rsync

Introduction

Besides for standard sync and backup, use rsync to,

1. Copy over network and measure speed.
2. Copy and resume large directory backups.

Rsync is much more reliable and allows for continuation in case of interruption.

The **biggest mistake** people don't do is to **not test their backup by restoring**. You'll be surprised to find later that despite no errors, upon restore you've lost some data. This is because failed copies due to special characters is not alway. Rsync has a nice dry-run function so you can test without having to do a full restore.

Backup and Sync

Most used to perform difference only mirror backup of source to destination and shows,

- Progress of copy
- Speed of copy
- Summary of bytes sent & received, total size and average transfer speed

On a desktop machines that might go to sleep using with a keep alive command,

```
caffeinate rsync [...] # Mac OS X
? # Linux
```

Backup Interactively

Here is the rsync command,

```
rsync --archive --delete --sparse --verbose --itemize-changes
--human-readable --progress /home/tempadmin/source
/home/tempadmin/destination
```

--dry-run = Use this first to ensure to simulate a run. Especially if you use --delete.

--archive = which includes,

- --recursive = subdirectories
- --links = copy symbolic links as symbolic links
- --perms = preserve permissions
- --times = preserve times
- --group = preserve group
- --owner = preserve owner
- --devices = preserve device files (super-user only)
- --specials = preserve special files

--delete = delete any files not in the source

--sparse =

--verbose =

--itemize-changes =

--human-readable =

--progress =

source =

destination = destination may be a directory but the most useful is actually to another system running the rsync service

Be very careful not to include a trailing slash in **source** and **destination**(maybe slash ok here but need to test) or you will end up deleting **everything** in your target.

Backup to rsync Service

Instead of a directory, the destination may be another system running the rsync service. This method is extremely fast compare to the everyday use AFP or SMB share protocols. However, I believe there are some limitations when it comes to special characters at least on Mac OS X. I'm still figuring this out.

```
rsync --archive --delete --sparse --compress --verbose --itemize-changes
--human-readable --progress /home/tempadmin/source
rsync://rsyncUser@destSystem/destination
```

--destination =

--compress =

A point of interest is that I find I often need to run rsync more than once, as it often finds discrepancies even right after the first transfer. Keep on running rsync until you stop seeing "files to consider".

Backup via Script

In order to run as a script add the following,

--password-file =

--log-file=

Straight Copy

Simple straight copy and can be used to resume (still need to modify for resume),

```
 rsync --archive --sparse --compress --verbose --itemize-changes
--human-readable --progress /home/tempadmin/tmp/source
/home/tempadmin/tmp/destination
```

Details

--archive

Same as -rlptgoD (no -H)

This is equivalent to recursive, links, perms, times, group, owner, specials. It is a quick way of saying you want recursion and want to preserve almost everything (with hard-links being a notable omission). The only exception to the above equivalence is when --files-from is specified, in which case -r is not implied.

> **--owner**
>
> This option causes rsync to set the owner of the destination file to be the same as the source file, but only if the receiving rsync is being run as the super-user (see also the --super option to force rsync to attempt super-user activities). Without this option, the owner is set to the invoking user on the receiving side.
>
> The preservation of ownership will associate matching names by default, but may fall back to using the ID number in some circumstances

(see also the --numeric-ids option for a full discussion).

--group

This option causes rsync to set the group of the destination file to be the same as the source file. If the receiving program is not running as the super-user (or if --no-super was specified), only groups that the invoking user on the receiving side is a member of will be preserved. Without this option, the group is set to the default group of the invoking user on the receiving side.

The preservation of group information will associate matching names by default, but may fall back to using the ID number in some circumstances (see also the --numeric-ids option for a full discussion).

> If you plan to rsync to another system, you should align your owner and group names and uid's to match up. Otherwise, if names do not match it uses uid numbers. Still to test this and write down examples and understand repercussions with possible work arounds (ie, using command to save all attributes as text file to apply on restore).

--delete

Delete extraneous files from destination directories.

--sparse

Try to handle sparse files efficiently so they take up less space on the destination. Conflicts with --inplace because it's not possible to overwrite data in a sparse fashion.

