

Goat Tool Documentation

Authors: - Deautau Ross - Jaret Smith - Johan Olvera - ChatGPT

How to Install (For Unix/Linux-Based OSes) 1. Clone the repository using `git clone https://github.com/dross1-dross1/goat-tool.git` 2. CD into the directory using `cd goat-tool/` 3. Compile and build the code with `make` or `./build.sh` 4. You're done! Use GoatTool by running (in only this directory) `./bin/goattool [--flag] (args)` 5. (Optional) Run the test script using `./tests.sh` (runs automatically if you compiled with `./build.sh`)

Table of Contents

1. [Project File Structure](#)
 2. [Introduction](#)
 3. [Command Switches](#)
 4. [Usage Examples](#)
 5. [Functions](#)
 6. [Function Call Graph](#)
 7. [Test Cases](#)
 8. [Code Test Analysis](#)
 9. [BASH Script Raw Output](#)
 10. [Privacy Disclosure](#)
-

Project File Structure

```
GoatTool/
|
├── LICENSE
|
├── README.md
|
├── .gitignore
|
├── Makefile
|
├── tests.sh
|
├── build.sh
|
├── docs/
|   └── DESIGN_DOCUMENT.md
```

```
|   └─ DESIGN_DOCUMENT.pdf
|
|─ src/
|   ├── main.c
|   ├── display.c
|   ├── file_operations.c
|   ├── search.c
|   ├── compression.c
|   ├── permissions.c
|   └─ utils.c
|
|─ include/
|   ├── display.h
|   ├── file_operations.h
|   ├── search.h
|   ├── compression.h
|   ├── permissions.h
|   └─ utils.h
|
|─ test_files/
|   ├── example_file.txt
|   ├── file1.txt
|   ├── file2.txt
|   ├── empty_file.txt
|   ├── special@file#.txt
|   ├── locked_file.txt
|   ├── locked_file.txt.goat
|   └─ locked_file.txt.goat.decompressed
|
|─ assets/
|   ├── function_call_diagram.png
|   └─ function_call_diagram.drawio
```

Introduction

Goat Tool is a command-line utility for Unix environments, designed to aid in various file and directory operations. To utilize the tool, navigate to a Unix terminal and enter the `./goattool` command followed by the desired switches.

Command Switches

- -h: Display the help menu.
 - Usage: `./goattool -h`
 - -p [FILENAMES...]: Print the contents of one or more files.
 - Usage: `./goattool -p example.txt`
 - -s [FILENAME]: Search a file for a string and display lines containing that string.
 - Usage: `./goattool -s "search_term" example.txt`
 - -z [FILENAME]: Print the size of a file.
 - Usage: `./goattool -z example.txt`
 - -m [OUTPUT_FILE] [INPUT_FILES...]: Merge multiple files.
 - Usage: `./goattool -m merged.txt file1.txt file2.txt`
 - -c [FILENAME]: Compress a file into .goat format.
 - Usage: `./goattool -c example.txt`
 - -d [FILENAME]: Decompress a .goat file.
 - Usage: `./goattool -d example.goat`
 - -r [FILENAME]: Print file permissions.
 - Usage: `./goattool -r example.txt`
 - -v [SOURCE] [DESTINATION]: Move or rename a file.
 - Usage: `./goattool -v source.txt dest.txt`
-

Usage Examples

1. Printing File Content:

```
> ./goattool -p example.txt
This is an example text file.
```

1. Searching for a String:

```
> ./goattool -s "text" example.txt
This is an example text file.
```

1. Compressing and Decompressing:

```
> ./goattool -c example.txt
File compressed to example.goat
```

```
> ./goattool -d example.goat
File decompressed to example.txt
```

Functions

int main(int argc, char argv)**

- **Description:** Entry point for the GoatTool program. Processes command-line arguments to execute corresponding functionalities.
- **Parameters:**
 - argc: Count of command-line arguments.
 - argv: Array of command-line arguments.
- **Return Value:** Returns 0 for successful execution, -1 for errors.
- **Side Effects:** Executes various functionalities based on command-line arguments and outputs respective results or error messages.

int display_help()

- **Description:** Displays help and usage instructions for GoatTool.
- **Parameters:** None.
- **Return Value:** Always returns 0.
- **Side Effects:** Outputs help information to the console.

int print_file_content(int num_files, const char filenames)**

- **Description:** Prints the content of specified files.
- **Parameters:**
 - num_files: The number of files to print.
 - filenames: Array of file paths.
- **Return Value:** 0 on success, non-zero on failure.
- **Side Effects:** Outputs file content or error messages to the console.

int print_size(const char* filename)

- **Description:** Prints the size of the specified file in bytes.
- **Parameters:**
 - filename: Path to the file.
- **Return Value:** 0 on success, -1 on failure.
- **Side Effects:** Outputs file size to the console or error messages.

int merge_file(const char* output_file, int num_files, const char input_files)**

- **Description:** Merges multiple files into one output file.
- **Parameters:**
 - output_file: Path to the output file.
 - num_files: Number of input files.
 - input_files: Array of paths to the input files.
- **Return Value:** 0 on success, -1 on failure.

- **Side Effects:** Merges files and outputs status messages.

int move_file(const char* source, const char* destination)

- **Description:** Moves or renames a file.
- **Parameters:**
 - source: Path to the source file.
 - destination: Path to the destination file or new file name.
- **Return Value:** 0 on success, -1 on failure.
- **Side Effects:** Moves/renames the file and outputs status messages.

int search_files(const char* search_term, const char* filename)

- **Description:** Searches for a specific term within a file and prints the lines containing that term.
- **Parameters:**
 - search_term: Term to search for within the file.
 - filename: Path to the file in which to search.
- **Return Value:** The number of lines where the search_term is found, or -1 in case of error.
- **Side Effects:** Outputs matching lines or error messages to the console.

int compress_goat(const char* filename)

- **Description:** Compresses a file into .goat format.
- **Parameters:**
 - filename: Path of the file to compress.
- **Return Value:** 0 on success, -1 on failure.
- **Side Effects:** Compresses file and outputs status messages.

int decompress_goat(const char* filename)

- **Description:** Decompresses a .goat file to its original content.
- **Parameters:**
 - filename: Path of the .goat file to decompress.
- **Return Value:** 0 on success, -1 on failure.
- **Side Effects:** Decompresses file and outputs status messages.

int print_file_permissions(const char* filename)

- **Description:** Prints the file permissions of the specified file as an octal number.
- **Parameters:**
 - filename: Path of the file whose permissions are to be printed.
- **Return Value:** 0 on success, -1 on failure.

