

How to Use Rsync to Transfer Files in Linux

Learn how to use rsync, a powerful command-line utility, to efficiently transfer and synchronize files locally or remotely in Linux with practical examples.

Contents

01	Introduction	3
02	Prerequisites	3
03	Install Rsync	3
04	Install Rsync on Ubuntu and Debian	3
05	Install Rsync on FreeBSD	4
06	Arch Linux	4
07	Install Rsync on Fedora, AlmaLinux, Rocky Linux, and VzLinux	4
08	Install Rsync on CentOS	4
09	Rsync Command Syntax	5
010	Transfer Files and Directories Using Rsync	7
011	Synchronize Data to a Remote Machine Using Rsync	10
012	Conclusion	11

Introduction

Rsync is a file transfer tool for transferring and synchronizing files across systems. It's commonly used for backups, file mirroring, and updating files between local directories or remote servers. Rsync reduces data transfer by copying only the differences between the source and destination files. Its speed, flexibility, and extensive options make it a popular choice for system administrators and developers.

This article explains how to install Rsync to transfer files in Linux. You will use Rsync to perform local and remote file transfers on your workstation.

Prerequisites

Before you begin, you need to:

- Have access to a [Linux-based workstation](#) as a non-root sudo user.
- Enough disk space for large file transfers.

Install Rsync

Rsync is available in the official repositories on most Linux distributions and is often pre-installed. Install Rsync on your system using the commands below, based on your distribution.

Install Rsync on Ubuntu and Debian

- Install Rsync using the APT package manager:

```
CONSOLE
```

```
$ sudo apt install rsync
```

Install Rsync on FreeBSD

- Install Rsync using the PKG package manager:

CONSOLE

```
$ sudo pkg install rsync
```

Arch Linux

- Install Rsync using the Pacman package manager:

CONSOLE

```
$ sudo pacman -S rsync
```

Install Rsync on Fedora, AlmaLinux, Rocky Linux, and VzLinux

- Install Rsync using the DNF package manager:

CONSOLE

```
$ sudo dnf install rsync
```

Install Rsync on CentOS

- Install Rsync using the YUM package manager:

CONSOLE

```
$ sudo yum install rsync
```

Rsync Command Syntax

Rsync uses the following command syntax.

```
rsync [OPTION] SRC DEST
```

Within the command:

- **[OPTION]**: Represents the options or flags passed to the Rsync command.
- **SRC**: The source files and directories you want to synchronize.
 - Use **USER@HOST:SRC** when transferring files remotely over SSH.
- **DEST**: The destination where the files or directories will be synchronized.
 - Use **USER@HOST:DEST** when transferring files remotely over SSH.

Rsync Command Options

The following are the most common Rsync command options:

Option	Long Form	Short Form	Description
Help	<code>--help</code>	<code>-h</code>	Show Rsync command help
Verbosity	<code>--verbose</code>	<code>-v</code>	Increase verbosity
	<code>--quiet</code>	<code>-q</code>	Suppress non-error messages
File Selection	<code>--archive</code>	<code>-a</code>	Archive mode
	<code>--recursive</code>	<code>-r</code>	Recurse into directories
	<code>--relative</code>	<code>-R</code>	Use relative path names
Updates	<code>--update</code>	<code>-u</code>	Skip files that are newer on the receiver
	<code>--inplace</code>		Update destination files in-place
Backups	<code>--backup</code>	<code>-b</code>	Make backups (see <code>--suffix</code> & <code>--backup-dir</code>)
	<code>--backup-dir=DIR</code>		Make backups into hierarchy based in DIR
Symlinks	<code>--links</code>	<code>-l</code>	Copy symlinks as symlinks

Option	Long Form	Short Form	Description
	<code>--copy-links</code>	<code>-L</code>	Transform symlink into referent file/dir
Permissions	<code>--perms</code>	<code>-p</code>	Preserve permissions
Timestamps	<code>--times</code>	<code>-t</code>	Preserve modification times
Deletion	<code>--delete</code>		Delete extraneous files from dest dirs
	<code>--delete-excluded</code>		Also delete excluded files from dest dirs
Simulation	<code>--dry-run</code>	<code>-n</code>	Perform a trial run with no changes made
Remote Shell	<code>--rsh=COMMAND</code>	<code>-e</code>	Specify the remote shell to use
Size Control	<code>--max-size=SIZE</code>		Don't transfer any file larger than <code>SIZE</code>
	<code>--min-size=SIZE</code>		Don't transfer any file smaller than <code>SIZE</code>
Partial Transfer	<code>--partial</code>		Keep incomplete transferred files
File Filtering	<code>--exclude=PATTERN</code>		Exclude files matching PATTERN
	<code>--include=PATTERN</code>		Don't exclude files matching PATTERN
Output Format	<code>--progress</code>		Show progress during transfer
	<code>-P</code>		Same as <code>--partial --progress</code>
	<code>--human-readable</code>	<code>-h</code>	Output numbers in a human-readable format
Compression	<code>--compress</code>	<code>-z</code>	Compress file data during the transfer
IP Version	<code>--ipv4</code>	<code>-4</code>	Prefer IPv4
	<code>--ipv6</code>	<code>-6</code>	Prefer IPv6

Transfer Files and Directories Using Rsync

This section provides step-by-step instructions on how to use Rsync to transfer files and directories. Follow the steps below to set up a test environment and execute Rsync commands.

