Oracle Linux 9

Managing the Ext File System

G26932-01
July 2025

Oracle Linux 9 Managing the Ext File System,

G26932-01

Copyright © 2025, Oracle and/or its affiliates.

Contents

Preface

Oracle Linux 9: Managing the Ext File System provides information about managing the Ext file system on Oracle Linux 9 systems.

Documentation License

The content in this document is licensed under the Creative Commons Attribution–Share Alike 4.0 (CC-BY-SA) license. In accordance with CC-BY-SA, if you distribute this content or an adaptation of it, you must provide attribution to Oracle and retain the original copyright notices.

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.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and

the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About the Ext File System

The extended file system, or Ext, is the first file system that was written for the Linux kernel and is in common usage across many Linux distributions. Ext has evolved through several successive updates and is available as the Ext4 file system, which is largely backward compatible with previous Ext file system releases but includes many added features. Key features available in Ext4, include:

- Large file system support: Ext4 can theoretically support volumes with sizes up to 1 EiB and single files with sizes up to 16 TiB.
- Use of extents instead of block mapping: improves large file performance and reduces fragmentation.
- Recognizes `fallocate` for persistent preallocation of on-disk space for a file: improves performance and helps to ensure contiguous disk allocation for a file.
- Use of allocate-on-flush: helps with performance and reduces fragmentation by delaying disk space allocation until the moment that data is flushed to disk.
- Checksum functionality: ensures data integrity.

For more information, visit https://docs.kernel.org/admin-guide/ext4.html.

For an overview of local file system management, see Oracle Linux 9: Performing File System Administration.

2

Installing Ext File System Utilities

Ext file system utilities are provided in the `e2fsprogs` package.

On most Oracle Linux systems, the `e2fsprogs` package is already installed, however you can use DNF to install it if it isn't available or to update the current package to the latest version.

The `e2fsprogs` package includes the `mkfs` commands to format a device with any of the available Ext file system versions, and also commands to perform actions including file system checks and tuning.

- Use the `dnf` command to install or update the `e2fsprogs` package.

  ```
  sudo dnf install -y e2fsprogs
  ```

3

Setting Up and Administering an Ext File System

Tools to create and administer Ext file systems, include the `mkfs.ext4`, `dumpe2fs`, `tune2fs`, `e4dfrag`, and `resize2fs` commands.

Creating and Mounting an Ext File System

Use the `mkfs.ext4` command to create an XFS file system on a block device, such as a partition, an LVM volume, a disk, or a similar hardware device.

You create Ext file systems by using the `mkfs.ext4` command. The default options for the command are appropriate for most common use cases. For more information, see the `mkfs.ext4(8)` manual page.

Note that you can also format a device with Ext 2 or Ext 3 by using the `mkfs.ext2` or `mkfs.ext3` commands, instead.

1. Identify the target device, partition, or file that you want to format with Ext.

   Typically, you might use the `lsblk` command to list block devices and partitions that are available to the system.

   Note that formatting a file system is a destructive operation and erases any data on the target device. Ensure that the following steps use the correct target device or file paths.

2. To create an Ext4 file system on a device, run:

   ```
   sudo mkfs.ext4 [options] <device>
   ```

   For example, to format a file system on the device */dev/sdb1* with the default options, run:

   ```
   sudo mkfs.ext4 /dev/sdb1
   ```

   Use the appropriate flags to set any custom options on the file system if you need to do so.

3. Mount the file system.

   For example, you can run:

   ```
   sudo mount /dev/sdb /mnt
   ```

4. Validate the file system, by checking file system information with the `dumpe2fs` command.

   For example, you can run:

   ```
   dumpe2fs -h /dev/sdb
   ```

Checking and Repairing an Ext File System

Use the `fsck` utility to check and repair file systems.

For file systems other than `root` (`/`) and `/boot`, `mount` invokes file system checking if more than a specified number of mounts have occurred or more than 180 days have elapsed without checking having being performed. You can run `fsck` manually if a file system hasn't been checked for several months.

For more information, see the `fsck(8)` manual page.

The following procedure describes how to check and repair an Ext file system.

> **NOT_SUPPORTED:**
>
> Running `fsck` on a mounted file system can corrupt the file system and cause data loss.

1. Unmount the file system:

   ```
   sudo umount filesystem
   ```

2. Use the `fsck` command to check the file system:

   ```
   sudo fsck [-y] file-system
   ```

   The *file-system* specifies a device name, a mount point, a label, or UUID specifier, for example:

   ```
   sudo fsck UUID=ad8113d7-b279-4da8-b6e4-cfba045f66ff
   ```

   By default, the `fsck` command prompts you to confirm whether to apply a suggested repair to the file system. If you specify the `-y` option, the command assumes a yes response to all such questions.

