



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'ostree.1'***

**\$ man ostree.1**

OSTREE(1)                    ostree                    OSTREE(1)

NAME

ostree - Manage multiple bootable versioned filesystem trees

SYNOPSIS

ostree {COMMAND} [OPTIONS...]

DESCRIPTION

OSTree is a tool for managing multiple bootable versioned filesystem trees, or just "tree" for short. In the OSTree model, operating systems no longer live in the physical "/" root directory. Instead, they parallel install to the new toplevel /ostree directory. Each installed system gets its own /ostree/deploy/stateroot directory. (stateroot is the newer term for osname).

Unlike rpm or dpkg, OSTree is only aware of complete filesystem trees. It has no built-in knowledge of what components went into creating the filesystem tree.

It is possible to use OSTree in several modes; the most basic form is to replicate pre-built trees from a build server. Usually, these pre-built trees are derived from packages. You might also be using

OSTree underneath a higher level tool which computes filesystem trees locally.

It must be emphasized that OSTree only supports read-only trees. To change to a different tree (upgrade, downgrade, install software), a new tree is checked out, and a 3-way merge of configuration is performed. The currently running tree is not ever modified; the new tree will become active on a system reboot.

To see the man page for a command run `man ostree COMMAND` or `man ostree-admin COMMAND`

## OPTIONS

The following options are understood:

`--repo`

For most commands, a repository is required. If unspecified, the current directory is used if it appears to be an OSTree repository.

If it isn't, either the `OSTREE_REPO` environment variable is used, or the system repository located at `/sysroot/ostree/repo`.

`-v, --verbose`

Produce debug level output.

`--version`

Print version information, including the features enabled at compile time, and exit.

## COMMANDS

System administrators will primarily interact with OSTree via the subcommand `ostree admin`.

`ostree-admin-cleanup(1)`

Delete untagged deployments and repository objects.

`ostree-admin-config-diff(1)`

See changes to `/etc` as compared to the current default (from `/usr/etc`).

`ostree-admin-deploy(1)`

Takes a particular commit or revision, and sets it up for the next boot.

`ostree-admin-init-fs(1)`

Initialize a root filesystem in a specified path.

ostree-admin-instutil(1)

Utility functions intended primarily for operating system  
installation programs

ostree-admin-os-init(1)

Initialize the deployment location for an operating system with a  
specified name.

ostree-admin-status(1)

Show and list the deployments.

ostree-admin-switch(1)

Choose a different ref to track from the same remote as the current  
tree.

ostree-admin-undeploy(1)

Remove the previously INDEX deployed tree from the bootloader  
configuration.

ostree-admin-upgrade(1)

Download the latest version for the current ref, and deploy it.

Both administrators and operating system builders may interact with  
OSTree via the regular filesystem manipulation commands.

ostree-cat(1)

Concatenate contents of files

ostree-checkout(1)

Check out a commit into a filesystem tree.

ostree-checksum(1)

Gives checksum of any file.

ostree-commit(1)

Given one or more trees, create a new commit using those contents.

ostree-config(1)

Change settings.

ostree-create-usb(1)

Put the given refs on an external drive for P2P distribution.

ostree-diff(1)

Concisely list differences between the given refs.

ostree-find-remotes(1)

Find remotes to serve the given refs.

ostree-fsck(1)

Check a repository for consistency.

ostree-init(1)

Initialize a new repository.

ostree-log(1)

Show revision log.

ostree-ls(1)

List the contents of a given commit.

ostree-prune(1)

Search for unreachable objects.

ostree-pull-local(1)

Copy data from source-repo.

ostree-pull(1)

Download data from remote repo. If you have libsoup.

ostree-refs(1)

List refs.

ostree-remote(1)

Manipulate remote archive configuration.

ostree-reset(1)

Reset a ref to a previous commit.

ostree-rev-parse(1)

Show the SHA256 corresponding to a given rev.

ostree-show(1)

Given an OSTree SHA256 checksum, display its contents.

ostree-static-delta(1)

Manage static delta files.

ostree-summary(1)

Regenerate the repository summary metadata.

ostree-trivial-httpd(1)

Simple webserver.

For specific examples, please see the man page regarding the specific ostree command. For example:

```
man ostree init or man ostree-admin status
```

## GPG VERIFICATION

OSTree supports signing commits with GPG. Operations on the system repository by default use keyring files in `/usr/share/ostree/trusted.gpg.d`. Any public key in a keyring file in that directory will be trusted by the client. No private keys should be present in this directory.

In addition to the system repository, OSTree supports two other paths.

First, there is a `gpgkeypath` option for remotes, which must point to the filename of an ASCII-armored GPG key, or a directory containing ASCII-armored GPG keys to import. Multiple file and directory paths to import from can be specified, as a comma-separated list of paths. This option may be specified by using `--set` in `ostree remote add`.

Second, there is support for a per-remote `remotename.trustedkeys.gpg` file stored in the toplevel of the repository (alongside `objects/` and `such`). This is particularly useful when downloading content that may not be fully trusted (e.g. you want to inspect it but not deploy it as an OS), or use it for containers. This file is written via `ostree remote add --gpg-import`.

## TERMINOLOGY

The following terms are commonly used throughout the man pages. Terms in upper case letters are literals used in command line arguments.

### BRANCH

Branch name. Part of a REF.

### CHECKSUM

A SHA256 hash of a object stored in the OSTree repository. This can be a content, a dirtree, a dirmeta or a commit object. If the SHA256 hash of a commit object is meant, the term COMMIT is used.

### COMMIT

A SHA256 hash of a commit object.

### REF

A reference to a particular commit. References are text files stored in refs/ that name (refer to) a particular commit. A reference can only be the branch name part, in which case a local reference is used (e.g. mybranch/stable). If a remote branch is referred to, the remote name followed by a colon and the branch name needs to be used (e.g. myremote:mybranch/stable).

#### REV, REFSPEC

A specific revision, a commit. This can be anything which can be resolved to a commit, e.g. a REF or a COMMIT.

#### SHA256

A cryptographic hash function used to store objects in the OSTree repository. The hashes have a length of 256 bites and are typically shown and passed to ostree in its 64 ASCII character long hexadecimal representation (e.g. 0fc70ed33cfd7d26fe99ae29afb7682ddd0e2157a4898bd8cfc8a03565b870).

#### SEE ALSO

[ostree.repo\(5\)](#)

[OSTree](#)

[OSTREE\(1\)](#)