> Don't use this option when the destination is a Solaris "tmpfs" filesystem.

--compress

Compress files during transfer.

> **Does not Compress**
> The default list of file extensions that will **not** be compressed is: `gz zip z rpm deb iso bz2 tbz tgz 7z mp3 mp4 mov avi ogg jpg jpeg`

> For images, media or any other already compressed files do **not** compress as you just slow things down.

--progress

Show progress.

> Don't use with cron.

Special Use

--whole-file

Use this for first time sync if you have lots of files.

With this option the incremental rsync algorithm is **not** used and the whole file is sent as-is instead. The transfer may be faster if this option is used when the bandwidth between the source and destination machines is higher than the bandwidth to disk (especially when the "disk" is actually a networked filesystem). This is the default when both the source and destination are specified as local paths.

--inplace

This option is useful for transfer of large files with block-based changes or appended data, and also on systems that are disk bound, not network bound.

Mac OS X to Backup iPhoto

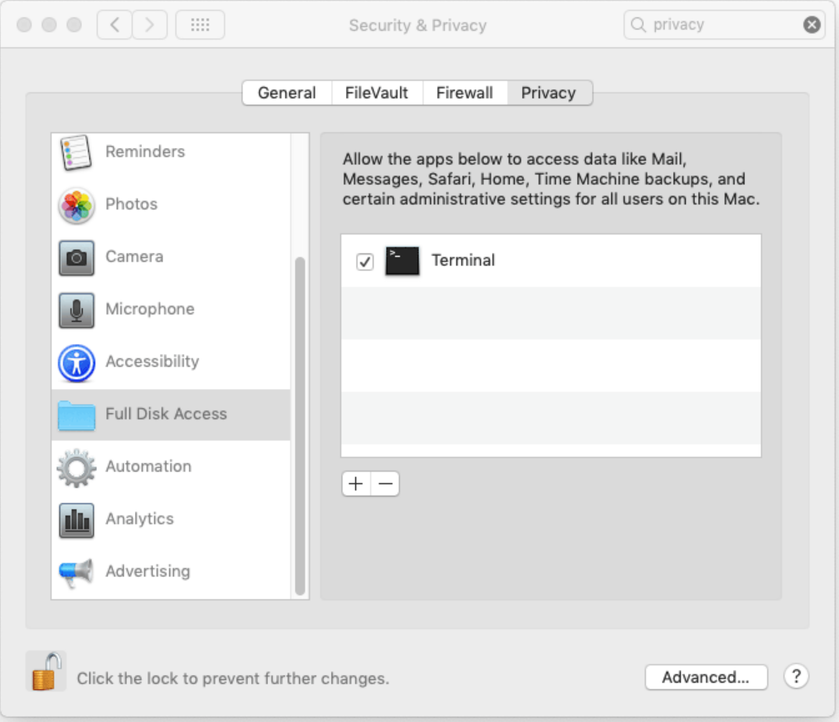
Terminal Restriction

Before using rsync you need to lift FDA (full disk access) restrictions in **System Preferences > Security & Privacy > Privacy > Full Disk Access** and add **Terminal** otherwise you may see the error messages when trying to rsync.

For me, I ran into this challenge with my Photos Library,

```
building file list
rsync: opendir "/Users/tin.pham/Pictures/Photos Library.photoslibrary"
failed: Operation not permitted (1)
1 file to consider
IO error encountered -- skipping file deletion
sent 102 bytes  received 16 bytes  21.45 bytes/sec
total size is 0  speedup is 0.00
rsync error: some files could not be transferred (code 23) at
/BuildRoot/Library/Caches/com.apple.xbs/Sources/rsync/rsync-52.200.2/rsync
/main.c(996) [sender=2.6.9]
```

After making the System Preferences change, it will ask you to restart your **Terminal** app for the changes to take effect. Here is how your Privacy will look after Terminal is added,



Allow the apps below to access data like Mail, Messages, Safari, Home, Time Machine backups, and certain administrative settings for all users on this Mac.

☑ Terminal

Permission Problems when Syncing with Your Synology Nas Drive

If you see this error message when syncing with your NAS drive,

```
rsync: failed to set permissions on "/photos/." (in rsync): Operation not
permitted (1)
```

...