For the `ext3`, and `ext4` file system types, other commands that are used to perform file system maintenance include `dumpe2fs` and `debugfs`. `dumpe2fs` prints super block and block group information for the file system on a specified device. `debugfs` is an interactive file system debugger that requires expert knowledge of the file system architecture.

Configuring Automatic File System Checking for Ext File Systems

Configure automatic file system checking on an Ext file system by using the `tune2fs` command.

For file systems other than `root` (`/`) and `/boot`, `mount` invokes file system checking if more than a specified number of mounts have occurred or more than 180 days have elapsed without checking having being performed. Mount-count dependent checking is disabled by default.

The following procedure applies to Ext file systems *only*. XFS file systems, which are the default file system type in Oracle Linux, detect errors automatically and don't require periodic file system checks at boot time.
For more information, see the `tune2fs(8)` manual page.

- Change the number of mounts before the system automatically checks the file system for consistency.

  ```
  sudo tune2fs -c mount_count device
  ```

  The *device* specifies the block device that corresponds to the file system.

  A *mount_count* of `0` or `-1` disables automatic checking, based on the number of mounts.

  > **Tip:**
  >
  > Specifying a different *mount_count* value for each file system reduces the probability that the system checks all the file systems at the same time.

- Change the interval between file system checks:

  ```
  sudo tune2fs -i interval[unit] device
  ```

  A *unit* can be `d` for days, `w` for weeks, or `m` for months.

  The default unit is `d` (for days). An *interval* of `0` disables checking, based on the time that has elapsed after the last check. Even if the interval is exceeded, the file system isn't checked until it's next mounted.

Defragmenting an Ext4 File System

You can defragment the file system to increase I/O performance.

Ext file systems can become fragmented over time, leading to performance degradation. Defragmentation helps to ensure that files use contiguous disk blocks which improves overall system efficiency.

1. Identify fragmented files by using the `e4defrag -c` command.

   For example, to view the fragmentation on the device `/dev/sda1`, run:

   ```
   sudo e4defrag -c /dev/sda1
   ```

2. Defragment files, directories, or devices with the `e4defrag -v` command:

   To defragment specific files or directories, you can run:

   ```
   sudo e4defrag -v /path/to/file_or_directory
   ```

   You can also defragment an entire file system by specifying the device path:

   ```
   sudo e4defrag -v /dev/sda1
   ```

3. Verify defragmentation status by running the `e4defrag -c` command again.

Resizing an Ext File System

Resize an Ext file system by using the `resize2fs` command.

Before resizing an Ext file system, the underlying block device, volume, or file image must have appropriate space to hold the file system if you're increasing the file system size.

Note that you can increase the size of an Ext file system without unmounting the file system first, but decreasing the size of the file system requires that you unmount the file system first. In general, it's good practice to unmount the file system, if possible, regardless of the operation.

See the `resize2fs(8)` manual page for more information.

1. Unmount the file system that you intend to resize.

   ```
   sudo umount /mnt
   ```

   Although not necessary when increasing the file system size, unmounting any file systems on the device ensures data integrity during the resizing process:

2. Check file system integrity.

   ```
   sudo e2fsck -f /dev/sda1
   ```

3. Resize the file system by using the `resize2fs` command.

   ```
   sudo resize2fs /dev/sda1
   ```

   If you don't specify a size parameter, the file system is sized to match the space on the underlying block device. To specify a size, you can use unit suffixes to allocate space in kilobytes (K), megabytes (M), or gigabytes (G). For example:

   ```
   sudo resize2fs /dev/sda1 120G
   ```

4. Remount the file system.

   If you unmounted the file system earlier, remount it now:

   ```
   mount /dev/sda1 /mnt
   ```

5. Verify the new file system size using the `df` command.

```
df -h /mnt
```

4

Convert an Ext2 or Ext3 File System to Ext4 In-Place

Convert an existing Ext2 or Ext3 file system to Ext4 without losing data.

Ext4 is an extension to Ext3, which in turn builds on features in Ext2. You can convert an earlier file system to a later version by enabling the features that are required for that version and then mounting it using the correct version type. The primary tool used for these changes is the `tune2fs` command.

> **Note:**
>
> The preferred method for upgrading an Ext2 file system to Ext4 is to back up the entire volume, reformat the storage device with Ext4, and restore the entire volume onto the newly formatted file system.

Always back up data before making file system modifications.

This procedure describes the steps to advance the file system from Ext2 to Ext4. If the file system that you're converting is an Ext3 file system, you can skip the intermediate step that converts the Ext2 file system to Ext3.

