# fescape

Filter unprintable characters from input stream.

## Design Considerations

Short and simple. Follow the original Unix philosophy of a program doing just one thing where both input and output are streamed and are therefore able to function well in a pipe sequence.

## File Locations

- Binary executable (`fescape`) located in `/usr/local/bin`
- Manpage (`fescape.1`) located in `/usr/local/share/man/man1`
- API documentation located in project folder as `fescape-apidoc.pdf` and `fescape-apidoc.html`. Doc source files in `docs` folder.

## Arguments

- **file** : file(s) to display or stdin if none specified

## Switches

- `-r, --repeats` : display a count for repeated characters instead of displaying the characters themselves
- `-h, --help` : display usage
- `-n, --newline` : don't filter newlines
- `-o, --octal` : display octal codes instead of default hex codes

## Key Functions

`main()`

- Handle switches
- Handle arguments
- Handle actions

```
void usage(const char *program)
```

- Display help to user

```
fescape(FILE *input_stream, FILE *output_stream, bool repeats,
bool octal)
```

- Performs the filter function.

### Dependencies

- system-actions.{c,h}

- actions.{c,h}

## Results

Haven't discovered any issues yet.

## Lessons

1. **##___VA_ARGS___**: GNU macro feature useful for variadic functions.
2. **VS Code**: By default, VSC doesn't play nice with make(1) when debugging from within VSC. The VSC extension "Makefile Tools" is needed.

## TODO

☒ Figure out best way to handle variadic function fronted by macro.
☒ Optimize functions in system-actions.c.