

Backing Up with rsync

Dave Emmons

October 2006

Backing Up with rsync

- Why
- How

Why?

- Backing up is a good thing to do
- rsync is free
- rsync does a good (not perfect) job

How?

- Open `/Applications/Utilities/Terminal.app`
- You should see something like:

```
Last login: Mon Oct 23 14:04:21 on ttty1
Welcome to Darwin!
68-117-126-130:~ dave$
```

How?

- Type in `man rsync` and press Return. Get familiar with the dozens of options available. What I'm showing is a very simplistic setup and may not be right for you.

- Now type in something like this example, updating the paths:

```
rsync --verbose --progress -a -e /Users/UN/* "/Volumes/Disk/backups/TARGET/"
```

How?

• What's all this mean?

- `rsync`: `rsync` is a free program being use to perform the backup
- `--verbose`: this option gives the user feedback and is optional
- `--progress`: This option tells `rsync` to print information showing the progress of the transfer. This gives a bored user something to watch. Implies `--verbose` without incrementing verbosity. When the file is transferring, the data looks like this: `782448 63% 110.64kB/s 0:00:04` This tells you the current file size, the percentage of the transfer that is complete, the current calculated file-completion rate (including both data over the wire and data being matched locally), and the estimated time remaining in this transfer. After the file is complete, it the data looks like this: `1238099 100% 146.38kB/s 0:00:08 (5, 57.1% of 396)` This tells you the final file size, that it's 100% complete, the final transfer rate for the file, the amount of elapsed time it took to transfer the file, and the addition of a total-transfer summary in parentheses. These additional numbers tell you how many files have been updated, and what percent of the total number of files has been scanned.
- `-a`: tells `rsync` to use archive emode
- `/Users/UN/* "/Volumes/BackupDisk/backups/TARGET/"` this portion tells `rsync` what to backup and where to put it. The target disk can can be a local disk as shown in this example or it can be a network volume. When connecting to a network volume, I prefer to connect via `ssh` so the traffic is encrypted. This is handy for backing up over the internet.

How?

- To backup to a network host, make sure remote login is enabled in the Sharing prefpane and you know the login info.
- Use this command `rsync --verbose --progress -a -e ssh /Users/username/* "itsupport@192.168.0.79:/Volumes/LaCie\ Disk/backups/target_dir/"`

How?

- The main difference is the addition of the ssh command and the target login and path. "`itsupport@192.168.0.79:/Volumes/LaCie\ Disk/backups/target_dir/`" The first part is the username@hostname or address followed by a colon. Then the remote path is defined. After pressing Return, you will be prompted for the remote user's login password.

How?

- You can automate the backup process by adding a line to the system crontab. To do this, type `crontab -e` on the command line. Add a line like

```
30 10 * * * root rsync -a -e /Users/username/* /Volumes/  
Backup\ Drive\ A/backups/targer_dir/ "
```

- and make sure you save the file before exiting.
- While it is possible to have this work with a remote network volume as well, it is beyond the scope of this presentation. for the curious, see this document or details on using ssh-keygen to help in this process

- <http://www.bombich.com/mactips/rsync.html>

Downside?

- Yes, there is a downside. The version of rsync Apple ships in OS X does NOT PRESERVE resource forks. This is critical if you're still using a lot of Classic applications or others that use the resource fork like Quark Xpress or Adobe Photoshop. While a version of rsyncX that preserves resource forks does exist, I have yet to be sold on the robustness of the version. your mileage may vary. You can read more about rsyncX, and download the latest build at:
<http://archive.macosxlabs.org/rsyncx/rsyncx.html>