NFS or FUSE File Restrictions

If copying to or copying from file systems that use mapped drives such as NFS or FUSE, you may run into some trouble seeing set times.

```
rsync: failed to set times on "." (in rsync): Operation not permitted (1)
```

You can suppress this with **--omit-dir-times**

Extended Attributes

--extended-attributes

Copy extended attributes and resource forks.

Exclude unnecessary Mac OS X System Files

List from Alan Smith,

```
--exclude='$RECYCLE.BIN' --exclude='$Recycle.Bin' --exclude='.AppleDB'
--exclude='.AppleDesktop' --exclude='.AppleDouble'
--exclude='.com.apple.timemachine.supported' --exclude='.dbfseventsd'
--exclude='.DocumentRevisions-V100*' --exclude='.DS_Store'
--exclude='.fseventsd' --exclude='.PKInstallSandboxManager'
--exclude='.Spotlight*' --exclude='.SymAV*' --exclude='.symSchedScanLockxz'
--exclude='.TemporaryItems' --exclude='.Trash*' --exclude='.vol'
--exclude='.VolumeIcon.icns' --exclude='Desktop DB' --exclude='Desktop DF'
--exclude='hiberfil.sys' --exclude='lost+found' --exclude='Network Trash
Folder' --exclude='pagefile.sys' --exclude='Recycled' --exclude='RECYCLER'
--exclude='System Volume Information' --exclude='Temporary Items'
--exclude='Thumbs.db'
```

Over SSH Protocol

rsync ....

```
# Rsync over the Internet
rsync --archive --verbose --compress --delete --progress -e "ssh -c arcfour
-o Compression=no -x" /source/folder
remotebackup@earth.com:/home/user:destination-folder

# Rsync over LAN
# Same but disable all compression.
```

caffeinate -s = prevents OS X from sleeping until command is done

--archive or -a = Archive mode. Performs recursion and preserves almost all attributes of the source files (with -H being a notable omission). Note that it does not preserve hard links, because finding multiply-linked files is expensive. You must separately specify -H.

-v = Verbose. Using -vv will provide additional detail. Additionally more v's may be added.

-e ssh = Specify remote shell to be ssh.

-c arcfour = uses the weakest but fastest **encryption** that ssh supports.

-o Compression=no = Disable ssh compression as we will be using rsync's own which is more efficient.

--compress or -z = Enable rsync's compression.

-x = turns off ssh's X tunneling feature (if you have it on by default).

--dry-run or -n = Very important to use first time or to test --delete. Performs trial run without making changes. Use in combination with -v and --itemize-changes. -vv will provide even more details.

--delete or -d = Delete on target to match source.

--itemize-changes or -i = List of changes for each file including attribute changes.

--human-readable or -h = Makes numbers in the log and stdout more readable when it comes to large units.

--progress = Shows progress of transfer. Make sure not to use when using cron.

Command Reference

Remote file copy - Synchronize file trees across local disks, directories or across a network.

Syntax

```
 Local file to Local file:
      rsync [option]... Source [Source]... Dest

 Local to Remote:
      rsync [option]... Source [Source]... [user@]host:Dest   #

      rsync [option...] [user@]host::Source... [Dest]

rsync [option...] rsync://[user@]host[:PORT]/Source... [Dest]

 Remote to Local:
      rsync [option]... [user@]host:Source... [Dest]          #

      rsync [option]... [user@]host::Dest

      rsync [option]... rsync://[user@]host[:PORT]/Dest

 # = via remote shell rather than the rsync daemon
```

