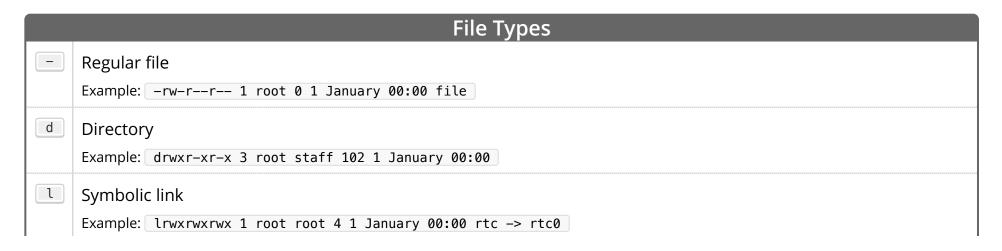
# **Unix File Permissions**

Enjoy this cheat sheet at its fullest within <u>Dash</u>, the macOS documentation browser.

Permissions		
Octal	Symbol	Permission
0		No permissions
1	x	Execute
2	_w_	Write
3	-wx	Write and execute
4		Read
5	r-x	Read and execute
6	rw-	Read and write
7	rwx	Read, write and execute



Block special device
Example: brw-rw---- 1 root disk 1 0 1 January 00:00 ram0

C Character device
Example: crw-rw-rw- 1 root root 1 3 1 January 00:00 null

S Unix socket
Example: srw-rw-rw- 1 root root 0 1 January 00:00 acpid.socket

P Named pipe
Example: prw-r---- 1 root root 0 1 January 00:00 pipe

## **Special Mode Bits**

#### setuid (Set User ID)

When the setuid permission is set on an executable file, a process that runs this file is granted access based on the owner of the file (usually root)

This special permission allows a user to access files and directories that are normally only available to the owner.

Example: The setuid permission on the passwd command makes it possible for a user to change passwords, assuming the permissions of the root ID:

-r-sr-sr-x 3 root sys 104580 Sep 16 12:02 /usr/bin/passwd

#### setgid (Set Group ID)

The set-group identification (setgid) permission is similar to setuid, except that the process's effective group ID (GID) is changed to the group owner of the file.

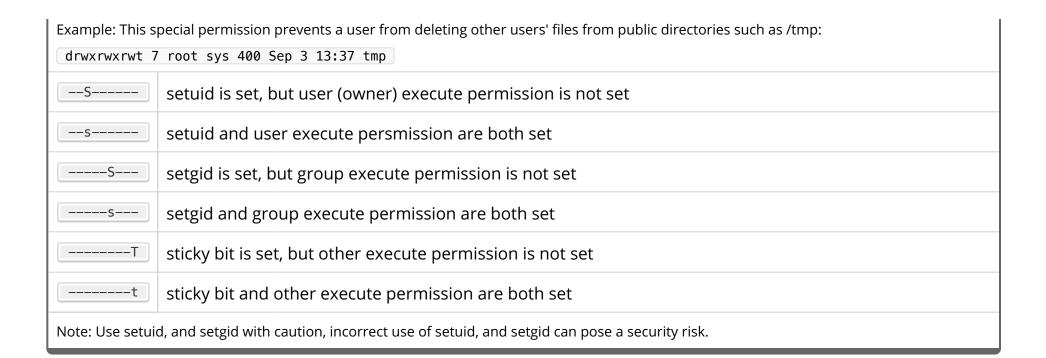
Example: The /usr/bin/mail command has setgid permissions:

-r-x--s--x 1 root mail 63628 Sep 16 12:01 /usr/bin/mail

#### Sticky Bit

The sticky bit is a permission bit that protects the files within a directory.

If the directory has the sticky bit set, a file can be deleted only by the owner of the file, the owner of the directory, or by root.



### **Notes**

- Based on these articles:
  - <u>Understanding and Setting UNIX File Permissions</u>
  - o Linux File Permissions, chmod, & umask
  - How to use SETUID SETGID and Stickybit Permissions
  - o Special File Permissions (setuid, setgid and Sticky Bit)
- Converted by Wesley Hill