

What is Rsync?

Rsync, short for "remote sync," is a file synchronization tool for both remote and local files. It employs an algorithm to minimize the amount of data transferred by only moving the parts of files that have changed.

Syntax that is used for Rsync

The syntax is quite similar to other tools like SSH, SCP, and CP.

First, change the directory:

```
$ cd ~
```

Next, create a directory:

```
$ mkdir dir1
```

Now, create another test directory:

```
$ rsync -anv dir1 dir2
```

Let's add some test files:

```
$ rsync -anv dir1 dir2
```

You will now have a directory called `dir1` containing 100 empty files. Confirm this by listing the files:

```
$ ls dir1
```

You also have an empty directory called `dir2`. To sync the contents of `dir1` to `dir2` on the same system, run Rsync with the `-r` flag, which stands for "recursive" and is necessary for syncing directories:

```
$ rsync -r dir1/ dir2
```

Alternatively, you can use the `-a` flag, which is a combination flag standing for "archive." This flag syncs recursively and preserves symbolic links, special and device files, modification times, groups, owners, and permissions. It is more commonly used than `-r` and is the recommended flag. Run the same command as before, but this time use the `-a` flag:

(Note: It is important to include the trailing slash (/) at the end of the first directory, as it signifies the contents of `dir1`. Without the slash, `dir1` itself, including the directory, would be placed in `dir2`, resulting in a hierarchy like `~/dir2/dir1/[files]`.)

Another tip is to double-check your arguments before executing an Rsync command. Rsync allows you to do this by using the `-n` or `--dry-run` option. The `-v` flag, which means "verbose," is also necessary to get detailed output. You can combine the `-a`, `-n`, and `-v` flags in the following command:

```
$ rsync -anv dir1/ dir2
```

```
$ rsync -anv dir1/ dir2
sending incremental file list
./
file1
file10
file100
file11
file12
file13
file14
file15
file16
file17
file18
file19
file2
file20
file21
file22
file23
file24
file25
file26
file27
file28
file29
file3
file30
file31
file32
file33
file34
file35
file36
file37
```

```
$ rsync -anv dir1 dir2
```

```
$ rsync -anv dir1 dir2
sending incremental file list
dir1/
dir1/file1
dir1/file10
dir1/file100
dir1/file11
dir1/file12
dir1/file13
dir1/file14
dir1/file15
dir1/file16
dir1/file17
dir1/file18
dir1/file19
dir1/file2
dir1/file20
dir1/file21
dir1/file22
dir1/file23
dir1/file24
dir1/file25
dir1/file26
dir1/file27
dir1/file28
dir1/file29
dir1/file3
dir1/file30
dir1/file31
dir1/file32
```

This output now indicates that the directory itself was transferred, rather than just the files within it.

Using Rsync to Sync with a Remote System

To synchronize with a remote system using Rsync, you must have SSH access set up between your local and remote machines, and Rsync installed on both.

Once you verify the SSH connection, you can sync the `dir1` folder from the earlier section to a remote machine using the following syntax. Remember to omit the trailing slash in this case to transfer the actual directory:

```
rsync -a ~/dir1 username@remote_host:destination_directory
```

This operation is referred to as a "push" operation because it "pushes" a directory from your local system to a remote one. Conversely, the "pull" operation is used to sync a remote directory to your local system. If the `dir1` directory were located on the remote system instead, the syntax would be:

```
rsync -a username@remote_host:/home/username/dir1 directory_to_sync_locally
```

As with the `cp` command and similar tools, the source is always the first argument, while the destination is always the second.

Reference

<https://www.digitalocean.com/community/tutorials/how-to-use-rsync-to-sync-local-and-remote-directories#understanding-rsync-syntax>