1. Create a directory named `rsync-test` to work with.

CONSOLE

```
$ mkdir rsync-test
```

2. Navigate into the newly created directory.

CONSOLE

```
$ cd rsync-test
```

3. Create three sample files.

CONSOLE

```
$ touch file1 file2 file3
```

Transfer Local Files

The following steps show how to transfer files between directories using Rsync.

1. Copy `file1`, `file2`, and `file3` to a new directory named `dest`. In Rsync, the last argument always represents the destination. Ensure the destination path is correctly specified to prevent accidental overwrites.

CONSOLE

```
$ rsync file1 file2 file3 dest
```

- `file1 file2 file3`: Source files to copy.
- `dest`: Destination directory where the files will be copied.

2. Verify that the files were copied successfully using `ls`.

CONSOLE

```
$ ls dest/
```

Your output should be similar to the one below.

```
./dest/file1  
./dest/file2  
./dest/file3
```

Transfer Local Directories

Rsync does not copy files inside a directory by default. Use the `-a` flag to enable recursive copying, which enables archive mode, preserving timestamps, permissions, group, owner, and symbolic links. Follow the steps below to copy local directories recursively using the Rsync command.

1. Copy the `dest` directory you created earlier.

CONSOLE

```
$ rsync -a dest/ dest-copy
```

- `-a`: Enables archive mode to preserve file attributes while copying recursively.
- `dest/`: The source directory to copy (trailing / means copy contents, not the directory itself. If you omit the trailing /, Rsync copies the entire directory instead of just its contents.)

- `dest-copy`: The destination directory.

2. Verify the copy operation using the `ls` command.

CONSOLE

```
$ ls dest-copy/
```

Your output should be similar to the one below. A `dest-copy` directory is created with the same content as the `dest` directory.

```
./dest-copy/file1  
./dest-copy/file2  
./dest-copy/file3
```

Exclude Files or Directories Using Rsync

To exclude specific files or directories from being copied use the Rsync `--exclude` flag.

1. Copy the contents of the current directory, excluding `file2` and all directories that start with `dest`.

CONSOLE

```
$ rsync -a --exclude 'file2' --exclude 'dest*' . output
```

- `-a`: Enable archive mode.
- `--exclude 'file2'`: Exclude files and directories named `file2`.
- `--exclude 'dest*'`: Exclude files and directories that starts with `dest`.
- `.`: Specify the current directory as the source.
- `output`: The destination directory.

2. Verify the copy operation using the `ls` command.

CONSOLE

```
$ ls output/
```

Your output should be similar to the one below. Notice that the `output` directory contains only `file1` and `file3`.

```
./output/file1
./output/file3
```

Synchronize Data to a Remote Machine Using Rsync

Follow the steps below to synchronize your files and directories to a remote machine using SSH.

1. Synchronize files and directories to a remote machine.

CONSOLE

```
$ rsync -hazvP --delete --stats .
user@192.0.2.1:synced_from_local
```

- `-h`: Output numbers in human-readable format.
- `-a`: Enables archive mode for preserving metadata.
- `-z`: Compresses data during transfer.
- `-v`: Enables verbose output.
- `-P`: Displays progress and resumes partial transfers.
- `--delete`: Removes files from the destination that no longer exist in the source.
- `--stats`: Provides a detailed summary of the transfer.
- `.`: Specify the current directory as the source.
- `user@192.0.2.1:synced_from_local`: Specify the remote machine as the destination. This consist of three parts:
 - `user`: The username to access the remote machine.
 - `192.0.2.1`: The IP address of the remote machine.
 - `synced_from_local`: The destination path.

2. Log in to the remote machine using SSH.

CONSOLE

```
$ ssh user@192.0.2.1
```

3. Verify that the files have been synced.

CONSOLE

```
$ ls synced_from_local/
```

Output:

```
synced_from_local/dest-copy/file1
synced_from_local/dest-copy/file2
synced_from_local/dest-copy/file3
synced_from_local/dest/file1
synced_from_local/dest/file2
synced_from_local/dest/file3
synced_from_local/file1
synced_from_local/file2
synced_from_local/file3
synced_from_local/output/file1
synced_from_local/output/file3
```

Warning on Using the `--delete` Rsync Option

Be cautious when using `--delete`, as it permanently removes files from the destination if they don't exist in the source. Always perform a dry run first:

CONSOLE

```
$ rsync -a --delete --dry-run SRC DEST
```

Conclusion

You have installed Rsync used it to synchronize data between local and remote machines. You also used common Rsync options like filtering files with `--exclude`

and copying directories recursively. Always double-check your Rsync commands, especially when using options like `--delete`. Run the `man rsync` command for more information and command options.



VULTR

