub4      acfs_rbytes_per_sec;
    ub4      acfs_wbytes_per_sec;
```

```
    ub8      acfs_timestamp;
    ub8      acfs_elapsed_secs;
} ACFS_METRIC1;
```

The ACFSMETRIC2_T is a list of Oracle ACFS write description records containing the fileID, starting offset, size, and sequence number of each write. The sequence number preserves the Oracle ACFS write record order as preserved by the plug-in driver. The sequence number provides a way for applications to order multiple message buffers returned from the API. It also provides detection of dropped write records due to the application not draining the message buffers fast enough through the API.

The write records are contained within multiple in-memory arrays. Each array of records may be fetched with the API with a buffer size currently set to 1 M. At the beginning of the fetched ioctl buffer is a struct which describes the array, including the number of records it contains. The kernel buffers drop the oldest write records if the buffers are filled because the buffers are not being read quickly enough.

Example:

```
typedef struct _ACFS_METRIC2 {
  ub2               acfs_version;
  ub2               acfs_type;
  ub4               acfs_num_recs;
  ub8               acfs_timestamp;
  ACFS_METRIC2_REC acfs_recs[1];
} ACFS_METRIC2;

typedef struct _ACFS_FILE_ID {
  ub8               acfs_fenum;
  ub4               acfs_genum;
  ub4               acfs_reserved1;
}

typedef struct _ACFS_METRIC2_REC {
  ACFS_FILE_ID      acfs_file_id;
  ub8               acfs_start_offset;
  ub8               acfs_size;
  ub8               acfs_seq_num;
} ACFS_METRIC2_rec;
```

Oracle ACFS Plug-in APIs

Purpose

The Oracle ACFS plug-in application programming interface (API) sends and receives messages to and from the local plug-in enabled Oracle ACFS driver from the application plug-in module.

Syntax

```
sb8 acfsplugin_metrics(ub4 metric_type,
  ub1 *metrics,
  ub4 metric_buf_len,
  oratext *mountp );

sb8 acfsfileid_lookup(ACFS_FILEID file_id,
  oratext *full_path,
  oratext *mountp );
```

Description

The acfsplugin_metrics API is used by an Oracle ACFS application plug-in module to retrieve metrics from the Oracle ACFS driver. The Oracle ACFS driver must first be enabled for plug-in communication using the acfsutil plugin enable command. The selected application plug-in metric type model must match the plug-in configuration defined with the Oracle ACFS plug-in enable command. For information about the acfsutil plugin enable command, refer to "acfsutil plugin enable". The application must provide a buffer large enough to store the metric structures described in "Oracle ACFS Pre-defined Metric Types".

If the provided buffer is NULL and metric_buf_len = 0, the return value is the size required to hold all the currently collected metrics. The application can first query Oracle ACFS to see how big a buffer is required, then allocate a buffer of the necessary size to pass back to Oracle ACFS.

The mount path must be provided to the API to identify the plug-in enabled Oracle ACFS file system that is being referenced.

A nonnegative value is returned for success: 0 for success with no more metrics to collect, 1 to indicate that more metrics are available, or 2 to indicate that no new metrics were collected during the interval. In the case of an error, a negative value is returned and errno is set on Linux environments.

When using metric type #2, the returned metrics include an ACFS_FILE_ID, which contains the fenum and genum pair. In order to translate from the fenum and genum pair to a file path, the application can use acfsfileid_lookup. The application must provide a buffer of length ACFS_FILEID_MAX_PATH_LEN to hold the path. If there are multiple hard links to a file, the returned path is the first one. This is same behavior when using acfsutil info id.

System administrator or Oracle ASM administrator privileges are required to send and receive messages to and from the plug-in enabled Oracle ACFS file system driver.

Writing Applications

To use the plugin API, applications must include the C header file acfslib.h which defines the API functions and structures.

```
#include <acfslib.h>
```

When building the application executable, the application must be linked with the acfs12 library. Check the platform-specific documentation for information about environment variables that must be defined. For example:

```
export LD_LIBRARY_PATH=${ORACLE_HOME}/lib:$
{LD_LIBRARY_PATH}
```

Then when linking, add the -lacfs12 flag.

Examples

In Example 7-1, the command enables an Oracle ACFS file system mounted on /humanresources for the plug-in service.

Example 7-1 Application Plug-in for Storage Visibility: Poll Model

```
$ /sbin/acfsutil plugin enable -m acfsmetric1 -t HRDATA /humanresources
```

With this command, the application plug-in polls the Oracle ACFS plug-in enabled driver for summary metrics associated with files tagged with `HRDATA`. The application code includes the following:

```
#include <acfslib.h>
...
/* allocate message buffers */
ACFS_METRIC1 *metrics = malloc (sizeof(ACFS_METRIC1));
/* poll for metric1 data */
while (condition) {
  /* read next summary message from ACFS driver */
   if ((rc = acfsplugin_metrics(ACFS_METRIC_TYPE1,(ub1*)metrics,sizeof(*metrics),
        mountp)) < 0) {
        perror("….Receive failure … ");
        break;
   }
   /* print message data */
   printf ("reads %8llu ", metrics->acfs_nreads);
   printf("writes %8llu ", metrics->acfs_nwrites);
   printf("avg read size %8u ", metrics->acfs_avgrsize);
   printf("avg write size %8u ", metrics->acfs_avgwsize);
   printf("min read size %8u ", metrics->acfs_minrsize);
   printf("max read size %8u ", metrics->acfs_maxrsize);
   ...
   sleep (timebeforenextpoll);
}
```

In Example 7-2, the command enables an Oracle ACFS file system mounted on `/humanresources` for the plug-in service.

Example 7-2   Application Plug-in for File Content: Post Model

```
$ /sbin/acfsutil plugin enable -m acfsmetric1 -t HRDATA -i 5m /humanresources
```

With this command, every 5 minutes the Oracle ACFS plug-in enabled driver posts file content metrics associated with files tagged with `HRDATA`. In the application code, the call to `acfsplugin_metrics()` is blocked until the metrics are posted. The application code includes the following:

```
#include <acfslib.h>
...
 ACFS_METRIC1 *metrics = malloc (sizeof(ACFS_METRIC1));

 /* Wait for metric Data */
  while (condition) {
    /* Wait for next file content posting from ACFS driver */
    rc = ACFS_PLUGIN_MORE_AVAIL;
    /* A return code of 1 indicates that more metrics are available
    * in the current set of metrics.
    */
    while( rc == ACFS_PLUGIN_MORE_AVAIL) {
      /* This call blocks until metrics are available. */
      rc = acfsplugin_metrics(ACFS_METRIC_TYPE1,(ub1*)metrics,sizeof(*metrics),
           mountp);
      if (rc < 0) {
        perror("….Receive failure … ");
        break;
      } else if (rc == ACFS_PLUGIN_NO_NEW_METRICS) {
        printf("No new metrics available.");
        break;
     }
```

```
      if (last_seqno != metrics->acfs_seqno-1 ) {
        printf("Warning: Unable to keep up with metrics collection.");
        printf("Missed %d sets of posted metrics.",
               (metrics->acfs_seqno-1)-last_seqno);
      }

       /* print message data */
       printf ("reads %8llu ", metrics->acfs_nreads);
       printf("writes %8llu ", metrics->acfs_nwrites);
       printf("avg read size %8u ", metrics->acfs_avgrsize);
       printf("avg write size %8u ", metrics->acfs_avgwsize);
       printf("min read size %8u ", metrics->acfs_minrsize);
       printf("max read size %8u ", metrics->acfs_maxrsize);
       ...

       last_seqno = metrics->acfs_seqno;
     }
   }

   free(metrics);
```

Example 7-3 Application for Resolving the File Path from a Fenum and Genum Pair

Oracle ACFS Tagging Generic Application Programming Interface

Oracle ACFS tagging operations are supported through a common operating system (OS) independent file tag (C library) application programming interface (API).

An Oracle ACFS tagging API demonstration utility is provided. The demo provides instructions to build the utility with a makefile on each supported platform.

On Solaris, Oracle ACFS tagging APIs can set tag names on symbolic link files, but backup and restore utilities do not save the tag names that are explicitly set on the symbolic link files. Also, symbolic link files lose explicitly set tag names if they have been moved, copied, tarred, or paxed.

The following files are included:

- `$ORACLE_HOME/usm/public/acfslib.h`
- `$ORACLE_HOME/usm/demo/acfstagsdemo.c`
- `$ORACLE_HOME/usm/demo/Makefile`

  Linux, Solaris, or AIX makefile for creating the demo utility.

The topics contained in this section are:

- Oracle ACFS Tag Name Specifications
- Oracle ACFS Tagging Error Values
- acfsgettag
- acfslisttags
- acfsremovetag
- acfssettag

Oracle ACFS Tag Name Specifications

An Oracle ACFS tag name can be from 1 to 32 characters in length and consist of a combination of the following set of characters only:

- uppercase and lowercase alphabetic characters (A-Z, a-z)
- numbers (0-9)
- hyphen (-)
- underscore (_)
- blank (space)

Oracle ACFS Tagging Error Values

The following are the values for Linux, Solaris, or AIX `errno` in case of failure:

- `EINVAL` – The tag name syntax is invalid or too long.
- `ENODATA` – The tag name does not exist for this file or directory.
- `ERANGE` - The value buffer is too small to hold the returned value.
- `EACCES` – Search permission denied for a directory in the path prefix of path; or the user does not have permission on the file to read tag names.
- `ENAMETOOLONG` – The file name is too long.
- `ENOENT` – A component of path does not exist.

acfsgettag

Purpose

Retrieves the value associated with an Oracle ACFS file tag name.

Syntax