- **Side Effects:** Outputs file permissions to the console or error messages.
-

Function Call Graph

Function Call Diagram
Function Call Diagram

Test Cases

Operation 1: -h (Display the help menu)

1. Print the help menu

- Input: `./goattool -h`

- Expected Output:

```
GoatTool Help and Usage Instructions
-----
-h : Display this help menu.
-p [FILENAMES...] : Print the contents of one or more
files.
-s [SEARCH_TERM] [FILENAME] : Search a file for a string.
-z [FILENAME] : Print the size of a file.
-m [OUTPUT_FILE] [INPUT_FILES...] : Merge multiple files.
-c [FILENAME] : Compress a file into .goat format.
-d [FILENAME] : Decompress a .goat file.
-r [FILENAME] : Print file permissions.
-v [SOURCE] [DESTINATION] : Move or rename a file.
```

Operation 2: -p [FILENAME] (Print the contents of a file)

1. Print the contents of a single file

- Input: `./goattool -p example_file.txt`

- Expected Output:

```
--- Content of 'test_files/example_file.txt' ---
This is an example file.
It has multiple lines.
And it is used for testing purposes.
Make sure you have fun testing!
```

```
This is a line with search_term.  
aaaaaaaaaaaaaaaaabbbb cccdde
```

2. Print the contents of a file that doesn't exist

- Input: `./goattool -p non_existent_file.txt`

- Expected Output:

```
Error: Cannot access 'test_files/non_existent_file.txt'  
because it does not exist.
```

3. Print the contents of multiple existing files

- Input: `./goattool -p file1.txt file2.txt`

- Expected Output:

```
--- Content of 'test_files/file1.txt' ---  
This is the first file.  
It will be merged with another.  
--- Content of 'test_files/file2.txt' ---  
This is the second file.  
Merged content goes below the first file's content.
```

4. Print the contents of a mix of existing and non-existing files

- Input: `./goattool -p file1.txt non_existent_file.txt`

- Expected Output:

```
--- Content of 'test_files/file1.txt' ---  
This is the first file.  
It will be merged with another.  
Error: Cannot access 'test_files/non_existent_file.txt'  
because it does not exist.
```

5. Print the contents with one of the files being a directory

- Input: `./goattool -p example_file.txt test_files`

- Expected Output:

```
--- Content of 'test_files/example_file.txt' ---  
This is an example file.  
It has multiple lines.  
And it is used for testing purposes.  
Make sure you have fun testing!  
This is a line with search_term.  
aaaaaaaaaaaaaaaaabbbb cccdde
```

Error: Cannot access 'test_files' because it is a directory.

6. Print the contents of an empty file

- Input: `./goattool -p empty_file.txt`

- Expected Output:

--- Content of 'test_files/empty_file.txt' ---

7. Print the contents with no files specified

- Input: `./goattool -p`

- Expected Output:

Error: Invalid command switch or missing arguments. Use -h for help.

Operation 3: -s [SEARCH_TERM] [FILENAME] (Search a file for a string)

1. Search for a term in a file (existing term)

- Input: `./goattool -s "search_term" example_file.txt`

- Expected Output: "Line 5: This is a line with search_term."

2. Search for a term that doesn't exist in the file

- Input: `./goattool -s "missing_term" example_file.txt`

- Expected Output: "No matches found for 'missing_term'."

3. No search term or filename provided

- Input: `./goattool -s`

- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

4. No filename provided

- Input: `./goattool -s "search_term"`

- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

5. File does not exist

- Input: `./goattool -s "search_term" non_existent_file.txt`

- Expected Output: "Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist."

6. Search term is an empty string

- Input: `./goattool -s "" example_file.txt`

- Expected Output:

Line 1: This is an example file.

Line 2: It has multiple lines.

Line 3: And it is used for testing purposes.

Line 4: Make sure you have fun testing!
Line 5: This is a line with search_term.
Line 6: aaaaaaaaaaaaaaaaaabbbb cccdde

7. Search in an empty file

- Input: `./goattool -s "search_term" empty_file.txt`
- Expected Output: "No matches found for 'search_term'."

Operation 4: `-z [FILENAME]` (Print the size of a file)

1. Print the size of an existing file

- Input: `./goattool -z example_file.txt`
- Expected Output: "Size: 180 bytes."

2. No filename provided

- Input: `./goattool -z`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

3. File does not exist

- Input: `./goattool -z non_existent_file.txt`
- Expected Output: "Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist."

4. Specified path is a directory

- Input: `./goattool -z test_files`
- Expected Output: "Error: Cannot access 'test_files' because it is a directory."

5. File with access denied (locked file)

- Input: `./goattool -z locked_file.txt`
- Expected Output: "Error: Cannot access 'test_files/locked_file.txt' due to insufficient permissions."

6. Print the size of an empty file

- Input: `./goattool -z empty_file.txt`
- Expected Output: "Size: 0 bytes."

7. File with special characters in the name

- Input: `./goattool -z special@file#.txt`
- Expected Output: "Size: 22 bytes."

Operation 5: `-m [OUTPUT_FILE] [INPUT_FILES...]` (Merge multiple files)

1. Merge two files together

- Input: `./goattool -m merged.txt file1.txt file2.txt`
- Expected Output:

```
Files merged successfully into 'test_files/merged.txt'.  
--- Content of 'test_files/merged.txt' ---  
This is the first file.
```

It will be merged with another.
This is the second file.
Merged content goes below the first file's content.

2. Merge with a non-existent file

- Input: `./goattool -m merged.txt non_existent_file.txt file2.txt`
- Expected Output: "Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist."

3. No output or input files provided

- Input: `./goattool -m`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

4. Only output file provided, no input files

- Input: `./goattool -m merged.txt`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

5. Merge with a file with access denied

- Input: `./goattool -m merged.txt file1.txt locked_file.txt`
- Expected Output: "Error: Cannot access 'test_files/locked_file.txt' due to insufficient permissions."

6. Merge with a directory instead of a file

- Input: `./goattool -m merged.txt test_files file2.txt`
- Expected Output: "Error: Cannot access 'test_files' because it is a directory."

