DESIGN & DEVELOPMENT ⌄   HOW TO ⌄   GEEK OUT ⌄   BACK TO PAIR

LET'S STAY IN TOUCH!

email address

SUBSCRIBE

Questions?
Contact our sales team
1 (877) 724-7638

HOW TO

Rsync: What is it and How do I Use it?

March 13, 2018

WHAT IS RSYNC?

Rsync is a Linux-based tool that can be used to sync files between remote and local servers. Rsync has many options that can help you define the connections you make. From deciding the type of shell that should be used to files that should be excluded in a transfer, rsync gives you the power to shape the transfer specifications.

It should be noted, however, that rsync makes it *very easy* to accidentally overwrite your files. So we recommend double-checking everything before you sync or using the `-dry-run` option.

Rsync is great for complex file syncs and for transferring a large number of files. In fact, it's better to use rsync for moving large batches of files than scp. When used with cron, rsync is also able to take automatic backups. While it may look and sound difficult, rsync can be very useful and accomplish things that less intimidating interfaces may not be able to.

TYPICAL FORMAT

The typical rsync format isn't very intimidating when you break it down.

```
rsync [option] [origin] [destination]
```

Each rsync command should begin with rsync. It is then followed by an option. You can see a list of options on the Options Summary of the Samba Rsync documentation page. The origin and destination format may vary based on whether they are a remote or local machine.

OPTIONS

Here are a list of basic rsync options to get your started.

-a	copy files recursively and preserve ownership of files when files are copied
–dry-run	runs a trial run of the command; does not make any actual changes
–delete	delete any extraneous files from the destination directory

Looking for something ? Search away

SEARCH

OUR CUSTOMERS LOVE US!

"I have been a pair customer since 2000. They have always been wonderful to me."

-Nancy R.

Customer since 2000

-e	tell rsync what shell to use
–exclude="*.*fileType*"	exclude all of a specific file type. Replace *fileType* with the actual file type that should be excluded
-h	produce easily readable output, not just numbers
–h	initiate rsync help
–progress	shows the transfer progress as the command is being run
-q	runs the quiet version of the command; all of the processes that are run will be run in the background and will not be shown
-v	runs the verbose version of the command; all of the processes that are run will be written out for the user to read
-z	compress the data synced

RSYNC NUANCES

Rsync isn't without its nuances. Here are some things to remember when using rsync.

COMBINING OPTIONS IN COMMANDS

If more than one option with a dash (-) is used, they may be consolidated into one bundle of options that has only one dash. For example, `-avh` would invoke options `-a`, `-v`, and `-h`.

BE AWARE OF THE TRAILING SLASH

```
rsync -avhz /home/user/documents/ user@host:/usr/home/public_html/domain.com/
```

It is important to pay attention to the trailing slash when writing an rsync command. The trailing slash is located at the end of a file path and indicates that the directory itself should not be copied over, only the contents. If you do not include the trailing slash, it will copy the directory as well, with contents inside.

USE CASES

SEND FILES FROM REMOTE ORIGIN TO LOCAL DESTINATION

In order to send files from a remote origin to a local destination, follow this format:

```
rsync [option] user@host:[file/directory origin] [destination]
```

Here's an example of what this command might look like:

```
rsync -avh username@remoteHosting:/usr/home/user/public_html/domain.com /home
```

As mentioned above, the options are consolidated into one bundle, instead of being written out as separate options with individual dashes (-).

SEND FILES FROM LOCAL ORIGIN TO REMOTE DESTINATION

In order to send files from a local origin to a remote destination, follow this format:

```
rsync [option] [origin] user@host:[file/directory destination]
```

Here's an example of what this command might look like:

```
rsync -avhz /home/user/documents username@remoteHosting:/usr/home/public_html
```

This process is very similar to the Send Files from Remote Origin to Local Destination. In fact, it really only switched the format of the origin and destination.

DELETING FILES IN THE DESTINATION THAT DON'T MATCH THE ORIGIN

Using the –delete option will delete any files in the destination directory that don't match the files being copied from the origin. It will look something like this:

```
rsync -avhz --delete /home/user/documents username@remoteHosting:/usr/home/pu
```

Regardless of whether the destination is remote or local, the files that don't have a match in the origin directory will be deleted.

SHOW RSYNC PROGRESS

If you're making a large file transfer or running a command that takes a while to process, you may want to use the –progress option. This option will show the transfer progress of your command. It will look like this in a command:

```
rsync -avhz --progress /home/user/documents username@remoteHosting:/usr/home/
```

EXCLUDE SPECIFIC FILE TYPES

When you want to exclude certain file types from the transfer, you can say so right in the code.

```
rsync -avhz --exclude="*.png" /home/user/documents username@remoteHosting:/us
```

The example above excludes all file types of the .png extension. You can change the file extension to any file type you don't want to be included in the transfer.

ONLY THE BEGINNING OF RSYNC

If you are looking for a tool to help you make complex file transfers, rsync may be what you are looking for. While this blog only covers the basics, you can find many more rsync commands on the developer's documentation site. Armed with the basics you found here, you should be able to start tackling more complicated rsync uses, like setting up unattended backups by combining rsync and cron, which will be covered in a future blog post. Be sure to stay tuned for future blog posts on rsync and other technical topics.

Don't want to miss anything? Subscribe to our blog to get automatic updates when a new blog comes out.

command line   file transfer   rsync

YOU MAY ALSO LIKE

What's a **Traceroute** and How Do I Use it?

What's a Traceroute and How Do I Use One?

What is Curl?

What is curl?