1. Unmount the file system to prevent data corruption.

   ```
   sudo umount /dev/sda1
   ```

2. Convert the file system from Ext2 to Ext3 by enabling the journal feature.

   ```
   sudo tune2fs -j /dev/sda1
   ```

3. Convert the file system from Ext3 to Ext4 by enabling key Ext4 features.

   ```
   sudo tune2fs -O extents,uninit_bg,dir_index /dev/sda1
   ```

4. Check the file system for errors.

   ```
   sudo e2fsck -f /dev/sda1
   ```

5. Edit the `/etc/fstab` file to change the file system type to `ext4` if the file system had a previous fstab entry.

   For example, you might change the entry to read:

   ```
   /dev/sda1        /mnt       ext4    defaults  1 1
   ```

6. Remount the file system as Ext4.

   ```
   sudo mount -t ext4 /dev/sda1 /mnt
   ```

7. Verify the file system type to ensure that it converted correctly.

   For example, you can check the file system type by running:

   ```
   sudo blkid /dev/sda1
   ```

5

Working With Disk Quotas

You can set disk quotas to restrict the amount of disk space or *blocks* that users or groups can use, to limit the number of files or *inodes* that users or groups can create, and to notify you when usage is reaching a specified limit. A hard limit specifies the maximum number of blocks or inodes that are available to a user or group on the file system. Users or groups can exceed a soft limit for a period, which is known as a *grace period*.

Enabling Disk Quotas on File Systems

Disk quota types are enabled at mount by specifying a mount option.

Mount Option	Description
`gqnoenforce`	Enable group quotas. Report usage, but don't enforce usage limits.
`gquota`	Enable group quotas and enforce usage limits.
`pqnoenforce`	Enable project quotas. Report usage, but don't enforce usage limits.
`pquota`	Enable project quotas and enforce usage limits.
`uqnoenforce`	Enable user quotas. Report usage, but don't enforce usage limits.
`uquota`	Enable user quotas and enforce usage limits.

Mounting a File System With Quotas Enabled

Mount a file system from the command line with a quota type enabled.

If a file system doesn't have a system mount configured in `/etc/fstab` or the entry doesn't include a quota option, you can enable the quota option when you mount the file system from the command line.

- Mount the file system from the command line using the `-o <quotatype>` option to enable the specified quota.

  For example, to enable user quotas, run:

  ```
  sudo mount -o uquota /dev/sdb1 /mnt
  ```

  Replace `uquota` with `uqnoenforce` to enable usage reporting without enforcing any limits.

Editing System Mounts to Use Quotas

Edit /etc/fstab to add quota options to a file system entry, to enable quotas when the file system is remounted.

1. Install the `quota` package on the system, if not already installed.

   ```
   sudo dnf install -y quota
   ```

2. Add the quota type options to the file system's `/etc/fstab` entry.

   For example, to add the `uquota` and `gquota` types, you can add:

   ```
   /dev/sdb1    /home     ext4    defaults,uquota,gquota   0 0
   ```

3. Remount the file system.

   ```
   sudo mount -o remount /home
   ```

Assigning Disk Quotas to Users and Groups

Use the `edquota` or `setquota` command to assign disk quotas to users and groups.

For more information, see the `edquota(8)` and `setquota(8)` manual pages.

1. Use the `edquota` command to configure quota limits for a user or group within a text editor.

   For a user, use the following command:

   ```
   sudo edquota username
   ```

   For a group, use the following command:

   ```
   sudo edquota -g group
   ```

   The command opens a text file in the default editor that's defined by the `EDITOR` environment variable. You can specify the limits for the user or group, for example:

   ```
   Disk quotas for user guest (uid 501)
   Filesystem  blocks  soft  hard  inodes  soft  hard
    /dev/sdb1   10325     0     0    1054     0     0
   ```

   The `blocks` and `inodes` entries reflect the user's current usage on a file system.

   > **Tip:**
   >
   > Setting a limit to `0` disables quota checking and enforcement for the corresponding `blocks` or `inodes` category.

   Edit the soft and hard block limits for the number of blocks and inodes, then save the changes.

2. Use the `setquota` command to configure quota limits from the command line.

To configure a user quota, use the `-u` option and then configure the soft and hard block and inode limits. For example, you can set a soft block limit to 500 Mb and a hard block limit to 1 Gb, while disabling inode limits as follows:

```
sudo setquota -u username 500M 1G 0 0 /mnt
```

To configure a group quota, use the `-g` option and then configure the soft and hard block or inode limits. For example, you can disable block limits for the group but set a soft inode limit to 9000 and a hard inode limit to 10000 .

```
sudo setquota -g groupname 0 0 9000 10000 /mnt
```

Setting Up Project Quotas

Configure project quotas to apply a quota to individual directory hierarchies.