7. Merge with an empty file

- Input: `./goattool -m merged.txt empty_file.txt file2.txt`
- Expected Output:

`Files merged successfully into 'test_files/merged.txt'.
--- Content of 'test_files/merged.txt' ---
This is the second file.
Merged content goes below the first file's content.`

Operation 6: -c [FILENAME] (Compress a file into .goat format)

1. Compress an existing file

- Input: `./goattool -c example_file.txt`
- Expected Output:

`File processed to test_files/example_file.txt.goat.
--- Content of 'test_files/example_file.txt.goat' ---
T1h1i1s1 i1s1 a1n1 e1x1a1m1p1l1e1 f1i1l1e1.
I1t1 h1a1s1 m1u1l1t1i1p1l1e1 l1i1n1e1s1.`

```
A1n1d1 i1t1 i1s1 u1s1e1d1 f1o1r1 t1e1s1t1i1n1g1
p1u1r1p1o1s1e1s1.
M1a1k1e1 s1u1r1e1 y1o1u1 h1a1v1e1 f1u1n1 t1e1s1t1i1n1g1!
T1h1i1s1 i1s1 a1 l1i1n1e1 w1i1t1h1 s1e1a1r1c1h1_t1e1r1m1.
a18b4 c3d2e1
```

2. No filename provided

- Input: `./goattool -c`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

3. File does not exist

- Input: `./goattool -c non_existent_file.txt`
- Expected Output: "Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist."

4. Specified path is a directory

- Input: `./goattool -c test_files`
- Expected Output: "Error: Cannot access 'test_files' because it is a directory."

5. File with access denied (locked file)

- Input: `./goattool -c locked_file.txt`
- Expected Output: "Error: Cannot access 'test_files/locked_file.txt' due to insufficient permissions."

6. Compress an empty file

- Input: `./goattool -c empty_file.txt`

- Expected Output:

```
File processed to test_files/empty_file.txt.goat.
--- Content of 'test_files/empty_file.txt.goat' ---
```

7. Compress a file with special characters in the name

- Input: `./goattool -c special@file#.txt`

- Expected Output:

```
File processed to test_files/special@file#.txt.goat.
--- Content of 'test_files/special@file#.txt.goat' ---
#!m1 @ \ $p1[-c1!@| / = ! | [-
```

Operation 7: -d [FILENAME] (Decompress a .goat file)

1. Decompress an existing .goat file

- Input: `./goattool -d example_file.txt.goat`
- Expected Output:

```

File processed to test_files/
example_file.txt.goat.decompressed.
--- Content of 'test_files/
example_file.txt.goat.decompressed' ---
This is an example file.
It has multiple lines.
And it is used for testing purposes.
Make sure you have fun testing!
This is a line with search_term.
aaaaaaaaaaaaaaaaabbbb cccdde

```

2. No filename provided

- Input: `./goattool -d`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

3. .goat file does not exist

- Input: `./goattool -d non_existent_file.txt.goat`
- Expected Output: "Error: Cannot access 'test_files/non_existent_file.txt.goat' because it does not exist."

4. Specified path is a directory

- Input: `./goattool -d test_files`
- Expected Output: "Error: Cannot access 'test_files' because it is a directory."

5. .goat file with access denied

- Input: `./goattool -d locked_file.txt.goat`
- Expected Output: "Error: Cannot access 'test_files/locked_file.txt.goat' due to insufficient permissions."

6. Decompress an empty .goat file

- Input: `./goattool -d empty_file.txt.goat`
- Expected Output:

```

File processed to test_files/
empty_file.txt.goat.decompressed.
--- Content of 'test_files/
empty_file.txt.goat.decompressed' ---

```

7. .goat file with special characters in the name

- Input: `./goattool -d special@file#.txt.goat`
- Expected Output:

```

File processed to test_files/
special@file#.txt.goat.decompressed.
--- Content of 'test_files/
special@file#.txt.goat.decompressed' ---
#!m @ \ $p [-c!@| / = ! | [-

```

Operation 8: -r [FILENAME] (Print file permissions)

1. Existing file with standard permissions

- Input: `./goattool -r example_file.txt`
- Expected Output: "Permissions for 'test_files/example_file.txt': 664"

2. No filename provided

- Input: `./goattool -r`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

3. Non-existent file

- Input: `./goattool -r non_existent_file.txt`
- Expected Output: "Error: File 'test_files/non_existent_file.txt' does not exist."

4. Locked file

- Input: `./goattool -r locked_file.txt`
- Expected Output: "Permissions for 'test_files/locked_file.txt': 000"

5. Directory

- Input: `./goattool -r test_files`
- Expected Output: "Permissions for 'test_files': 775"

6. File with special characters in name

- Input: `./goattool -r special@file#.txt`
- Expected Output: "Permissions for 'test_files/special@file#.txt': 664"

7. Executable file

- Input: `./goattool -r empty_file.txt`
- Expected Output: "Permissions for 'test_files/empty_file.txt': 777"

Operation 9: -v [SOURCE] [DESTINATION] (Move or rename a file)

1. Locked source file

- Input: `./goattool -v locked_file.txt destination_file.txt`
- Expected Output: "Error: Cannot access 'test_files/locked_file.txt' due to insufficient permissions."

2. Successful file move/rename

- Input: `./goattool -v merged.txt destination_file.txt`
- Expected Output:

File moved/renamed successfully.
--- Content of 'test_files/destination_file.txt' ---
This is the second file.
Merged content goes below the first file's content.

3. No source or destination provided

- Input: `./goattool -v`

- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

4. **Non-existent source file**

- Input: `./goattool -v non_existent_file.txt destination_file.txt`
- Expected Output: "Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist."

5. **No destination provided**

- Input: `./goattool -v merged.txt`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

6. **Source is a directory**

- Input: `./goattool -v test_files destination_file.txt`
- Expected Output: "Error: Cannot access 'test_files' because it is a directory."

7. **Destination already exists**

- Input: `./goattool -v example_file.txt.goat destination_file.txt`
- Expected Output:

File moved/renamed successfully.
--- Content of 'test_files/destination_file.txt' ---
T1h1i1s1 i1s1 a1n1 e1x1a1m1p1l1e1 f1i1l1e1.
I1t1 h1a1s1 m1u1l1t1i1p1l1e1 l1i1n1e1s1.
A1n1d1 i1t1 i1s1 u1s1e1d1 f1o1r1 t1e1s1t1i1n1g1
p1u1r1p1o1s1e1s1.
M1a1k1e1 s1u1r1e1 y1o1u1 h1a1v1e1 f1u1n1 t1e1s1t1i1n1g1!
T1h1i1s1 i1s1 a1 l1i1n1e1 w1i1t1h1 s1e1a1r1c1h1_t1e1r1m1.
a18b4 c3d2e1

Default Cases (No switch provided or invalid switch provided)

1. **No arguments provided**

- Input: `./goattool`
- Expected Output: "Error: Insufficient argument count. Use -h for help."

2. **Invalid switch provided**

- Input: `./goattool -e`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

3. **No switch provided but arguments exist**

- Input: `./goattool file1.txt file2.txt`
- Expected Output: "Error: Invalid command switch or missing arguments. Use -h for help."