```
sb8 acfsgettag(const oratext *path, const oratext *tagname, oratext *value,
               size_t size, ub4 flags);
```

Table 7-4 contains the options available with the `acfsgettag` command.

Table 7-4 Options for the acfsgettag command

Option	Description
path	Specifies a pointer to a file or directory path name.
tagname	Specifies a pointer to a NULL-terminated Oracle ACFS tag name in the format of a valid tag name for regular files and directories.
value	Specifies the memory buffer to retrieve the Oracle ACFS tag value.
size	Specifies the byte size of the memory buffer that holds the returned Oracle ACFS tag value.
flags	Reserved for future use. Must be set to 0.

Description

The acfsgettag library call retrieves the value string of the Oracle ACFS tag name. The return value is the nonzero byte length of the output *value* string on success or ACFS_TAG_FAIL on failure. For information about operating system-specific extended error information values that may be obtained when an ACFS_TAG_FAIL is returned, refer to "Oracle ACFS Tagging Error Values".

Because Oracle ACFS tag names currently use a fixed value string of 0 (the number zero character with a byte length of one) the value is the same for all Oracle ACFS tag name entries. The size of the *value* buffer can be determined by calling acfsgettag with a NULL *value* and 0 *size*. The library call returns the byte size necessary to hold the value string of the tag name. acfsgettag returns an ENODATA error when the tag name is not set on the file.

Examples

Example 7-4 is an example of the use of the acfsgettag function call.

Example 7-4 Retrieving a file tag value

```
sb8 rc;
size_t size;
oratext value[2];
const oratext *path = "/mnt/dir1/dir2/file2";
const oratext *tagname = "patch_set_11_1";
size = 1; (byte)
memset((void *)value, 0, 2*sizeof(oratext));
rc = acfsgettag (path, tagname, value, size, 0);
If (rc == ACFS_TAG_FAIL)
  /* check errno or GetLastError() to process error returns /*
```

acfslisttags

Purpose

Lists the tag names assigned to an Oracle ACFS file. For additional information, refer to "acfsutil tag info".

Syntax

```
sb8 acfslisttags(const oratext *path, oratext *list, size_t size, ub4 flags);
```

Table 7-4 contains the options available with the acfslisttags command.

Table 7-5 Options for the acfslisttags command

Option	Description
path	Specifies a pointer to a file or directory path name.
list	Specifies a pointer to a memory buffer containing the list of Oracle ACFS tag names.
size	Specifies the size (bytes) of the memory buffer that holds the returned Oracle ACFS tag name list.
flags	Reserved for future use. Must be set to 0.

Description

The acfslisttags library call retrieves all the tag names assigned to an Oracle ACFS file. acfslisttags returns a list of tag names into the *list* memory buffer. Each tag name in the list is terminated with a NULL. If a file has no tag names then the list is empty. The memory buffer must be large enough to hold all of the tag names assigned to an Oracle ACFS file.

An application must allocate a buffer and specify a list size large enough to hold all of the tag names assigned to an Oracle ACFS file. An application can optionally obtain the list buffer size needed by first calling acfslisttags with a zero value buffer size and NULL list buffer. The application then checks for nonzero, positive list size return values to allocate a list buffer and call acfslisttags to retrieve the actual tag name list.

On success, the return value is a positive byte size of the tag name list or 0 when the file has no tag names. On failure, the return value is ACFS_TAG_FAIL. For information about operating system-specific extended error information values that may be obtained when an ACFS_TAG_FAIL is returned, refer to "Oracle ACFS Tagging Error Values".

Examples

Example 7-5 is an example of the use of the acfslisttags function call.

Example 7-5   Listing file tags

```
sb8 listsize;
sb8 listsize2;
const oratext *path = "/mnt/dir1/dir2/file2";
oratext *list;
/* Determine size of buffer to store list */
listsize = acfslisttags (path, NULL, 0, 0);
if (listsize == ACFS_TAG_FAIL)
/* retrieve the error code and return */

if (listsize)
{
    list = malloc(listsize)
    /* Retrieve list of tag names */
    listsize2 = acfslisttags (path, list, listsize, 0);
    if (listsize2 == ACFS_TAG_FAIL)
        /* check errno or GetLastError() to process error returns */
    if (listsize2 > 0)
        /* file has a list of tag names to process */
    else
        /* file has no tag names. */
}
else
/* file has no tag names. */
```

acfsremovetag

Purpose

Removes the tag name on an Oracle ACFS file.

Syntax

```
sb8 acfsremovetag(const oratext *path, const oratext *tagname, ub4 flags);
```

Table 7-6 contains the options available with the `acfsremovetag` command.

Table 7-6 Options for the acfsremovetag command

Option	Description
path	Specifies a pointer to a file or directory path name.
tagname	Specifies a pointer to a NULL-terminated Oracle ACFS tag name in the format of a valid tag name for regular files and directories.
flags	Reserved for future use. Must be set to `0`.

Description

The `acfsremovetag` library call removes a tag name on an Oracle ACFS file. The return value is `ACFS_TAG_SUCCESS` or `ACFS_TAG_FAIL`. For information about operating system-specific extended error information values that may be obtained when an `ACFS_TAG_FAIL` is returned, refer to "Oracle ACFS Tagging Error Values".

Examples

Example 7-6 is an example of the use of the `acfsremovetag` function call.

Example 7-6 Removing file tags

```
sb8 rc;
const oratext *path= "/mnt/dir1/dir2/file2";
const oratext *tagname = "patch_set_11_1";
rc = acfsremovetag (path, tagname, 0);
If (rc == ACFS_TAG_FAIL)
  /* check errno or GetLastError() to process error returns */
```

acfssettag

Purpose

Sets the tag name on an Oracle ACFS file. For additional information, refer to "acfsutil tag set".

Syntax

```
sb8 acfssettag(const oratext *path, const oratext *tagname, oratext *value,
               size_t size, ub4 flags);
```

Table 7-7 contains the options available with the `acfssettag` command.

Table 7-7 Options for the acfssettag command

Option	Description
path	Specifies a pointer to a file or directory path name.

Table 7-7 (Cont.) Options for the acfssettag command

Option	Description
tagname	Specifies a pointer to a NULL-terminated Oracle ACFS tag name in the format of a valid tag name for regular files and directories.
value	Specifies the memory buffer to set the Oracle ACFS tag value.
size	Specifies the byte size of the Oracle ACFS tag value.
flags	Reserved for future use. Must be set to 0.

Description

The `acfssettag` library call sets a tag name on an Oracle ACFS file. The return value is `ACFS_TAG_SUCCESS` or `ACFS_TAG_FAIL`. For information about operating system-specific extended error information values that may be obtained when an `ACFS_TAG_FAIL` is returned, refer to "Oracle ACFS Tagging Error Values".

Because Oracle ACFS tag names currently use a fixed value string of `0` (the number zero character with a byte length of one) the *`value`* is the same for all Oracle ACFS tag name entries.

Examples

Example 7-7 is an example of the use of the `acfssettag` function call.

Example 7-7 Setting file tags

```
sb8 rc;
size_t size;
const oratext *value ;
const oratext *path= "/mnt/dir1/dir2/file2";
const oratext *tagname = "patch_set_11_1";
value = "0"; /* zero */
size = 1; (byte)
rc = acfssettag (path, tagname, (oratext *)value, size, 0);
If (rc == ACFS_TAG_FAIL)
  /* check errno and GetLastError() to process error returns */
```

Understanding Oracle ACFS I/O Failure Console Messages

Oracle ACFS logs information for I/O failures in the operating-specific system event log.

A console message has the following format:

```
[Oracle ACFS]: I/O failure (error_code) with device device_name during a operation_name op_type.
file_entry_num Starting offset: offset. Length of data transfer: io_length bytes.
Impact: acfs_type   Object: object_type   Oper.Context: operation_context
Snapshot?: yes_or_no AcfsObjectID: acfs_object_id . Internal ACFS Location: code_location.
```

The italicized variables in the console message syntax correspond to the following:

- I/O failure

The operating system-specific error code, in Hex, seen by Oracle ACFS for a failed I/O. This may indicate a hardware problem, or it might indicate a failure to initiate the I/O for some other reason.

- Device

  The device involved, usually the ADVM device file, but under some circumstances it might be a string indicating the device minor number

- Operation name

  The kind of operation involved:

  `user data`, `metadata`, or `paging`

- Operation type

  The type of operation involved:

  `synch read`, `synch write`, `asynch read`, or `asynch write`

- File entry number

  The Oracle ACFS File entry number of the file system object involved, as a decimal number. The `acfsutil info fileid` tool finds the corresponding file name.

- Offset

  The disk offset of the I/O, as a decimal number.

- Length of I/O

  The length of the I/O in bytes, as decimal number.

- File system object impacted

  An indication that the file system object involved is either node-local, or is a resource accessed clusterwide. For example:

  `Node` or `Cluster`

- Type of object impacted

  A string indicating the kind of file system object involved, when possible. For example:

  `Unknown`, `User Dir.`, `User Symlink`, `User File`, `Sys.Dir`, `Sys.File`, or `MetaData`

  – `Sys.Dir.`

    Oracle ACFS-administered directory within the visible namespace

  – `sys.File`

    Oracle ACFS-administered file within the visible namespace

  – `MetaData`

    Oracle ACFS-administered resources outside of the visible namespace

- Operational context

  A higher-level view of what code context was issuing the I/O. This is for use by Oracle Support Services. For example:

  `Unknown`, `Read`, `Write`, `Grow`, `Shrink`, `Commit`, or `Recovery`

- Snapshot

  An indication of whether, if possible to determine, the data involved was from a Snapshot. For example:

Yes, No, or ?

- Object type of the file system

  An internal identifier for the type of file system object. For use by Oracle Support Services.

- Location of the code

  An internal identifier of the code location issuing this message. For use by Oracle Support Services.

The following is an example from /var/log/messages in a Linux environment:

```
[Oracle ACFS]: I/O failure (0xc0000001) with device /dev/sdb during a metadata synch write .
Fenum Unknown. Starting offset: 67113984. Length of data transfer: 2560 bytes.
Impact: Node   Object: MetaData   Oper.Context: Write
Snapshot?: ?  AcfsObjectID: 8   . Internal ACFS Location: 5 .
```

Configuring Oracle ACFS Snapshot-Based Replication

The requirements for Oracle ACFS snapshot-based replication are discussed in this section.

This section describes how to configure Oracle ACFS snapshot-based replication available with release 12.2 or higher. As with Oracle ACFS replication installations before release 12.2, the overall functional goal of snapshot-based replication is to ensure that updates from a primary cluster are replicated to a standby cluster. However, the snapshot based replication technology uses snapshots of the primary storage location and transfers the differences between successive snapshots to the standby storage location using the standard ssh command. Oracle ACFS replication functionality before release 12.2 replicated changes continuously, building on Oracle networking technologies, notably Network Foundation Technologies (NFT), to ensure connectivity between the primary and standby clusters.

This change in the design and implementation of Oracle ACFS replication introduces some differences in how replication is configured and used. For example, the use of ssh requires setting up host and user keys appropriately on the primary and standby nodes where replication is performed.

Oracle ACFS replication also provides role reversal and failover capabilites that you can configure by enabling both the primary cluster and the standby cluster to communicate with each other as required. In role reversal, the standby assumes the role of the primary and the primary becomes the standby. Failover may involve either role reversal or the establishment of a new standby for the new primary to use.

This section contains the following topics:

- Configuring ssh for Use With Oracle ACFS Replication

See Also:

- Oracle ACFS Replication for an overview of Oracle ACFS replication
- Oracle ACFS Command-Line Tools for Replication for information about Oracle ACFS replication commands
- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM privileges

Choosing an Oracle ACFS Replication User

The user identity under which replication is performed (the *repluser*) must be carefully managed. This is the user that will scan the primary looking for files to be replicated, and that will create, update or delete replicated files on the standby. When `ssh` is in use, `ssh` will log in as this user on the standby node involved in replication.

The user chosen as *repluser* should have Oracle ASM administrator privileges. The user specified to the Oracle installer when the Oracle software was first installed usually belongs to the needed groups, so it is convenient to choose this user as the replication user. In this discussion, the replication user is identified as *repluser*; however, you would replace *repluser* with the actual user name that you have selected. For information about running Oracle ACFS `acfsutil` commands, refer to About Using Oracle ACFS Command-Line Tools.

Note:

The same user and group identities must be specified for *repluser* on both your primary cluster and your standby cluster. Additionally, the mappings between user names and numeric `uids`, and between group names and numeric `gids`, must be identical on both the primary cluster and the standby cluster. This is required to ensure that the numeric values are used in the same manner on both clusters because replication transfers only the numeric values from the primary to standby.

Choosing a Transport for Oracle ACFS Replication

Starting with version 23ai of the Oracle Grid Infrastructure software, a new transport has been introduced for communication between the primary and standby replication sites. The new transport is based on Secure Sockets Layer (SSL) instead of Secure Shell (ssh).

On OL8 / X64 platforms, the user can now choose either to use SSL-based replication, or to continue using ssh-based replication. The use of ssh continues to be fully supported. On platforms other than OL8 / X64, ssh is the only transport provided.

The transport to be used is normally chosen when replication is initiated. However, support is provided to update an existing replication relationship from ssh-based to SSL-based.

SSL-Based Replication

SSL-based replication provides new authentication and secure transport mechanisms. Networking using ssh is replaced by Posix sockets. Authentication using ssh host and user keys is replaced by OpenSSL and Oracle wallets. Encryption and message authentication

(HMAC) code is replaced by Intel IPPS cryptography primitives, on platforms where they are available.

From a user standpoint, the biggest advantage of using SSL-based replication is simplified configuration. Instead of requiring host and user keys to be set up and distributed for each machine to be used in replication, SSL-based replication depends on establishing a single set of credentials shared between the primary and standby sites. This can be done trivially using command-line options, or (if desired) by manual communication of site credentials.

SSL-based replication will also perform better than ssh-based replication in most contexts.

SSH-Based Replication

The ssh transport in 23ai is the same transport provided by all past releases of snapshot-based replication. An enhancement in 23ai is a key setup assistant, `acfsreplssh`, that may be used to ease the configuration of host and user keys for machines to be used in replication. The use of `acfsreplssh` is recommended but not required.

Configuring SSL-Based Oracle ACFS Replication

Configuring SSL-Based Oracle ACFS Replication

This topic describes how to configure SSL-based replication in release 23ai or higher on OL8/ X64 platforms. The only configuration step needed is to ensure that the two replication sites share a set of credentials, as described below..

Initiating Replication

The use of SSL-based replication is specified with the option `-T ssl`, given to both `acfsutil repl init` commands. A corresponding option `-T ssh` specifies the use of ssh-based replication. The use of `ssh` is the default, so this latter option is not normally needed.

Credential Definition and Distribution

To use SSL-based replication, the user must distribute credentials to be shared between the primary and standby replication sites. This is analogous to (but much simpler than) setting up host and user keys in configuring ssh-based replication.

The credentials consist of an X.509 certificate and a public-private key pair. Credentials are established on one of the two replication sites, then securely copied to the other site. Thus SSL-based replication uses a shared-secret model for authentication, and both sides of a replication relationship (i.e. the primary and standby sites) use the same credentials. Since credentials are site-wide, all instances of replication active on a given site also use the same credentials. Credentials are created by any of these commands, unless credentials already exist on the local site:

- Explicit creation with `acfsutil repl update -o sslCreateCredentials`
- Implicit creation via `acfsutil repl init standby -T ssl`
- Implicit creation via `acfsutil repl init primary -T ssl`
- Implicit creation as part of updating ssh-based replication to be SSL-based, using `acfsutil repl update -T ssl`

The recommended way to create credentials is to do so on the standby site, as in this example of the `acfsutil repl init standby` command:

```
$ acfsutil repl init standby -T ssl -u repluser /standbyFS
```

This command will establish credentials if none exist on the standby site, or will re-use credentials that exist already. Once the credentials exist, there are three methods for distributing them to the primary site:

- Maximally secure - export at site A, secure copy to site B, import at site B
- Fairly secure - manual credential sync initiated on command line
- Fairly secure - automatic credential sync upon authentication failure

Export/import is the most secure method for distributing credentials. It is the default. The administrator exports the credentials to an export file at the local site, copies the file by a secure manner of her choosing ("sneaker net", scp, etc.) to a remote site, and then imports (installs) the credentials at the remote site.

Manual credential sync is fairly secure. The transfer of credentials is encrypted by OpenSSL but not authenticated. An administrator enables credential syncing at both replication sites using this command on both sites:

```
$ acfsutil repl update -o sslPermitCredentialSync
```

and then uses this command on the primary site:

```
$ acfsutil repl update -o sslSyncCredentials -s standby_address
```

to fetch the credentials from the standby site and import them at the primary site.

Lastly, if credential syncing is enabled as above, an automatic credential sync may occur if replication experiences an authentication failure during a replication operation. This is basically the same as manual credential syncing but is done automatically and seamlessly.

By default, credential syncing is disabled and can only occur if the administrator has explicitly enabled it at both sites. If credential syncing is permitted and a credential sync occurs, credential syncing is automatically disabled after the successful sync. This is to prevent a latent "open door" that would allow unintentional syncing.

Example: Initial Credential Distribution using Import / Export
In this example, the admin chooses the maximally secure (in this case "sneaker net") transfer of the credentials with their manual installation on the remote site. We assume that **boston-clu** is our standby cluster, and that **nashua-clu** is our primary cluster.

On cluster **boston-clu**, the admin initiates replication with the commands:

```
$ acfsutil repl init standby -T ssl -u repluser /standbyFS
$ acfsutil repl update -o sslExportCredentials=/mySecureThumbDrive/
mycredentials
```

The admin then hand-carries the secure thumb drive from Boston to Nashua. On cluster **nashua-clu**, the admin initiates replication with commands like these:

```
$ acfsutil repl update -o sslImportCredentials=/mySecureThumbDrive/
mycredentials
$ acfsutil repl init primary -T ssl -i 10m -s repluser@boston-clu -m /
standbyFS /primaryFS
```

Example: Manual Initial Credential Distribution

In this example, credentials at a standby site are copied and installed to a primary site at replication initialization time, via an explicit manual request. Once the credential transfer succeeds, credential syncing is automatically disabled and all subsequent interactions are authenticated using the installed credentials.

On cluster **boston-clu**, the admin allows credential distribution with a command like this one:

```
$ acfsutil repl update -o sslPermitCredentialSync
```

On cluster **nashua-clu**, the admin syncs credentials with commands like these:

```
$ acfsutil repl update -o sslPermitCredentialSync
$ acfsutil repl update -o sslSyncCredentials -s boston-clu
```

In this scenario, the second `acfsutil repl update` command at **nashua-clu** will sync credentials from **boston-clu**. Once the credentials are synced, credential syncing is automatically disabled. Then all future `acfsutil repl init` commands will use the credentials installed at each site.

So replication can be initialized as usual, with the addition of -`T ssl` to specify the transport. It's initialized first on the standby:

```
$ acfsutil repl init standby -T ssl -u repluser /standbyFS
```

And then on the primary:

```
$ acfsutil repl init primary -T ssl -i 10m -s repluser@boston-clu -m /
standbyFS /primaryFS
```

Example: Automated Initial Credential Distribution

In this example, credentials at a standby site are copied and installed to a primary site at replication initialization time, implicitly by the `acfsutil repl init` commands. Once the credential transfer succeeds, credential syncing is automatically disabled and all subsequent interactions are authenticated using the installed credentials.

On cluster **boston-clu**, the admin initiates replication with a command like this one:

```
 $ acfsutil repl init standby -T ssl -o sslPermitCredentialSync -u
repluser /standbyFS
```

On cluster **nashua-clu**, the admin initiates replication with a command like this one:

```
$ acfsutil repl init primary -T ssl -o sslPermitCredentialSync -i 10m -s
repluser@boston-clu -m /standbyFS /primaryFS
```

In this scenario, the `acfsutil repl init primary` command will encounter an authentication failure internally and will automatically sync credentials with the **boston-clu** site, with no human intervention required. Once the credentials are synced, credential syncing is automatically disabled.

Note that we also demonstrate in this example that if desired the `-o sslPermitCredentialSync` option can be included on the `acfsutil repl init` command lines. This is equivalent to specifying the option using separate `acfsutil repl update` commands.

Credential Updates
If the credentials in use by replication have been updated, the updated credentials can be distributed to the remote site by any of the methods mentioned above.

Example: Credential Update using Import / Export
An updated credential can be distributed by exporting and then importing the credentials as follows. At the replication site containing the updated credentials, the admin issues a command like:

```
$ acfsutil repl update -o sslExportCredentials=/mySecureThumbDrive/
mycredentials
```

to capture the credentials in a file. Then on the remote site, the file is used with a command like:

```
$ acfsutil repl update -o sslImportCredentials=/mySecureThumbDrive/
mycredentials
```

to import the credentials.

Example: Manual Credential Update
A credential update can be performed manually as follows. At both replication sites, the admin issues the command:

```
$ acfsutil repl update -o sslPermitCredentialSync
```

and then on the local site uses the command:

```
$ acfsutil repl update -o sslSyncCredentials -s remote-address
```

to fetch the credentials from the remote site and import them at the local site. Once this operation is complete, credential syncing is automatically disabled.

Example: Automatic Credential Update

A credential update can be performed automatically as follows. At both replication sites, the admin issues the command:

```
$ acfsutil repl update -o sslPermitCredentialSync
```

and then on the local site simply waits for replication to discover the update (i.e., receive an authentication failure), at which point it will sync credentials, disabling further syncing once this operation is complete.

Updating from SSH-Based Replication
An existing ssh-based instance of replication can be updated to use SSL as its transport. Once updated to use SSL, the instance of replication cannot revert to using ssh instead. This update is performed using the acfsutil repl update command, run on both the primary and the standby clusters.

You should update the standby cluster first, then the primary cluster. Just like when you specify SSL-based replication when initializing replication with `acfsutil repl init`, you can create and distribute credentials using import / export, manually or automatically.

In all cases, once the credentials have been synced, SSL takes over as the transport for the replication instance. Until that point, ssh remains in use as the transport.

Example: Update to SSL using Credential Import / Export
First, on the standby cluster **boston-clu**, the user creates credentials and updates to SSL-based replication, then exports the credentials just created, using these commands:

```
$ acfsutil repl update -T ssl /standbyFS
$ acfsutil repl update -o sslExportCredentials=/mySecureThumbDrive/
mycredentials
```

Next, on the primary cluster **nashua-clu**, the user imports the credentials and updates to SSL-based replication:

```
$ acfsutil repl update -o sslImportCredentials=/mySecureThumbDrive/
mycredentials
$ acfsutil repl update -T ssl /primaryFS
```

Example: Update to SSL using Manual Credential Syncing
Here we show an example of manual distribution, which depends on allowing credential syncing on both the primary and standby sites. First, on the standby cluster **boston-clu**, the user creates credentials and updates to SSL-based replication with the commands:

```
$ acfsutil repl update -o sslPermitCredentialSync
$ acfsutil repl update -T ssl /standbyFS
```

Then, on the primary cluster **nashua-clu**, the user enables syncing the credentials, updates to SSL-based replication, and finally performs the credential sync:

```
$ acfsutil repl update -o sslPermitCredentialSync
$ acfsutil repl update -o sslSyncCredentials -s boston-clu
$ acfsutil repl update -T ssl /primaryFS
```

Example: Update to SSL using Automated Credential Syncing
First, on the standby cluster **boston-clu**, the user creates credentials and updates to SSL-based replication with the commands:

```
$ acfsutil repl update -o sslPermitCredentialSync
$ acfsutil repl update -T ssl /standbyFS
```

Then, on the primary cluster **nashua-clu**, the user enables syncing the credentials and updates to SSL-based replication:

```
$ acfsutil repl update -o sslPermitCredentialSync
$ acfsutil repl update -T ssl /primaryFS
```

In this case, replication will discover the need to sync credentials and will perform the operation automatically.

Configuring `ssh` for Use With Oracle ACFS Replication

This topic describes how to configure `ssh` for use by Oracle ACFS snapshot-based replication available with release 12.2 or higher. Configuration can be performed in either of two ways:

- You can use the key setup assistant `acfsreplssh` to ease setting up the needed user and host keys
- Alternatively, you can configure the needed keys manually

To support the full capabilities of replication when ssh is used as its transport, `ssh` must be usable in either direction between the clusters — from the primary cluster to the standby cluster and from the standby to the primary. This means that passwordless operation of ssh must be enabled in both directions – that is, that host and user keys must be defined to allow ssh to connect from the local to the remote cluster with no manual intervention. These are the ssh keys that must be configured before replication is used:

- A public key for *repluser*, as defined on each node of the local cluster (both the primary and the standby), must appear in the `authorized_keys2` file for *repluser* on each node of the remote cluster (both standby and primary).
- A host key for each node of the remote cluster must appear in the `known_hosts` file on each node of the local cluster, unless replication is configured with strict host key checking disabled.

Configuring ssh Using the Key Setup Assistant

The ssh key setup assistant `acfsreplssh` may be used to configure, verify or delete the host and user keys required when `ssh` is used as the transport for Oracle ACFS snapshot-based replication. This topic describes how to use `acfsreplssh` to configure ssh for replication in one direction – from the primary to the standby. To configure ssh completely, you must perform these steps a second time with the primary and standby roles reversed.

To generalize our description, we will speak in terms of the local and the remote cluster. Run the command once with the primary as the local cluster, then a second time with the standby as the local cluster.

Command Synopsis

The command line interface for `acfsreplssh` looks like this:

```
acfsreplssh{ configure | verify | remove }
    [-v] [-m]
    [-p remote_password]
    [-V remote_vip]
    [-o sshStrictKey=ynvalue]
    { -c remote_cluster | remote_host1 [remote_hostn...] }
```

where the command-line options have these meanings:

`-c remote_cluster`

Uses *`remote_cluster`* as the name of a network endpoint to contact the remote cluster, then runs the acfsutil cluster info command on the remote cluster to identify the cluster members. Each hostname shown as a member is then processed as if it had been specified as a remote_host directly on the command line.

`-m`

Runs in "mockup mode" – shows what operations the command would perform if run normally, but does not run any of the operations. May be used with -v to see more details of what would happen.

`-o sshStrictKey=ynvalue` Specifies whether or not host keys must already exist. A ynvalue starting with "y" signifies "yes"; a ynvalue starting with "n" signifies "no". For the `configure` command, "yes" means that host keys must already exist (the command will not configure them), while "no" means the command will accept (and configure) any host key presented by a remote node. For the `verify` command, this option has no effect – a host key must already exist for all remote nodes. (The `verify` command never adds or modifies keys.) The default value of this option is "no" – that is, the `configure` command will configure any host key presented. See the `-o sshStrictKey` option to the `acfsutil repl init` and `acfsutil repl update` commands for more information on host key checking.

`-p remote_password`

Specifies remote_password as the password to be used in logging in as repluser on the remote cluster. Note that specifying the password on the command line in this way is not secure, and should be done only when there can be no security-related consequences. If -p is not given, `acfsreplssh` will prompt for the remote password (just once per invocation).

`-v`

Runs in "verbose mode" – shows details of the command's execution.

`-V remote_vip`

Identifies *remote_vip* as an existing VIP for the remote cluster, consisting of the set of *`remote_hosts`* named on the command line. This means that *`remote_vip`* will be added to the line in the local `known_hosts` file for each *`remote_host`* given on the command line.

> **Note:**
>
> All of the examples below show `-p` not being specified. That is, they assume that `acfsreplssh` will prompt for the remote cluster password each time the command is run.
> The assistant should be run on each node of the local cluster to operate on the host and user keys for a single remote cluster. It should be invoked as repluser. The assistant will output the identity of the invoking user for confirmation, and will then set up keys for that user to allow replication between the node where the assistant is run and the named remote cluster.
>
> To identify the hosts in the remote cluster, either:
>
> - Specify `-c` remote_cluster to name a network endpoint in the remote cluster, which will be queried to obtain the cluster members, or
> - Name each host in the remote cluster as a remote_host on the command line.

Configuring `ssh` on each local host

The `acfsreplssh configure` command is used to configure the ssh keys that will be required by replication. The command can be invoked either as part of setting up replication for the first time, or at some later point, to ensure that all the keys needed for replication are still present on the clusters involved. Each run of `acfsreplssh configure` – even multiple successive runs – will add a user or host key only where needed, preserving existing keys.

On each host, `acfsreplssh` will save the current key-related data for *repluser* before it modifies the data, unless this has been done already. That is, any pre-existent copy of the directory `~repluser/.ssh` will be renamed with the suffix `.acfsreplssh.backup`, unless a backup directory with this suffix already exists. Any directory created will have permissions 0400 (readable by owner only) and will be owned by *repluser*.

Here's an example. Let's say we have a 4-node primary cluster *bosc26*, with nodes *n1* though *n4*. We also have a 2-node standby cluster *chic26*, with nodes *n1* and *n2*. We have a SCAN VIP defined on each cluster, *bosc26vip* and *chic26vip*, and we wish to use the SCAN VIP on each cluster as the network endpoint for replication. To avoid having to specify each remote hostname on the command, we use `-c` with the VIP name as its value.

To configure the full set of keys required by replication, using the VIP defined on each cluster, run `acfsreplssh` as follows.

First, run this command on each node of *bosc26*:

```
$ acfsreplssh configure -V chic26vip -c chic26vip
```

Next, run this command on each node of *chic26*:

```
$ acfsreplssh configure -V bosc26vip -c bosc26vip
```

The first command above configures a public key for the current user (which is assumed to be *repluser*) on the host within *bosc26* where the command is run (if a key does not exist already), then adds the key to the `authorized_keys2` file for the current user on each host within *chic26*. Then a host key for each *chic26* host is added to the `known_hosts` file for the current user on the *bosc26* host, unless `-o sshStrictKey=yes` has been specified. If that option is given, then all needed host keys must be present already on the *bosc26* host.

With these commands, acfsutil cluster info will be run using chic26vip or bosc26vip, respectively, to determine the members of the remote cluster.

The second command performs the same operations, but in the opposite direction. That is, the command configures a public key for the current user on the host within *chic26* where the command is run (if a key does not exist already), then adds the key to the `authorized_keys2` file for the current user on each host within *bosc26*. Then a host key for each *bosc26* host is added to the `known_hosts` file for the current user on the *chic26* host, unless `-o sshStrictKey=yes` has been specified. If that option is given, then all needed host keys must be present already on the *chic26* host.

As an alternative example, let's look at how to deal with the same two clusters, but without a VIP defined for replication to use. Here, we use `-c` with one of the remote node names as its value. First, run this command on each node of bosc26:

```
# acfsreplssh configure -c chic26n1
```

Next, run this command on each node of *chic26*:

```
# acfsreplssh configure -c bosc26n1
```

With these commands, `acfsutil cluster info` will be run on *chic26n1* or *bosc26n1*, respectively, to determine the members of the remote cluster.

Verifying ssh configuration on each local host

The `acfsreplssh verify` command can be used to verify an existing set of keys. This command does not add or modify any keys – it simply tries to use the ones already present. To verify the keys that are present to support a given instance of replication, use the same form of command line as you used to configure those keys, just with the `verify` keyword specified.

To verify the full set of keys required by replication, using the VIP defined on each cluster, run `acfsreplssh` as follows.

First, to verify the keys needed for replication from the primary cluster to the standby cluster, run this command on each node of *bosc26*:

```
# acfsreplssh verify -V chic26vip -c chic26vip
```

Next, to verify the keys needed for potential future replication from the current standby to the current primary, run this command on each node of *chic26*:

```
# acfsreplssh verify -V bosc26vip -c bosc26vip
```

Alternatively, to verify the full set of keys required by replication, without using a VIP, run `acfsreplssh` as follows. First, to verify the keys needed for replication from the primary cluster to the standby cluster, run this command on each node of *bosc26*:

```
# acfsreplssh verify -c chic26n1
```

Next, to verify the keys needed for potential future replication from the current standby to the current primary, run this command on each node of *chic26*:

```
# acfsreplssh verify -c bosc26n1
```

Removing `ssh` configuration info on each local host

The `acfsreplssh remove` command can be used to remove an existing set of keys if (for instance) the user updates the value of *repluser* with the `acfsutil repl update` command. The `acfsreplssh remove` command must be run on each local host where keys are to be removed. The command will remove keys associated with the named remote cluster, as well. To remove the keys on a given local host that are present to support an old repluser, use the same form of command line as you used to configure those keys on that host, just with the `remove` keyword specified instead of `configure`.

For instance, to return to our previous example of replication involving the use of a VIP, you can remove the set of keys that were previously configured, using the VIP defined on each cluster, by running `acfsreplssh` as follows.

First, run this command on each node of *bosc26*:

```
$ acfsreplssh remove -V chic26vip -c chic26vip
```

Next, run this command on each node of *chic26*:

```
$ acfsreplssh remove -V bosc26vip -c bosc26vip
```

Configuring `ssh` manually

The procedures in this topic describe the manual steps needed to configure `ssh` for replication in one direction — from the primary to the standby. To configure `ssh` completely, you must perform the instructions a second time with the primary and standby roles reversed. When you perform the instructions the first time, complete the steps as written for the primary cluster and the standby cluster. The second time, reverse the primary and standby roles. Perform the steps marked as necessary on the *primary* cluster on your standby cluster and perform the steps marked as necessary on the *standby* cluster on your primary cluster. The procedures that must be performed twice are described in:

- Distributing keys for Oracle ACFS replication
- Getting a public key for *repluser* from the primary cluster
- Getting host keys for the standby cluster
- Notes on permissions for ssh-related files
- Notes on sshd configuration
- Validating your ssh-related key configuration

After you have completed all the necessary procedures, you can use the instructions described in Validating your ssh-related key configuration to confirm that you have configured `ssh` correctly in both directions.

Distributing keys for Oracle ACFS replication

The process of distributing keys for Oracle ACFS replication includes getting a public key from the primary cluster, getting host keys for the standby cluster, ensuring permissions are configured properly for `ssh`-related files, configuring `sshd` as necessary, and lastly validating the `ssh` configuration.

> **Note:**
>
> When creating host keys, ensure that you create keys for both fully-qualified domain hostnames and the local hostnames.

Getting a public key for *repluser* from the primary cluster

A public key for *repluser* defined on each node of your primary cluster must be known to *repluser* on each node of your standby cluster.

To make this key known, the directory `~repluser/.ssh` must exist on each standby node. If this directory does not exist, then create it with access only for *repluser*. Ensure that an `ls` command for the `.ssh` directory displays output similar to:

```
repluser@standby $ ls -ld ~/.ssh
drwx------ 2 repluser dba 4096 Jan 27 17:01 .ssh
```

If a public key file for *repluser* exists on a given primary node, then add its contents to the set of keys authorized to log in as *repluser* on each node of the standby where replication is run. Append the key to the file `~repluser/.ssh/authorized_keys2` on each standby node, creating this file if necessary.

If a public key file does not exist, generate a public and private key pair on the primary by running the following command as *repluser*.

