

Rsync as a Backup Solution

Rsync is a command-line utility that synchronizes sets of directories and files between file systems. It was written primarily for remote file copying, but it works really well for local file copies too.

Here's an example, showing how I use it for backups:

```
rsync -lrt --delete /home/jdoe/Documents /media/HD2/fullsync
```

When you issue this command, you end up with a synchronized copy of the Documents folder on HD2 (HD2 is the target, /home/jdoe/Documents is the source/working copy). The target path ends up being /media/HD2/fullsync/Documents.

The command line arguments are as follows:

- -l copy symlinks as symlinks
- -r recurse into directories
- -t preserve modification times
- --delete delete extraneous files from destination directories (this ensures that when you delete a file in your source directory, it doesn't hang around in your target directory)

There are plenty of additional command line arguments, but these are just the ones I use for my needs.

I've also created a Python script to simplify the backup process:

[fullsync.py](#)

```
#!/usr/bin/python

import glob
import os
import os.path
import shlex
import statvfs
import subprocess
import sys

def ExecBackup(source, target, skipit):
    try:
        if skipit == True:
            print "Skipped " + source
        else:
            if not os.path.exists(target):
                os.makedirs(target)
            if os.path.exists(source):
                procName = 'rsync -lrt --delete "' + source + '" "' +
target + '"'
                myarg = shlex.split (procName)
                myarg = myarg[:-1] + glob.glob(myarg[-1])
```

```
        p = subprocess.Popen(myarg)
        p.communicate()
        print "Synced " + source
    else:
        print "Invalid source path: " + source
except Exception as ex:
    print 'Call to ' + procName + ' failed: ' + str(ex)

### Main() starts here ###

# NOTE: The newTarget setting and ExecBackup calls are examples.
# Modify to suit your needs.

# Target folder for synchronized copies.
newTarget = '/media/HD2/fullsync'

# Each call to ExecBackup synchronizes a single directory and all of
# its subdirectories
ExecBackup('/home/jdoe/Documents', newTarget, False)
ExecBackup('/home/jdoe/Music', newTarget, False)

# Show free space remaining on target drive
f = os.statvfs(newTarget)
totalSize = (f[statvfs.F_BSIZE] * f[statvfs.F_BFREE]) / 1024/1024/1024
print '\nSpace remaining: ' + str(totalSize) + 'G'

print '\nFinished successfully.'
```

I use rsync in Linux, but there are various implementations available for Windows. You can find a list in the rsync Wikipedia entry [here](#). If you want to use rsync in Windows, I personally recommend installing [Cygwin](#). It will give you rsync, and also a lot of other really useful utilities.

From:

<https://kbase.devtoprd.com/> - Knowledge Base

Permanent link:

https://kbase.devtoprd.com/doku.php?id=rsync_backup_solution

Last update: **2025/06/08 07:21**