Sample Files: 1. **example_file.txt**:

This is an example file.
It has multiple lines.
And it is used for testing purposes.
Make sure you have fun testing!
This is a line with search_term.
aaaaaaaaaaaaaaaaabbbb cccdde

1. **file1.txt**:

This is the first file.
It will be merged with another.

1. **file2.txt**:

This is the second file.
Merged content goes below the first file's content.

1. **special@file#.txt**:

#!m @ \$p[-c!@| /=!|[-

1. **empty_file.txt**:

1. **locked_file.txt**: (Note: This file is locked or has permissions set to disallow reading. Contents cannot be displayed.)

You shouldn't be seeing this, as this file is locked.

1. **locked_file.txt.goat** (Note: This file is locked or has permissions set to disallow reading. Contents cannot be displayed.)

Y1o1u1 s1h1o1u1l1d1n1't1 b1e1 s1e2i1n1g1 t1h1i1s1, a1s1 t1h1i1s1
f1i1l1e1 i1s1 l1o1c1k1e1d1.

1. **locked_file.txt.goat.decompressed** (Note: This file is locked or has permissions set to disallow reading. Contents cannot be displayed.)

You shouldn't be seeing this, as this file is locked.

Code Test Analysis

1. Test Coverage

Our code testing strategy has been comprehensive, achieving 100% coverage. This means that every line of code within the program was

exercised during the testing process, ensuring that each function and pathway was rigorously evaluated for correctness.

2. Test Case Results

The results of our test cases were highly positive, with every test passing successfully. This was verified through an automated process where the output of each test was compared against an expected outcome using a Bash script. Such a system allowed for precise and efficient verification of each test case's success or failure.

3. Real-World Predictability

While our test cases provide a strong indicator of the program's behavior, it's important to note that they are not a perfect representation of real-world usage. In actual deployment, users often interact with software in ways that are difficult to predict or replicate in controlled test environments. Unforeseen variables such as user input patterns, system configurations, and interaction with other applications can affect the program's behavior. Therefore, while our tests are thorough and give us confidence in the program's reliability, we remain aware of the potential for unexpected issues in diverse real-world scenarios.

BASH Script Raw Output

```
[GT-Test] START
[GT-Test] Preparing test environment
[GT-Test] Test environment prepared

[GT-Test] [-h] Test case: Print the help menu
[GT-Test] CASE: ./bin/goattool -h
[GT-Test] EXPECTED:
GoatTool Help and Usage Instructions
-----
-h : Display this help menu.
-p [FILENAMES...] : Print the contents of one or more files.
-s [SEARCH_TERM] [FILENAME] : Search a file for a string.
-z [FILENAME] : Print the size of a file.
-m [OUTPUT_FILE] [INPUT_FILES...] : Merge multiple files.
-c [FILENAME] : Compress a file into .goat format.
-d [FILENAME] : Decompress a .goat file.
-r [FILENAME] : Print file permissions.
-v [SOURCE] [DESTINATION] : Move or rename a file.
[GT-Test] ACTUAL:
GoatTool Help and Usage Instructions
-----
```


-h : Display this help menu.
-p [FILENAMES...] : Print the contents of one or more files.
-s [SEARCH_TERM] [FILENAME] : Search a file for a string.
-z [FILENAME] : Print the size of a file.
-m [OUTPUT_FILE] [INPUT_FILES...] : Merge multiple files.
-c [FILENAME] : Compress a file into .goat format.
-d [FILENAME] : Decompress a .goat file.
-r [FILENAME] : Print file permissions.
-v [SOURCE] [DESTINATION] : Move or rename a file.
[GT-Test] RESULT: PASS

[GT-Test] [-p] Test case: Print the contents of a single file
[GT-Test] CASE: ./bin/goattool -p test_files/example_file.txt
[GT-Test] EXPECTED:
--- Content of 'test_files/example_file.txt' ---
This is an example file.
It has multiple lines.
And it is used for testing purposes.
Make sure you have fun testing!
This is a line with search_term.
aaaaaaaaaaaaaaaaabbbb cccdde
[GT-Test] ACTUAL:
--- Content of 'test_files/example_file.txt' ---
This is an example file.
It has multiple lines.
And it is used for testing purposes.
Make sure you have fun testing!
This is a line with search_term.
aaaaaaaaaaaaaaaaabbbb cccdde
[GT-Test] RESULT: PASS

[GT-Test] [-p] Test case: Print the contents of a file that
doesn't exist
[GT-Test] CASE: ./bin/goattool -p test_files/
non_existent_file.txt
[GT-Test] EXPECTED:
Error: Cannot access 'test_files/non_existent_file.txt' because
it does not exist.
[GT-Test] ACTUAL:
Error: Cannot access 'test_files/non_existent_file.txt' because
it does not exist.
[GT-Test] RESULT: PASS

[GT-Test] [-p] Test case: Print the contents of multiple existing

files

[GT-Test] CASE: ./bin/goattool -p test_files/file1.txt
test_files/file2.txt

[GT-Test] EXPECTED:

--- Content of 'test_files/file1.txt' ---

This is the first file.

It will be merged with another.

--- Content of 'test_files/file2.txt' ---

This is the second file.

Merged content goes below the first file's content.

[GT-Test] ACTUAL:

--- Content of 'test_files/file1.txt' ---

This is the first file.

It will be merged with another.

--- Content of 'test_files/file2.txt' ---

This is the second file.

Merged content goes below the first file's content.

[GT-Test] RESULT: PASS

[GT-Test] [-p] Test case: Print the contents of a mix of existing
and non-existing files

[GT-Test] CASE: ./bin/goattool -p test_files/file1.txt
test_files/non_existent_file.txt

[GT-Test] EXPECTED:

--- Content of 'test_files/file1.txt' ---

This is the first file.

It will be merged with another.

Error: Cannot access 'test_files/non_existent_file.txt' because
it does not exist.

[GT-Test] ACTUAL:

--- Content of 'test_files/file1.txt' ---

This is the first file.

It will be merged with another.

Error: Cannot access 'test_files/non_existent_file.txt' because
it does not exist.

[GT-Test] RESULT: PASS

[GT-Test] [-p] Test case: Print the contents with one of the
files being a directory

[GT-Test] CASE: ./bin/goattool -p test_files/example_file.txt
test_files

[GT-Test] EXPECTED:

--- Content of 'test_files/example_file.txt' ---

This is an example file.