```
$ ssh-keygen -t rsa
```

You can press the enter key in response to each prompt issued by the command. Copy the resulting `.pub` file to each standby node.

You have the option to share the same public/private key pair for *repluser* across all of the nodes in your primary cluster, or to establish a different key pair for each primary node. If the same public key is valid for *repluser* across all nodes in your primary cluster, then only that key must be added to the file `~repluser/.ssh/authorized_keys2` on each node of your standby cluster. If each primary node has its own public key for *repluser*, then all the public keys must be added to the file. In either case, you can minimize work by copying the updated `authorized_keys2` file on a given node of the standby to the other nodes of the cluster.

Getting host keys for the standby cluster

A host key for each standby node where replication may run must be known on each primary node where replication may run. One way to generate the correct key is to run `ssh` manually as *repluser* from each primary node to each standby node. If the correct host key is not known already, then a warning displays and you can enable `ssh` to add the key.

The following is an example of obtaining a host key:

```
[repluser@primary data]$ ssh repluser@standby date
The authenticity of host 'standby (10.137.13.85)' can't be established.
RSA key fingerprint is 1b:a9:c6:68:47:b4:ec:7c:df:3a:f0:2a:6f:cf:a7:0a.
Are you sure you want to continue connecting (yes/no)?
```

If you respond with `yes`, then the `ssh` setup is complete. A host key for host standby is stored in the `known_hosts` file (`~repluser/.ssh/known_hosts`) on the host *primary* for the user *repluser*.

After the host key setup for standby nodes is complete on a given primary node, you need to perform an additional step if you use a Virtual IP address (VIP) to communicate with your standby cluster. You must add the VIP name or address at the start of each line of the `known_hosts` file that refers to a host in the standby cluster. For example, if you use a VIP with the name `standby12_vip`, and your `known_hosts` file contains the following two lines that refer to your standby:

```
standby1,10.242.20.22 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC3pM2YTd4UUiEWEoCKDGgaTgsmPkQToDrdtU+JtVIq/96muivU
BaJUK83aqzeNIQkh+hUULsUdgKoKT5bxrWYqhY6AlTEqNgBHjBrJt9C73BbQd9y48jsc2G+WQWyuI/
+s1Q+hIJdBNMxvMBQAfisPWWUcaIx9Y/JzlPgF6lRP2cbfqAzixDot9fqRrAKL3G6A75A/6TbwmEW07d1zqOv
l7ZGyeDYf5zQ72F/V0P9UgMEt/5DmcYTn3kTVGjOTbnRBe4A4lY4rVw5c+nZBDFre66XtORfQgwQB5ztW/Pi
08GYbcIszKoZx2HST9AZxYIAgcrnNYG2Ae0K6QLxxxScP
standby2,10.242.20.23 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDIszcjzNtKN03SY8Kl846skFTVP1HF/ykswbmkctEjL6KTWTW+NR
U4MGbvkBqqdXxuPCR7aoGO2U3PEOg1UVf3DWUoux8IRvqKU+dJcdTibMFkDAIhTnzb14gZ/lRTjn+GYsuP5
Qz2vgL/U0ki887mZCRjWVL1b5FNH8sXBUV2QcD7bjF98VXF6n4gd5UiIC3jv6l2nVTKDwtNHpUTS1dQAi+1D
tr0AieZTsuxXMaDdUZHgKDotjciMB3mCkKm/u3IFoioDqdZE4+vITX9G7DBN4CVPXawp+b5Kg8X9P+08Eehu
tMlBJ5lafy1bxoVlXUDLVIIFBJNKrsqBvxxxpS7
```

To enable the use of the VIP, you would modify these two lines to read as follows:

```
standby12_vip,standby1,10.242.20.22 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC3pM2YTd4UUiEWEoCKDGgaTgsmPkQToDrdtU+JtVIq/96muivU
BaJUK83aqzeNIQkh+hUULsUdgKoKT5bxrWYqhY6AlTEqNgBHjBrJt9C73BbQd9y48jsc2G+WQWyuI/
+s1Q+hIJdBNMxvMBQAfisPWWUcaIx9Y/JzlPgF6lRP2cbfqAzixDot9fqRrAKL3G6A75A/6TbwmEW07d1zqOv
l7ZGyeDYf5zQ72F/V0P9UgMEt/5DmcYTn3kTVGjOTbnRBe4A4lY4rVw5c+nZBDFre66XtORfQgwQB5ztW/Pi
08GYbcIszKoZx2HST9AZxYIAgcrnNYG2Ae0K6QLxxxScP
standby12_vip,standby2,10.242.20.23 ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDIszcjzNtKN03SY8Kl846skFTVP1HF/ykswbmkctEjL6KTWTW+NR
U4MGbvkBqqdXxuPCR7aoGO2U3PEOg1UVf3DWUoux8IRvqKU+dJcdTibMFkDAIhTnzb14gZ/lRTjn+GYsuP5
Qz2vgL/U0ki887mZCRjWVL1b5FNH8sXBUV2QcD7bjF98VXF6n4gd5UiIC3jv6l2nVTKDwtNHpUTS1dQAi+1D
tr0AieZTsuxXMaDdUZHgKDotjciMB3mCkKm/u3IFoioDqdZE4+vITX9G7DBN4CVPXawp+b5Kg8X9P+08Eehu
tMlBJ5lafy1bxoVlXUDLVIIFBJNKrsqBvxxxpS7
```

Ultimately, the host key configuration performed on this first node of your primary cluster must be performed on every node in your primary cluster; the result of the above sequence, or an equivalent, must exist on each primary node. One way to minimize the manual effort required to achieve this configuration is to update the `known_hosts` file on one node of the primary cluster, then copy the updated file to the other nodes of the cluster.

> **Note:**
>
> By default, replication enables strict host key checking by `ssh`, to ensure that the primary node connects to the intended standby node or cluster when it runs `ssh`. However, if you are certain that this checking is unneeded, such as the case when the primary and standby clusters communicate over a private network, the use of strict host key checking by `ssh` can be disabled. For information about disabling strict host key checking, refer to the `-o sshStrictKey=no` option of the `acfsutil repl init primary` command. If strict host key checking is disabled, then no host key setup is required. For information about the `acfsutil repl init` command, refer to acfsutil repl init.

Notes on permissions for ssh-related files

For `ssh` to work with the keys you have established, you must ensure that permissions are set properly on each node for the `.ssh` directory for *repluser* and some of the files the directory contains.

For details on the permissions that should be given to each `.ssh` directory and key files within the directory, refer to the documentation for your `ssh` implementation, such as the `FILES` section of the `ssh(1)` manual page.

Notes on sshd configuration

After you begin using replication, `ssh` is started frequently to perform replication operations. On some platforms, the `ssh` daemon `sshd` may be configured to log a message through `syslog` or a similar facility each time an `ssh` connection is established. To avoid this, the server configuration file `/etc/ssh/sshd_config` can be modified to specify a lower frequency of logging. The parameter that controls logging is called `LogLevel`. Connection messages are issued at level `INFO`. Any lower `LogLevel` setting, such as `ERROR`, suppresses those messages. For example, you can suppress log messages by adding the following line to the file:

```
LogLevel ERROR
```

Validating your ssh-related key configuration

After you have established the host and user keys for `ssh` on both your primary and your standby clusters, you can use the command `acfsutil repl info -c -u` to validate the keys. You run this command as *repluser* on each node of each cluster. It takes as arguments all the hostnames or addresses on the remote cluster that the local cluster may use in the future to perform replication.

If you are not using a VIP to connect to your remote cluster, then for a given replication relationship, only one remote hostname or address is provided to `acfsutil repl init primary`. However, if future relationships involve other remote host addresses, specify the complete set of remote addresses when running the `acfsutil repl info -c -u` command.

If you are using a VIP to connect to your remote cluster, then you should specify the names or host-specific addresses of all remote hosts on which the VIP may be active. Do not specify the VIP name or an address associated with the VIP. When replication uses `ssh` to connect to a VIP, the host key returned is the key associated with the host where the VIP is currently active. Only the hostnames or addresses of individual remote nodes are used by `ssh` in this situation.

The validation command to run on each node of your primary cluster has the following format:

```
$ acfsutil repl info -c -u repluser standby1 [standby2 ...] [snap_shot@]primary-
mountpoint
```

In the command, `standbyn` specifies the standby cluster hostname or address. The validation command confirms that user *repluser* can use `ssh` to connect to each standby hostname or address given, in the same manner as replication initialization. Use the same command format if you are using a VIP, such as `standby12_vip`, to connect to the cluster. Do not specify the name of the VIP.

If you plan to disable strict host key checking, you can skip this checking by adding the `-o sshStrictKey=no` option to the command line.

After you have confirmed that each node of your primary cluster can connect to all nodes of your standby cluster, run the validation command again. This time run the command on each node of your standby cluster. Specify a hostname or IP address for all nodes of your primary cluster using the following format:

```
$ acfsutil repl info -c -u repluser primary1 [primary2 ...] [snap_shot@]standby-
mountpoint
```

In the command, `primaryn` specifies the primary cluster hostname or address.

Oracle Patching and Oracle ACFS

This section discusses patching with Oracle ACFS in a Grid Infrastructure environment.

- Overview of Oracle ACFS Patching
- Updating Oracle Grid Infrastructure Files
- Verifying Oracle ACFS Patching

Overview of Oracle ACFS Patching

Oracle ACFS is installed as part of Oracle Grid Infrastructure. However, Oracle ACFS runs from various system locations, such as `/lib/modules` and `/sbin` on Linux.

Oracle ACFS integrates with the Oracle Grid Infrastructure delivery and patch mechanisms: OUI and OPatch. Regardless of the delivery mechanism; Oracle Release, Oracle Patchset, Oracle Release Update, or Oracle One-off; Oracle ACFS content is delivered in patches.

When updating the Oracle Grid Infrastructure, without Oracle Zero Downtime Grid Infrastructure Patching, Oracle ACFS is also updated in the system locations, ensuring seamless operation of the Oracle Grid software. During the updates; whether Release, Release Update, Patchset, or One-off; the Oracle Clusterware stack is stopped on a local node and services are migrated to other nodes. The Oracle Grid Software is then patched and services are restarted on the local node.

Patching Without Oracle Zero Downtime Oracle Grid Infrastructure Patching

During the patch operation, Oracle ACFS software is updated first in the Grid Home by the OPatch or OUI file placement operation, and then later moved to the appropriate system locations and loaded into memory while Oracle Clusterware is down. The restart of Oracle Clusterware has the side effect of freeing up in operating system (OS) kernel references so that Oracle ACFS can be updated in the OS Kernel.

Patching With Zero Downtime Oracle Grid Infrastructure

When using Zero Downtime Oracle Grid Infrastructure Patching, only the Oracle Grid Infrastructure user space binaries in the Oracle Grid Home are patched. Commands which run out of the Oracle Grid Home immediately use the latest versions. Oracle Grid Infrastructure components that are installed outside of the Oracle Grid Home; such as ACFS, AFD, OLFS, and OKA OS system software (OS kernel modules and system tools); are updated in the Grid Home, but not installed to the system locations. They continue to run the version previous to the patch version. After patching, this results in the OPatch inventory displaying the new patch number in the OPatch inventory. However, the running software does not contain these changes, only the software that is available in the Grid Home. Until the newly available software is loaded into memory and accompanying user tools are copied to system locations, the system does not utilize the available fixes that are in the Oracle Grid Infrastructure Home.

To determine which Oracle ACFS system software is running and installed, the following commands can be used:

- `crsctl query driver activeversion -all`

  This command shows the Active Version of Oracle ACFS on all nodes of the cluster. The Active Version is the version of the Oracle ACFS Driver that is currently loaded and running on the system. This also implicitly indicates the ACFS system tools version. The `crsctl query` command, available from 18c and onwards, shows data from all nodes of the cluster.

  In the following example, 19.4 is available in the Oracle Home, but 19.2 is the current running version. OPatch `lsinventory` reports 19.4 as the patched version. Oracle Grid Infrastructure OS drivers are only running 19.2.

  ```
  crsctl query driver activeversion -all
  Node Name : node1
  Driver Name : ACFS
  BuildNumber : 200114
  BuildVersion : 19.0.0.0.0 (19.2.0.0.0)
  ```

- `crsctl query driver softwareversion -all`

  This command shows the available Software Version of the Oracle Grid Infrastructure software (and by extension, the available Software Version of the Oracle ACFS Software) that is currently installed in the Oracle Grid Home. The `crsctl query` command, available from 18c and onwards, shows data from all nodes of the cluster.

  ```
  crsctl query driver softwareversion -all
  Node Name : node1
  Driver Name : ACFS
  BuildNumber : 200628
  BuildVersion : 19.0.0.0.0 (19.4.0.0.0)
  ```

- `acfsdriverstate version -v`

  This command shows the full information on the running Oracle ACFS modules on the local node. The ACFS-9548 and ACFS-9547 messages displays the version of the Oracle ACFS software that is available in the Oracle Grid Infrastructure home.

`acfsdriverstate` reports on the local node only. Bug numbers are only available when running one-off patches.

```
acfsdriverstate version -v
ACFS-9325: Driver OS kernel version = 4.1.12-112.16.4.el7uek.x86_64.
ACFS-9326: Driver build number = 200114.
ACFS-9212: Driver build version = 19.0.0.0.0 (19.2.0.0.0).
ACFS-9547: Driver available build number = 200628.
ACFS-9548: Driver available build version = 19.0.0.0.0 (19.2.0.0.0)
ACFS-9549: Kernel and command versions.
Kernel:
 Build version: 19.0.0.0.0
 Build full version: 19.2.0.0.0
 Build hash: 9256567290
 Bug numbers: NoTransactionInformation
Commands:
 Build version: 19.0.0.0.0
 Build full version: 19.2.0.0.0
 Build hash: 9256567290
 Bug numbers: NoTransactionInformation
```

Updating Oracle Grid Infrastructure Files

Until the Oracle Clusterware stack is stopped and the Oracle ACFS driver modules are updated, Oracle ACFS fixes are not loaded into memory. The process that loads the Oracle ACFS fixes into system memory also installs the required tools for Oracle ACFS operation into system locations.

You can perform one of the following procedures:

1. To load Oracle ACFS fixes into memory and system locations, the following commands must be issued on a node by node basis:
   - `crsctl stop crs -f`
     Stops the CRS stack and all applications on the local node
   - `root.sh -updateosfiles`
     Updates Oracle ACFS and other Oracle Grid Infrastructure Kernel modules on the system to the latest version
   - `crsctl start crs -wait`
     Restarts CRS on the node
2. Alternatively, if a node reboots with a kernel version change, then newer drivers are automatically loaded and newer system tools installed into the system directories. It is assumed that all nodes in the cluster change kernel versions at the same time.

After one of these events has occurred, the `crsctl query activeversion` and `crsctl query softwareversion` commands report the same information: the loaded and running operating system (OS) software is the same as the latest available in the Oracle Grid Infrastructure Home. You can run other Oracle ACFS version commands as described in Verifying Oracle ACFS Patching.

Verifying Oracle ACFS Patching

When using standard OPatch patches to apply Oracle Release Updates and Patches, the inventory accurately reflects what is installed in the Grid Infrastructure home and on the system. For example:

```
[grid@racnode1]$ opatch lsinventory
...
..
Oracle Grid Infrastructure 19c                                      19.0.0.0.0
There are 1 products installed in this Oracle Home.

Interim patches (5) :

Patch  30501910: applied on Sat Mar 07 15:42:08 AEDT 2020
Unique Patch ID:  23299902
Patch description:  "Grid Infrastructure Jan 2020 Release Update :
19.4.0.0.200628 (30501910)"
   Created on 28 Dec 2019, 10:44:46 hrs PST8PDT
   Bugs fixed:
```

The output in the `lsinventory` example lists the OPatch RU and other patches that are applied, as well as the bug numbers and other information. These patches are applied to the Grid Infrastructure home. During normal patching operations, they are also applied to the operating system (OS) locations and loaded into memory, ensuring that Oracle Grid Infrastructure OS system software fixes are in sync with the Grid Infrastructure home. However, when using Zero Downtime Grid Infrastructure Patching, the content for Oracle Grid Infrastructure system software installed on the system, such as Oracle ACFS, is not updated at the same time.

The `crsctl query driver` and `acfsdriverstate` commands can be used to verify whether the installed Oracle Grid Infrastructure system software level is the same as the software level of the Grid Infrastructure home. Refer to the discussion about Zero Downtime Oracle Grid Infrastructure patching in Overview of Oracle ACFS Patching.

For patching and update operations applied without Zero Downtime Oracle Grid Infrastructure Patching, the active and software version should always be the same.

If it is necessary to install updated Oracle Grid Infrastructure OS system software, refer to the procedures in Updating Oracle Grid Infrastructure Files.

After all the Oracle Grid Infrastructure OS system software is updated, the version should be the same as the Opatch `lsinventory` output displayed for any patches or updates to the Grid Infrastructure home, in this case, 19.4.0.0.0. Additionally, the Oracle Grid Infrastructure OS system software that is available and active should have the same version number displayed. For example:

```
Output from the lsinventory command:
Patch description:  "Grid Infrastructure Jan 2020 Release Update :
19.4.0.0.0.200628 (30501910)"

crsctl query driver activeversion -all
Node Name : node1
Driver Name : ACFS
BuildNumber : 200628
BuildVersion : 19.0.0.0.0 (19.4.0.0.0)

crsctl query driver softwareversion -all
Node Name : node1
Driver Name : ACFS
BuildNumber : 200628
BuildVersion : 19.0.0.0.0 (19.4.0.0.0)
```

You can run the `acfsdriverstate version` command for additional Oracle ACFS information on the local node, including information on commands and utilities. For example:

```
acfsdriverstate version
 ACFS-9325: Driver OS kernel version = 4.1.12-112.16.4.el7uek.x86_64.
 ACFS-9326: Driver build number = 200628.
 ACFS-9212: Driver build version = 19.0.0.0.0 (19.4.0.0.0)
 ACFS-9547: Driver available build number = 200628.
 ACFS-9548: Driver available build version = 19.0.0.0.0 (19.4.0.0.0).
```

Glossary

ASMLIB

ASMLIB is an application programming interface (API) developed by Oracle to simplify the operating system–to-database interface and to exploit the capabilities of vendors' storage arrays on Linux-based systems. Oracle ASM Filter Driver is the recommended replacement for ASMLIB.

Cluster File System (CFS)

A Cluster File System, or CFS, is a file system that is distributed across multiple computer systems. Generally, the computer systems share access to a disk connected through a Storage Area Network (SAN). The CFS component on each individual computer system coordinates access to the disks using a global communication mechanism.

Cluster Synchronization Services (CSS)

Cluster Synchronization Services (CSS) provide the foundation for coordinated, inter-process communication activities across a set of nodes. CSS services include group services, lock services, node information and cluster configuration services, and node monitoring services that detect nodes joining and leaving the cluster.

CSS Cluster

A CSS cluster is the cluster configuration defined by the CSS cluster membership services. CSS maintains the authoritative membership for an Oracle cluster and monitors nodes joining and leaving the cluster configuration. There is at most one CSS cluster defined for a given set of nodes.

disk group

An Oracle ASM disk group is a collection of disks that Oracle ASM manages as a unit. Within a disk group, Oracle ASM exposes a file system interface for Oracle Database files. The content of files that are stored in a disk group are evenly distributed, or striped, to eliminate hot spots and to provide uniform performance across the disks. Oracle ASM files may also be optionally mirrored within a disk group. The performance of disks in a disk group is comparable to the performance of raw devices.

Direct Attached Storage (DAS)

Direct Attached Storage, or DAS, consists of storage devices that attach directly to host without an intervening network. DAS generally costs less than SAN attached storage, but also offers less flexibility and functionality. Examples of DAS include SATA disks, common to most desktop systems, and SCSI disks that are found on many servers.

extent

An extent, also called data extent, is the raw storage used to hold the contents of a file. Each extent consists of one or more allocation units (AU) on a specific disk.

extent map

An extent map is a list of extent pointers that point to all the data extents of a file. This is the raw storage for the file. Each extent pointer gives the disk and allocation unit of the data extent. For reliability, each extent pointer also includes a check byte to ensure it has not been damaged. This is particularly important when using an in-memory copy of the extent map to direct file I/O to a particular disk location.

fibre channel

Fibre channel is a set of standards that define the interface between computer systems and peripherals. The fibre channel interface is a serial bus interface originally designed to supersede the SCSI standard. However, both the fibre channel and SCSI interfaces have evolved independently of each other and have benefited from the existence of the other. Fibre is spelled with **re** rather than an **er** to indicate a distinction from terms such as *fiber optics*. Both copper and optical fiber are commonly used as a media for fibre channel.

file system

A file system is a software component providing structured access to disks. File systems present objects, such as files, to application programs. Access to files is generally specified with standard API defining operating system calls such as Open/Close and Read/Write that the application program uses for accessing files. File systems are usually provided as a component of an operating system, but may be provided as an independent software component.

file

A file is an object presented to an application program by a file system. A file is subdivided into blocks by the file system. A file system typically places what appears to the application program as consecutive blocks, into nonconsecutive locations on disks. The mapping of the file blocks to the locations on disks is kept in what is known as an extent map.

host bus adapter (HBA)

A host bus adapter (HBA), or host adapter, connects a host system to other network and storage devices, such as, eSATA, and SCSI devices.

Logical Unit Number (LUN)

In storage technology, the term LUN is often used to denote a disk presented to a computer system by a storage array. In SCSI terminology, a Logical Unit Number, or LUN, is a number representing a device address presented to a computer system.

mirroring

In storage systems, mirroring is a means for protecting the integrity of data by storing copies of data on multiple disks. If a disk fails, then a secondary copy of the data is available on a second or third disk. Generally, mirroring can be deployed in one of two ways. In the most common case, as with a RAID storage array, a fixed amount of physical storage space on a single disk is uniformly copied on multiple disks. Through software on the storage array, the physically mirrored space on multiple devices is presented to the host as a single disk. Another approach to mirroring is to copy fixed segments of a file onto multiple disks. Oracle ASM uses the second approach for maximum flexibility.

Network Attached Storage (NAS)

Network Attached Storage, or NAS, comprises a class of systems providing file services to host computers. A device driver in the operating system accesses data using protocols such as NFS or Common Internet File System (CIFS). From the application program's point of view, Network Attached Storage appears as a file system local to the host computer on which the application program resides.

Network File System (NFS)

A network file system, or NFS, is an industry standard providing Network Attached Storage on many platforms. NFS extends the local file system framework to provide device sharing to users on an IP network. The user view of an NFS is that a remote file system on a host appears as if it were locally mounted.

Oracle Automatic Storage Management Cluster

An Oracle Automatic Storage Management (Oracle ASM) cluster is a collection of interconnected nodes, each with an Oracle ASM instance, operating as a unified cluster using Oracle Clusterware. An Oracle ASM cluster presents a shared pool of storage to one or more Oracle Databases that are also operating on the nodes. The databases can also be clustered using Oracle Real Application Clusters, but that is not a requirement. The disks that provide the shared storage pool must be accessible from all of the nodes in the Oracle ASM cluster.

partition

Operating systems typically provide a means for splitting a disk into sections called partitions. To an application, a partition appears as an independent disk, even though that is not the case. The operating system provides a command for managing the size and locations of partitions on a disk. This command writes a partition map onto the disk in an area that is not visible to the application program.

Redundant Array of Inexpensive Disks (RAID)

Redundant Array of Inexpensive Disks, or RAID, is a means for combining multiple disks through software and presenting them to a host as a collection of apparently distinct disks. Mirroring and striping are two means used for combining the disks as a collection. RAID can be implemented as a software component of a computer system or as software operating inside of a storage array. RAID operates on the physical presentation of storage as opposed to Oracle ASM mirroring and striping at the logical file level.

Storage Area Network (SAN)

Storage Area Network, or SAN, is a specialized network designed for attaching storage devices, such as disk arrays and tape drives, to computer systems. The most common network infrastructure used in SANs is fibre channel. SANs provide great flexibility by allowing a common interconnect infrastructure in which any storage device can be connected to any computer system. Zoning and LUN masking are commonly used for access control for SANs. Zoning is a mechanism where the SAN infrastructure determines the access rights of storage devices to computer systems. LUN masking is a mechanism where the storage device determines the access rights of what computer systems are permitted to connect to the device.

Small Computer System Interface (SCSI)

Small Computer System Interface, or SCSI, is a standard defining the interface between computer systems and peripherals, most commonly storage devices. The SCSI standard defines both the logical and physical aspects of the interface between computer systems and peripherals.

striping

In storage systems, striping is a means for spreading data across multiple disks as opposed to storing the data on a single disk. Striping is usually done to improve performance. Generally, striping can be deployed in one of two ways. In the most common case, as with a RAID storage array, a fixed amount of physical storage space that could have been stored on a single disk is uniformly striped across multiple disks. Through software on the storage array, the physically striped space on multiple devices is presented to the host as a single disk. Another approach to striping is to stripe fixed segments of a file across multiple disks connected to the host. Oracle ASM uses the second approach for maximum performance.

volume

In the storage realm, the meaning of volume has many related definitions. Volumes are often thought of as the virtual entity represented as a Logical Unit Number (LUN). Volumes often are presented as an aggregation of pieces from several disks. A volume is managed by a software component called a volume manager.

volume manager

A volume manager is a software component that manages the mapping of the collection of the pieces of the disks into a volume.

Index

Symbols

A

B

C

D

P

T

U

V