Project quotas can be set on individual directory hierarchies, which are known as *managed trees*. Each managed tree is uniquely identified by a *project ID* and an optional *project name*. Projects are defined in the `/etc/projects` and `/etc/projid` configuration files. For more information, see the `projects(5)` and `projid(5)` manual pages.

The ability to control the disk usage of a directory hierarchy is useful if you don't otherwise want to set quota limits for a privileged user, for example, `/var/log`, or if many users or groups have write access to a directory, for example, `/var/tmp`.

To define a project and set quota limits for it:

1. Mount the file system with project quotas enabled.

   ```
   sudo mount -o pquota device mountpoint
   ```

   For example, to enable project quotas for the file system mounted at `/mnt`, you would use the following command:

   ```
   sudo mount -o pquota /dev/sdc1 /mnt
   ```

2. Define a unique project ID for the directory hierarchy in the `/etc/projects` file.

   ```
   sudo echo project_ID:mountpoint/directory |sudo tee -a /etc/projects
   ```

   For example, you would set a project ID of 51 for the directory hierarchy `/mnt/testdir` as follows:

   ```
   sudo echo 51:/mnt/testdir |sudo tee -a /etc/projects
   ```

3. Create an entry in the `/etc/projid` file that maps a project name to the project ID.

   ```
   sudo echo project_name:project_ID |sudo tee -a /etc/projid
   ```

For example, you would map the project name `testproj` to the project with ID 51 as follows:

```
sudo echo testproj:51 |sudo tee -a /etc/projid
```

Setting a Grace Period for Soft Limits

Set the grace period for soft limits by using the `edquota` command.

- Run the `edquota -t` command to set a grace period for soft limits.

  ```
  sudo edquota -t
  ```

  Running the command opens a text file in a default text editor. Edit the file to specify the appropriate grace period, as shown in the following example:

  ```
  Grace period before enforcing soft limits for users:
  Time units may be: days, hours, minutes, or seconds
    Filesystem     Block grace period     Inode grace period
    /dev/sdb1             7days                  7days
  ```

  Specify the grace periods for the soft limits on the number of blocks and inodes, then save the changes.

  For more information, see the `edquota(8)` manual page.

Displaying Disk Quotas

Use the `quota` command to view information about disk quotas across the system.

Running the quota command as a system administrator lets you see the quota information for all users and groups on the system.

Individual users can also use the `quota` command to display disk usage for themselves and their own group.

For more information, see the `quota(1)` manual page.

- To display a user's disk usage, use the `quota` command without any options and specifying the username as an argument.

  ```
  sudo quota username
  ```

- To display a group's disk usage, use the `quota -g` command.

  ```
  sudo quota -g group
  ```

- To display information about file systems, where usage is over the quota limits, use the `quota -q` command

  ```
  sudo quota -q
  ```

Enabling and Disabling Disk Quotas

Use the quotaoff and quotaon commands to disable and enable disk quotas.

For more information, see the `quotaon(8)` manual page. The following steps provide examples for enabling and disabling quotas.

- Disable disk quotas for all users, groups on a specific file system.

  ```
  sudo quotaoff -guv /mnt
  ```

- Disable disk quotas on all automatically mounted file systems for all users, groups.

  ```
  sudo quotaoff -aguv
  ```

- Reactivate disk quotas on all automatically mounted file systems for all users and groups.

  ```
  sudo quotaon -aguv
  ```

Reporting on Disk Quota Usage

Use the `repquota` command to report on disk quota usage.

For more information, see the `repquota(8)` manual page.

- Display the disk quota usage for a specific file system.

  ```
  sudo repquota /mnt
  ```

- Display the disk quota usage for all automatically mounted file systems.

  ```
  sudo repquota -a
  ```

Maintaining the Accuracy of Disk Quota Reporting

Rebuild the quota database with the `quotacheck` command to fix inaccuracies in disk quota reports.

Uncontrolled system shutdowns can lead to inaccuracies in disk quota reports. You can rebuild the quota database to fix these inaccuracies.

For more information, see the `quotacheck(8)` manual page.

1. Disable disk quotas for the file system.

   ```
   sudo quotaoff -guv /mnt
   ```

2. Unmount the file system.

   ```
   sudo umount /mnt
   ```

3. Rebuild the quota databases.

   ```
   sudo quotacheck -guv /mnt
   ```

4. Mount the file system.

   ```
   sudo mount /mnt
   ```

5. Enable disk quotas for the file system.

   ```
   sudo quotaon -guv /mnt
   ```