It has multiple lines.
And it is used for testing purposes.
Make sure you have fun testing!
This is a line with search_term.
aaaaaaaaaaaaaaaaaaaaabbbb cccdde
Error: Cannot access 'test_files' because it is a directory.
[GT-Test] ACTUAL:
--- Content of 'test_files/example_file.txt' ---
This is an example file.
It has multiple lines.
And it is used for testing purposes.
Make sure you have fun testing!
This is a line with search_term.
aaaaaaaaaaaaaaaaaaaaabbbb cccdde
Error: Cannot access 'test_files' because it is a directory.
[GT-Test] RESULT: PASS

[GT-Test] [-p] Test case: Print the contents of an empty file
[GT-Test] CASE: ./bin/goattool -p test_files/empty_file.txt
[GT-Test] EXPECTED:
--- Content of 'test_files/empty_file.txt' ---
[GT-Test] ACTUAL:
--- Content of 'test_files/empty_file.txt' ---
[GT-Test] RESULT: PASS

[GT-Test] [-p] Test case: Print the contents with no files
specified
[GT-Test] CASE: ./bin/goattool -p
[GT-Test] EXPECTED:
Error: Invalid command switch or missing arguments. Use -h for
help.
[GT-Test] ACTUAL:
Error: Invalid command switch or missing arguments. Use -h for
help.
[GT-Test] RESULT: PASS

[GT-Test] [-s] Test case: Search for a term in a file (existing
term)
[GT-Test] CASE: ./bin/goattool -s "search_term" test_files/
example_file.txt
[GT-Test] EXPECTED:
Line 5: This is a line with search_term.
[GT-Test] ACTUAL:
Line 5: This is a line with search_term.

[GT-Test] RESULT: PASS

[GT-Test] [-s] Test case: Search for a term that doesn't exist in the file

[GT-Test] CASE: ./bin/goattool -s "missing_term" test_files/example_file.txt

[GT-Test] EXPECTED:

No matches found for 'missing_term'.

[GT-Test] ACTUAL:

No matches found for 'missing_term'.

[GT-Test] RESULT: PASS

[GT-Test] [-s] Test case: No search term or filename provided

[GT-Test] CASE: ./bin/goattool -s

[GT-Test] EXPECTED:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] ACTUAL:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] RESULT: PASS

[GT-Test] [-s] Test case: No filename provided

[GT-Test] CASE: ./bin/goattool -s "search_term"

[GT-Test] EXPECTED:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] ACTUAL:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] RESULT: PASS

[GT-Test] [-s] Test case: File does not exist

[GT-Test] CASE: ./bin/goattool -s "search_term" test_files/non_existent_file.txt

[GT-Test] EXPECTED:

Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist.

[GT-Test] RESULT: PASS

[GT-Test] [-s] Test case: Search term is an empty string

```
[GT-Test] CASE: ./bin/goattool -s "" test_files/example_file.txt
[GT-Test] EXPECTED:
Line 1: This is an example file.
Line 2: It has multiple lines.
Line 3: And it is used for testing purposes.
Line 4: Make sure you have fun testing!
Line 5: This is a line with search_term.
Line 6: aaaaaaaaaaaaaaaaaabbbb cccdde
[GT-Test] ACTUAL:
Line 1: This is an example file.
Line 2: It has multiple lines.
Line 3: And it is used for testing purposes.
Line 4: Make sure you have fun testing!
Line 5: This is a line with search_term.
Line 6: aaaaaaaaaaaaaaaaaabbbb cccdde
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-s] Test case: Search in an empty file
[GT-Test] CASE: ./bin/goattool -s "search_term" test_files/
empty_file.txt
[GT-Test] EXPECTED:
No matches found for 'search_term'.
[GT-Test] ACTUAL:
No matches found for 'search_term'.
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-z] Test case: Print the size of an existing file
[GT-Test] CASE: ./bin/goattool -z test_files/example_file.txt
[GT-Test] EXPECTED:
Size: 180 bytes.
[GT-Test] ACTUAL:
Size: 180 bytes.
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-z] Test case: No filename provided
[GT-Test] CASE: ./bin/goattool -z
[GT-Test] EXPECTED:
Error: Invalid command switch or missing arguments. Use -h for
help.
[GT-Test] ACTUAL:
Error: Invalid command switch or missing arguments. Use -h for
help.
[GT-Test] RESULT: PASS
```

[GT-Test] [-z] Test case: File does not exist
[GT-Test] CASE: ./bin/goattool -z test_files/
non_existent_file.txt
[GT-Test] EXPECTED:
Error: Cannot access 'test_files/non_existent_file.txt' because
it does not exist.
[GT-Test] ACTUAL:
Error: Cannot access 'test_files/non_existent_file.txt' because
it does not exist.
[GT-Test] RESULT: PASS

[GT-Test] [-z] Test case: Specified path is a directory
[GT-Test] CASE: ./bin/goattool -z test_files
[GT-Test] EXPECTED:
Error: Cannot access 'test_files' because it is a directory.
[GT-Test] ACTUAL:
Error: Cannot access 'test_files' because it is a directory.
[GT-Test] RESULT: PASS

[GT-Test] [-z] Test case: File with access denied (locked file)
[GT-Test] CASE: ./bin/goattool -z test_files/locked_file.txt
[GT-Test] EXPECTED:
Error: Cannot access 'test_files/locked_file.txt' due to
insufficient permissions.
[GT-Test] ACTUAL:
Error: Cannot access 'test_files/locked_file.txt' due to
insufficient permissions.
[GT-Test] RESULT: PASS

[GT-Test] [-z] Test case: Print the size of an empty file
[GT-Test] CASE: ./bin/goattool -z test_files/empty_file.txt
[GT-Test] EXPECTED:
Size: 0 bytes.
[GT-Test] ACTUAL:
Size: 0 bytes.
[GT-Test] RESULT: PASS

[GT-Test] [-z] Test case: File with special characters in the
name
[GT-Test] CASE: ./bin/goattool -z test_files/special@file#.txt
[GT-Test] EXPECTED:
Size: 22 bytes.
[GT-Test] ACTUAL:
Size: 22 bytes.

[GT-Test] RESULT: PASS

[GT-Test] [-m] Test case: Merge two files together

[GT-Test] CASE: ./bin/goattool -m test_files/merged.txt
test_files/file1.txt test_files/file2.txt && ./bin/goattool -p
test_files/merged.txt

[GT-Test] EXPECTED:

Files merged successfully into 'test_files/merged.txt'.

--- Content of 'test_files/merged.txt' ---

This is the first file.

It will be merged with another.

This is the second file.

Merged content goes below the first file's content.

