

Requirements Document: Fix-Spaces Shell Script (fix-spaces.sh)

Field	Value
Project	Fix-Spaces Shell Script (fix-spaces.sh)
Version	1.0
Date	2025-10-19
Author	Sanjeet Prasad
Email	sanjeet8.23@gmail.com
LinkedIn	https://www.linkedin.com/in/sanjeet9019/
GitHub	https://github.com/sanjeet9019/

1. Project Goal and Constraints

The goal is to develop a shell script that recursively renames files and directories containing **spaces** in their names by replacing those spaces with **dashes (-)**. The script must support selective renaming via command-line flags and perform safe, error-checked operations.

Requirement	Description
Core Logic	Replace all spaces in file and directory names with dashes. Multiple spaces become multiple dashes.
Input/Output	Accepts -f, -d, or both flags, followed by a directory name. Renames files and/or directories recursively.
Constraint	Must not overwrite existing files. Must not use temporary files. Must handle permission errors gracefully.
Environment	Must be a Bourne-compatible shell script (#!/bin/bash). Must run on Ubuntu 22.04 or equivalent.

2. Functional Requirements

2.1. R1: Command-Line Option Handling

Option	Behaviour
-f	Rename files only
-d	Rename directories only
-f -d	Rename both files and directories
directory-name	Final argument; specifies the target directory to process

- At least one of -f or -d must be provided.

- The script must reject . and .. as target directories.

2.2. R2: Recursive Renaming Logic

Component	Requirement
Files	Use find -type f -name "*" "*" to locate files with spaces.
Directories	Use find -type d -name "*" "*" to locate directories with spaces.
Renaming	Replace spaces with dashes using string substitution.
Multiple Spaces	Convert "a b" to "a----b" (preserve dash count).

2.3. R3: my_rename() Function

Check	Description
Writable Parent	Ensure the parent directory of \$1 is writable.
Collision Check	Ensure \$2 does not already exist.
Rename Execution	Use mv "\$1" "\$2" and check exit status.
Error Reporting	Print clear error messages on failure.

3. Test Case List (Automated Script Validation) The script must pass the following 10 core tests as defined in the automated grading harness.

Test ID	Category	Scenario	Expected Behaviour
T1	Error Handling	No arguments	Exit with usage message
T2	Error Handling	Invalid flag	Exit with error
T3	Error Handling	Directory is . or ..	Exit with error
T4	Basic	Rename files only (-f)	Files renamed, directories untouched
T5	Basic	Rename directories only (-d)	Directories renamed, files untouched
T6	Combined	Rename both (-f -d)	All names processed
T7	Nested	Deep directory structure	All levels renamed correctly
T8	Multiple Spaces	"a b" → "a----b"	Dash count matches space count
T9	Collision	Target name already exists	Skip with warning
T10	Permissions	Read-only directory	Handle gracefully, no crash