OPTIONS SUMMARY

Here is a short summary of the options available in rsync.
Please refer to the FULL List of OPTIONS for a complete description.

```
What to copy:
 -r, --recursive             recurse into directories
 -R, --relative              use relative path names
     --exclude=PATTERN       Exclude files matching PATTERN
     --exclude-from=FILE     Read exclude patterns from FILE
 -I, --ignore-times          Don't exclude files that match length and time
     --size-only             only use file size when determining if a file should be transferred
     --modify-window=NUM     Timestamp window (seconds) for file match (default=0)
     --include=PATTERN       Don't exclude files matching PATTERN
     --include-from=FILE     Read include patterns from FILE

How to copy it:  -n, --dry-run               Perform a trial run with no changes made
 -l, --links                 Copy symlinks as symlinks
 -L, --copy-links            Transform symlink into referent file/dir
     --copy-unsafe-links     Only "unsafe" symlinks are transformed
     --safe-links            Ignore links outside the destination tree
 -H, --hard-links            Preserve hard links
 -D, --devices               Preserve devices (super-user only)
 -g, --group                 Preserve group
 -o, --owner                 Preserve owner (super-user only)
 -p, --perms                 Preserve permissions
 -t, --times                 Preserve times
 -S, --sparse                Handle sparse files efficiently
 -x, --one-file-system       Don't cross filesystem boundaries
 -B, --block-size=SIZE       Force a fixed checksum block-size (default 700)
 -e, --rsh=COMMAND           Specify rsh replacement
     --rsync-path=PATH       Specify path to rsync on the remote machine
     --numeric-ids           Don't map uid/gid values by user/group name
     --timeout=TIME          Set IO timeout in seconds
 -W, --whole-file            Copy whole files, no incremental checks

Destination options:  -a, --archive               Archive mode
 -b, --backup                Make backups (see --suffix & --backup-dir)
     --backup-dir=DIR        Make backups into this directory
     --suffix=SUFFIX         Override backup suffix
 -z, --compress              Compress file data during the transfer
 -c, --checksum              Skip based on checksum, not mod-time & size
 -C, --cvs-exclude           Auto ignore files in the same way CVS does
     --existing              Only update files that already exist
     --delete                Delete files that don't exist on the sending side
     --delete-excluded       also delete excluded files on the receiving side
     --delete-after          Receiver deletes after transfer, not during
     --force                 Force deletion of directories even if not empty
     --ignore-errors         Delete even if there are IO errors
     --max-delete=NUM        Don't delete more than NUM files
     --log-format=FORMAT     Log file transfers using specified format
     --partial               Keep partially transferred files
     --progress              Show progress during transfer
 -P                          equivalent to --partial --progress
     --stats                 Give some file transfer stats
 -T  --temp-dir=DIR          Create temporary files in directory DIR
     --compare-dest=DIR      also compare destination files relative to DIR
 -u, --update                update only (don't overwrite newer files)

Misc Others:      --address=ADDRESS       bind to the specified address
     --blocking-io           Use blocking IO for the remote shell
     --bwlimit=KBPS          Limit I/O bandwidth, KBytes per second
     --config=FILE           Specify alternate rsyncd.conf file
     --daemon                Run as a rsync daemon
     --no-detach             Do not detach from the parent
     --password-file=FILE    Get password from FILE
     --port=PORT             Specify alternate rsyncd port number
 -f, --read-batch=FILE       Read batch file
 -F, --write-batch           Write batch file
     --version               Print version number
 -v, --verbose               Increase verbosity
 -q, --quiet                 Decrease verbosity
 -4, --ipv4                  Prefer IPv4
 -6, --ipv6                  Prefer IPv6
 -h, --help                  show this help screen
```

TBC - Roderick

References

Good practical overview - http://jimmyg.org/blog/2007/rsync-basics.html

Review of the most common flags - http://www.evbackup.com/support-commonly-used-rsync-arguments/

GUI to learn and execute rsync - http://www.linuxjournal.com/content/rsync-its-grrrraphical

For MAC OS X consider - http://osxdaily.com/2009/02/19/command-line-back-ups-in-os-x/

Prevent MAC OX from sleeping - http://www.pcadvisor.co.uk/news/software/3382592/top-20-os-x-command-line-secrets-for-power-users/

Solution to OpenDir Error for Photos on Mac - https://www.reddit.com/r/MacOS/comments/bvo5wt/rsync_error_copying_libraryphotoslibrary/

Solution to FUSE or NFS on MAC - https://stackoverflow.com/questions/667992/rsync-error-failed-to-set-times-on-foo-bar-operation-not-permitted/668049#668049