[GT-Test] ACTUAL:

Files merged successfully into 'test_files/merged.txt'.

--- Content of 'test_files/merged.txt' ---

This is the first file.

It will be merged with another.

This is the second file.

Merged content goes below the first file's content.

[GT-Test] RESULT: PASS

[GT-Test] [-m] Test case: Merge with a non-existent file

[GT-Test] CASE: ./bin/goattool -m test_files/merged.txt
test_files/non_existent_file.txt test_files/file2.txt

[GT-Test] EXPECTED:

Error: Cannot access 'test_files/non_existent_file.txt' because
it does not exist.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files/non_existent_file.txt' because
it does not exist.

[GT-Test] RESULT: PASS

[GT-Test] [-m] Test case: No output or input files provided

[GT-Test] CASE: ./bin/goattool -m

[GT-Test] EXPECTED:

Error: Invalid command switch or missing arguments. Use -h for
help.

[GT-Test] ACTUAL:

Error: Invalid command switch or missing arguments. Use -h for
help.

[GT-Test] RESULT: PASS

[GT-Test] [-m] Test case: Only output file provided, no input

files

[GT-Test] CASE: ./bin/goattool -m test_files/merged.txt

[GT-Test] EXPECTED:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] ACTUAL:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] RESULT: PASS

[GT-Test] [-m] Test case: Merge with a file with access denied

[GT-Test] CASE: ./bin/goattool -m test_files/merged.txt

test_files/file1.txt test_files/locked_file.txt

[GT-Test] EXPECTED:

Error: Cannot access 'test_files/locked_file.txt' due to insufficient permissions.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files/locked_file.txt' due to insufficient permissions.

[GT-Test] RESULT: PASS

[GT-Test] [-m] Test case: Merge with a directory instead of a file

[GT-Test] CASE: ./bin/goattool -m test_files/merged.txt

test_files test_files/file2.txt

[GT-Test] EXPECTED:

Error: Cannot access 'test_files' because it is a directory.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files' because it is a directory.

[GT-Test] RESULT: PASS

[GT-Test] [-m] Test case: Merge with an empty file

[GT-Test] CASE: ./bin/goattool -m test_files/merged.txt

test_files/empty_file.txt test_files/file2.txt && ./bin/goattool
-p test_files/merged.txt

[GT-Test] EXPECTED:

Files merged successfully into 'test_files/merged.txt'.

--- Content of 'test_files/merged.txt' ---

This is the second file.

Merged content goes below the first file's content.

[GT-Test] ACTUAL:

Files merged successfully into 'test_files/merged.txt'.

--- Content of 'test_files/merged.txt' ---

This is the second file.

Merged content goes below the first file's content.

[GT-Test] RESULT: PASS

[GT-Test] [-c] Test case: Compress an existing file

[GT-Test] CASE: ./bin/goattool -c test_files/example_file.txt

&& ./bin/goattool -p test_files/example_file.txt.goat

[GT-Test] EXPECTED:

File processed to test_files/example_file.txt.goat.

--- Content of 'test_files/example_file.txt.goat' ---

T1h1i1s1 i1s1 a1n1 e1x1a1m1p1l1e1 f1i1l1e1.

I1t1 h1a1s1 m1u1l1t1i1p1l1e1 l1i1n1e1s1.

A1n1d1 i1t1 i1s1 u1s1e1d1 f1o1r1 t1e1s1t1i1n1g1 p1u1r1p1o1s1e1s1.

M1a1k1e1 s1u1r1e1 y1o1u1 h1a1v1e1 f1u1n1 t1e1s1t1i1n1g1!

T1h1i1s1 i1s1 a1 l1i1n1e1 w1i1t1h1 s1e1a1r1c1h1_t1e1r1m1.

a18b4 c3d2e1

[GT-Test] ACTUAL:

File processed to test_files/example_file.txt.goat.

--- Content of 'test_files/example_file.txt.goat' ---

T1h1i1s1 i1s1 a1n1 e1x1a1m1p1l1e1 f1i1l1e1.

I1t1 h1a1s1 m1u1l1t1i1p1l1e1 l1i1n1e1s1.

A1n1d1 i1t1 i1s1 u1s1e1d1 f1o1r1 t1e1s1t1i1n1g1 p1u1r1p1o1s1e1s1.

M1a1k1e1 s1u1r1e1 y1o1u1 h1a1v1e1 f1u1n1 t1e1s1t1i1n1g1!

T1h1i1s1 i1s1 a1 l1i1n1e1 w1i1t1h1 s1e1a1r1c1h1_t1e1r1m1.

a18b4 c3d2e1

[GT-Test] RESULT: PASS

[GT-Test] [-c] Test case: No filename provided

[GT-Test] CASE: ./bin/goattool -c

[GT-Test] EXPECTED:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] ACTUAL:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] RESULT: PASS

[GT-Test] [-c] Test case: File does not exist

[GT-Test] CASE: ./bin/goattool -c test_files/
non_existent_file.txt

[GT-Test] EXPECTED:

Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files/non_existent_file.txt' because

it does not exist.

[GT-Test] RESULT: PASS

[GT-Test] [-c] Test case: Specified path is a directory

[GT-Test] CASE: ./bin/goattool -c test_files

[GT-Test] EXPECTED:

Error: Cannot access 'test_files' because it is a directory.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files' because it is a directory.

[GT-Test] RESULT: PASS

[GT-Test] [-c] Test case: File with access denied (locked file)

[GT-Test] CASE: ./bin/goattool -c test_files/locked_file.txt

[GT-Test] EXPECTED:

Error: Cannot access 'test_files/locked_file.txt' due to insufficient permissions.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files/locked_file.txt' due to insufficient permissions.

[GT-Test] RESULT: PASS

[GT-Test] [-c] Test case: Compress an empty file

[GT-Test] CASE: ./bin/goattool -c test_files/empty_file.txt && ./bin/goattool -p test_files/empty_file.txt.goat

[GT-Test] EXPECTED:

File processed to test_files/empty_file.txt.goat.

--- Content of 'test_files/empty_file.txt.goat' ---

[GT-Test] ACTUAL:

File processed to test_files/empty_file.txt.goat.

--- Content of 'test_files/empty_file.txt.goat' ---

[GT-Test] RESULT: PASS

[GT-Test] [-c] Test case: Compress a file with special characters in the name

[GT-Test] CASE: ./bin/goattool -c test_files/special@file#.txt && ./bin/goattool -p test_files/special@file#.txt.goat

[GT-Test] EXPECTED:

File processed to test_files/special@file#.txt.goat.

