Rsync is used utilities for remotely syncing data in Linux. We have explained the simplest way to set up the Rsync for directories.



Linux Commands

# How To Use the Rsync Command in Linux

6 months ago • by Prateek Jangid

Synchronizing files and data among multiple servers is crucial for smooth functioning. Fortunately, many tools are available online for file synchronization, and Rsync is one of them. Rsync is one of the most popular and widely used utilities for remotely syncing data in Linux.

Rsync features efficient file transfer, preservation of file metadata, updating existing files, partial transfers, and more. This makes Rsync an ideal choice for nearly all administrators. So, this guide will be all about using the Rsync command in Linux without hassles.

## How To Use the Rsync Command in Linux

Most Linux distributions contain the Rsync utility, but you have to install it through the following command:

| Operating System | Command |
|---|---|
| Debian/Ubuntu | sudo apt install rsync |
| Fedora | sudo dnf install rsync |
| Arch Linux | sudo pacman -Sy rsync |

After completing the installation, please run the below command to initiate data syncing between the source and the target:

```
rsync -o source target
```

Here, you should replace the source with the directory from where you want to synchronize the data and target with the directory where you want to store that data. For

example, let's sync the Videos and Documents directories by running the following command:

```
rsync  -o Videos Documents
```


If you want to copy-paste data within the same system, use the following command:

```
sudo rsync -avz /source/path /target/path/
```

1. The '-a' or '–archive' keeps the file attributes intact during a data transfer.
2. The '-v' or '–verbose' option is to display what data is being transferred.
3. Although optional, you should use the '-z' or '–compress' option to compress the data during transfer. This aids in speeding up the synchronization process.

Let's take an example and use the above rsync command to synchronize files from the Scripts directory to the Python directory:

```
sudo rsync -avz ~/Scripts ~/Python
```

Moreover, the primary purpose of rsync is to transfer data remotely between two devices or servers connected over a network:

```
rsync -av -e ssh user@remote_host:/source/path/ /target/path
```

Here, the '-e ssh' option commands your system to use the secure shell/SSH for this transaction. Furthermore, if the system encounters any interruption during a remote file transfer, don't worry. You can resume it through the '–partial' option:

```
rsync --partial -av -e ssh user@remote_host:/source/path/ /target/path
```

## Dry Run

Rsync initiates the file transfer immediately after you enter a command. Therefore, to avoid any unintended consequences, you should always perform a dry run first. During a dry run, your system simply demonstrates the actions of your command without an actual data transfer. Hence, here you can add the '–dry-run' option to start a dry run. For instance, to see what's going to happen during a data sync from Python to Scripts directory, use:

```
rsync -avz --dry-run ~/Python ~/Scripts
```

## Make Identical Servers

In case there are some files in the target directory that are not available in the source directory, This results in non-uniformity, and in some cases, it even causes unnecessary