--- Content of 'test_files/special@file#.txt.goat' ---

#!/m1 @ \$p1[-c1!@| /!=|][-

[GT-Test] ACTUAL:

File processed to test_files/special@file#.txt.goat.

--- Content of 'test_files/special@file#.txt.goat' ---

#!/m1 @ \$p1[-c1!@| /!=|][-

[GT-Test] RESULT: PASS

[GT-Test] [-d] Test case: Decompress an existing .goat file

[GT-Test] CASE: ./bin/goattool -d test_files/

example_file.txt.goat && ./bin/goattool -p test_files/

example_file.txt.goat.decompressed

[GT-Test] EXPECTED:

File processed to test_files/example_file.txt.goat.decompressed.

--- Content of 'test_files/example_file.txt.goat.decompressed'

This is an example file.

It has multiple lines.

And it is used for testing purposes.

Make sure you have fun testing!

This is a line with search_term.

aaaaaaaaaaaaaaaaaaaaabbbb cccdde

[GT-Test] ACTUAL:

File processed to test_files/example_file.txt.goat.decompressed.

--- Content of 'test_files/example_file.txt.goat.decompressed'

This is an example file.

It has multiple lines.

And it is used for testing purposes.

Make sure you have fun testing!

This is a line with search_term.

aaaaaaaaaaaaaaaaaaaaabbbb cccdde

[GT-Test] RESULT: PASS

[GT-Test] [-d] Test case: No filename provided

[GT-Test] CASE: ./bin/goattool -d

[GT-Test] EXPECTED:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] ACTUAL:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] RESULT: PASS

[GT-Test] [-d] Test case: .goat file does not exist

[GT-Test] CASE: ./bin/goattool -d test_files/

non_existent_file.txt.goat

[GT-Test] EXPECTED:

Error: Cannot access 'test_files/non_existent_file.txt.goat' because it does not exist.

[GT-Test] ACTUAL:
Error: Cannot access 'test_files/non_existent_file.txt.goat'
because it does not exist.
[GT-Test] RESULT: PASS

[GT-Test] [-d] Test case: Specified path is a directory
[GT-Test] CASE: ./bin/goattool -d test_files
[GT-Test] EXPECTED:
Error: Cannot access 'test_files' because it is a directory.
[GT-Test] ACTUAL:
Error: Cannot access 'test_files' because it is a directory.
[GT-Test] RESULT: PASS

[GT-Test] [-d] Test case: .goat file with access denied
[GT-Test] CASE: ./bin/goattool -d test_files/locked_file.txt.goat
[GT-Test] EXPECTED:
Error: Cannot access 'test_files/locked_file.txt.goat' due to
insufficient permissions.
[GT-Test] ACTUAL:
Error: Cannot access 'test_files/locked_file.txt.goat' due to
insufficient permissions.
[GT-Test] RESULT: PASS

[GT-Test] [-d] Test case: Decompress an empty .goat file
[GT-Test] CASE: ./bin/goattool -d test_files/empty_file.txt.goat
&& ./bin/goattool -p test_files/empty_file.txt.goat.decompressed
[GT-Test] EXPECTED:
File processed to test_files/empty_file.txt.goat.decompressed.
--- Content of 'test_files/empty_file.txt.goat.decompressed' ---
[GT-Test] ACTUAL:
File processed to test_files/empty_file.txt.goat.decompressed.
--- Content of 'test_files/empty_file.txt.goat.decompressed' ---
[GT-Test] RESULT: PASS

[GT-Test] [-d] Test case: .goat file with special characters in
the name
[GT-Test] CASE: ./bin/goattool -d test_files/
special@file#.txt.goat && ./bin/goattool -p test_files/
special@file#.txt.goat.decompressed
[GT-Test] EXPECTED:
File processed to test_files/special@file#.txt.goat.decompressed.
--- Content of 'test_files/special@file#.txt.goat.decompressed'

#!/m @ \$p[-c!@| /=!|[-

```
[GT-Test] ACTUAL:
File processed to test_files/special@file#.txt.goat.decompressed.
--- Content of 'test_files/special@file#.txt.goat.decompressed'
---
#!/m @ $p[-c!@| /=!|[-
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-r] Test case: Existing file with standard permissions
[GT-Test] CASE: ./bin/goattool -r test_files/example_file.txt
[GT-Test] EXPECTED:
Permissions for 'test_files/example_file.txt': 664
[GT-Test] ACTUAL:
Permissions for 'test_files/example_file.txt': 664
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-r] Test case: No filename provided
[GT-Test] CASE: ./bin/goattool -r
[GT-Test] EXPECTED:
Error: Invalid command switch or missing arguments. Use -h for
help.
[GT-Test] ACTUAL:
Error: Invalid command switch or missing arguments. Use -h for
help.
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-r] Test case: Non-existent file
[GT-Test] CASE: ./bin/goattool -r test_files/
non_existent_file.txt
[GT-Test] EXPECTED:
Error: File 'test_files/non_existent_file.txt' does not exist.
[GT-Test] ACTUAL:
Error: File 'test_files/non_existent_file.txt' does not exist.
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-r] Test case: Locked file
[GT-Test] CASE: ./bin/goattool -r test_files/locked_file.txt
[GT-Test] EXPECTED:
Permissions for 'test_files/locked_file.txt': 000
[GT-Test] ACTUAL:
Permissions for 'test_files/locked_file.txt': 000
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-r] Test case: Directory
[GT-Test] CASE: ./bin/goattool -r test_files
```

[GT-Test] EXPECTED:
Permissions for 'test_files': 775
[GT-Test] ACTUAL:
Permissions for 'test_files': 775
[GT-Test] RESULT: PASS

[GT-Test] [-r] Test case: File with special characters in name
[GT-Test] CASE: ./bin/goattool -r test_files/special@file#.txt
[GT-Test] EXPECTED:
Permissions for 'test_files/special@file#.txt': 664
[GT-Test] ACTUAL:
Permissions for 'test_files/special@file#.txt': 664
[GT-Test] RESULT: PASS

[GT-Test] [-r] Test case: Executable file
[GT-Test] CASE: ./bin/goattool -r test_files/empty_file.txt
[GT-Test] EXPECTED:
Permissions for 'test_files/empty_file.txt': 777
[GT-Test] ACTUAL:
Permissions for 'test_files/empty_file.txt': 777
[GT-Test] RESULT: PASS

[GT-Test] [-v] Test case: Locked source file
[GT-Test] CASE: ./bin/goattool -v test_files/locked_file.txt
test_files/destination_file.txt
[GT-Test] EXPECTED:
Error: Cannot access 'test_files/locked_file.txt' due to
insufficient permissions.
[GT-Test] ACTUAL:
Error: Cannot access 'test_files/locked_file.txt' due to
insufficient permissions.
[GT-Test] RESULT: PASS

[GT-Test] [-v] Test case: Successful file move/rename
[GT-Test] CASE: ./bin/goattool -v test_files/merged.txt
test_files/destination_file.txt && ./bin/goattool -p test_files/
destination_file.txt
[GT-Test] EXPECTED:
File moved/renamed successfully.
--- Content of 'test_files/destination_file.txt' ---
This is the second file.
Merged content goes below the first file's content.
[GT-Test] ACTUAL:
File moved/renamed successfully.

--- Content of 'test_files/destination_file.txt' ---

This is the second file.

Merged content goes below the first file's content.

[GT-Test] RESULT: PASS

[GT-Test] [-v] Test case: No source or destination provided

[GT-Test] CASE: ./bin/goattool -v

[GT-Test] EXPECTED:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] ACTUAL:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] RESULT: PASS

[GT-Test] [-v] Test case: Non-existent source file

[GT-Test] CASE: ./bin/goattool -v test_files/
non_existent_file.txt test_files/destination_file.txt

[GT-Test] EXPECTED:

Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files/non_existent_file.txt' because it does not exist.

[GT-Test] RESULT: PASS

[GT-Test] [-v] Test case: No destination provided

[GT-Test] CASE: ./bin/goattool -v test_files/merged.txt

[GT-Test] EXPECTED:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] ACTUAL:

Error: Invalid command switch or missing arguments. Use -h for help.

[GT-Test] RESULT: PASS

[GT-Test] [-v] Test case: Source is a directory

[GT-Test] CASE: ./bin/goattool -v test_files test_files/
destination_file.txt

[GT-Test] EXPECTED:

Error: Cannot access 'test_files' because it is a directory.

[GT-Test] ACTUAL:

Error: Cannot access 'test_files' because it is a directory.

[GT-Test] RESULT: PASS

```
[GT-Test] [-v] Test case: Destination already exists
[GT-Test] CASE: ./bin/goattool -v test_files/
example_file.txt.goat test_files/destination_file.txt && ./bin/
goattool -p test_files/destination_file.txt
[GT-Test] EXPECTED:
File moved/renamed successfully.
--- Content of 'test_files/destination_file.txt' ---
T1h1i1s1 i1s1 a1n1 e1x1a1m1p1l1e1 f1i1l1e1.
I1t1 h1a1s1 m1u1l1t1i1p1l1e1 l1i1n1e1s1.
A1n1d1 i1t1 i1s1 u1s1e1d1 f1o1r1 t1e1s1t1i1n1g1 p1u1r1p1o1s1e1s1.
M1a1k1e1 s1u1r1e1 y1o1u1 h1a1v1e1 f1u1n1 t1e1s1t1i1n1g1!
T1h1i1s1 i1s1 a1 l1i1n1e1 w1i1t1h1 s1e1a1r1c1h1_t1e1r1m1.
a18b4 c3d2e1
[GT-Test] ACTUAL:
File moved/renamed successfully.
--- Content of 'test_files/destination_file.txt' ---
T1h1i1s1 i1s1 a1n1 e1x1a1m1p1l1e1 f1i1l1e1.
I1t1 h1a1s1 m1u1l1t1i1p1l1e1 l1i1n1e1s1.
A1n1d1 i1t1 i1s1 u1s1e1d1 f1o1r1 t1e1s1t1i1n1g1 p1u1r1p1o1s1e1s1.
M1a1k1e1 s1u1r1e1 y1o1u1 h1a1v1e1 f1u1n1 t1e1s1t1i1n1g1!
T1h1i1s1 i1s1 a1 l1i1n1e1 w1i1t1h1 s1e1a1r1c1h1_t1e1r1m1.
a18b4 c3d2e1
[GT-Test] RESULT: PASS
```

```
[GT-Test] [] Test case: No arguments provided
[GT-Test] CASE: ./bin/goattool
[GT-Test] EXPECTED:
Error: Insuffecient argument count. Use -h for help.
[GT-Test] ACTUAL:
Error: Insuffecient argument count. Use -h for help.
[GT-Test] RESULT: PASS
```

```
[GT-Test] [-e] Test case: Invalid switch provided
[GT-Test] CASE: ./bin/goattool
[GT-Test] EXPECTED:
Error: Insuffecient argument count. Use -h for help.
[GT-Test] ACTUAL:
Error: Insuffecient argument count. Use -h for help.
[GT-Test] RESULT: PASS
```

```
[GT-Test] [] Test case: No switch provided but arguments exist
[GT-Test] CASE: ./bin/goattool test_files/file1.txt test_files/
file2.txt
```



```
[GT-Test] EXPECTED:
Error: Invalid command switch or missing arguments. Use -h for
help.
[GT-Test] ACTUAL:
Error: Invalid command switch or missing arguments. Use -h for
help.
[GT-Test] RESULT: PASS

[GT-Test] Clearing test environment
[GT-Test] Test environment cleared
[GT-Test] END

[GT-Test] TEST SUMMARY
[GT-Test] 60 tests total
[GT-Test] 60 tests passed
[GT-Test] 0 tests failed
```

Privacy Disclosure

Welcome to GoatTool. We genuinely appreciate your trust and are firmly committed to preserving the privacy and security of your data.

Our Commitment:

GoatTool, while being a versatile command-line utility, is designed with the utmost care to respect your privacy. It operates primarily in your local environment, ensuring that your data stays where it belongs: with you.

Data We Access:

1. **Files and Directories:** GoatTool may access your files and directories to perform its core functionalities. This is solely dependent on the commands you execute.
2. **Search Queries:** Queries you use for searching within files are processed, but not stored or transmitted.
3. **Encryption Passwords:** If you employ encryption features, the passwords are utilized for that specific purpose alone and are neither stored nor transmitted elsewhere.

What We Do Not Do:

- GoatTool will never transmit or share your data with external servers or third parties.
- We do not integrate with or use third-party services that might jeopardize your privacy.
- Data is not sent over networks unless you explicitly command the tool to do so.

Your Responsibility:

We urge you to exercise caution when providing sensitive information or credentials, especially while scripting or performing automated tasks with GoatTool. Always ensure you are only feeding the necessary and appropriate data for your tasks.

Conclusion:

Your trust means everything to us. We want to ensure that while using GoatTool, you can focus on the tasks at hand without having to worry about privacy concerns. By using GoatTool, you acknowledge that you understand and consent to the practices detailed in this Privacy Disclosure.

Thank you for choosing GoatTool.