

MEL Editor - User Manual

Table of Contents

1. Introduction
 2. Main Features
 3. Supported Platforms
 4. Supported Languages & Syntax Highlighting
 5. Installation Guide
 6. Using MEL Editor
 7. Command-Line Options
 8. Multiple CLI Options
 9. Content Loading Methods
 10. Keybindings
 11. Working with Resource-Constrained Environments
 12. Backup Mechanism
 13. Integration with Large Language Models (LLMs)
 14. Uninstallation
 15. FAQ and Troubleshooting
 16. Roadmap for Future Development
 17. License and Contact Information
-

1. Introduction

MEL (Mini Embedded Light) Editor is a **terminal-based, minimalistic text editor** for Linux and Unix platforms, written in **pure C** from scratch. It is designed for **low-resource environments**, such as:

- **Kubernetes Pods**
- **Docker Containers**
- **IoT Devices (ARM64, Embedded C)**

and

- **Cloud Platforms (AWS, GCP, OCI, etc.)**

MEL provides **syntax highlighting**, efficient **keyboard shortcuts**, and a dependency-free approach (it does not use `curses` or any external libraries).

MEL is **open-source** and distributed under the **GNU GPL3** license.

2. Main Features

- **Lightweight & Fast:** No external dependencies, runs efficiently even in constrained environments.
 - **Cross-Platform Support:** Works on Linux, macOS, Raspberry Pi OS, and embedded systems.
 - **Syntax Highlighting:** Supports multiple languages and file formats.
 - **Efficient Keyboard Shortcuts:** Allows fast and intuitive text editing.
 - **Minimalist Design:** No GUI, optimized for CLI workflows.
 - **Backup Mechanism:** Automatically creates backups of modified files upon the **first** save operation.
 - **LLM Integration:** Supports various **Large Language Models** through an **original** framework.
-

3. Supported Platforms

MEL has been tested on:

- **Linux Distributions:** Ubuntu, Debian, RedHat EL, Fedora, CentOS, Linux Mint, Oracle Linux, etc.
 - **macOS** (Intel & Apple Silicon/M1 chips).
 - **Raspberry Pi OS.**
 - **Embedded & Cloud Environments:** AWS, Kubernetes, Docker, IoT.
-

4. Supported Languages & Syntax Highlighting

MEL supports syntax highlighting for the following languages:

Language	File Extensions
C	.c, .h
C++	.cpp, .hpp, .cc, .cxx
Java	.java
Python	.py, .pyc, .pyo, .pyw
PHP	.php, .phtml
Bash	.sh
Mshell	.ms
JavaScript	.js, .jsx
JSON	.json, .jsonp

Language	File Extensions
XML	.xml (partial support)
SQL	.sql
Ruby	.rb
Go	.go

Custom syntax formats can be added upon request.

5. Installation Guide

Compiling MEL from Source

To build MEL from source, use the following commands:

```
git clone https://github.com/igor101964/mel.git
cd mel/
make
```

Downloading and Installing MEL

If you prefer the pre-built binary, you can use files from the **binaries directory** from [MEL's GitHub repository](https://github.com/igor101964/mel) (<https://github.com/igor101964/mel>)

It was tested on main **Linux OS platforms, macOS, and Raspberry Pi OS** using:

- **mel_mac_os** (for macOS)
 - **mel_raspberrypi_os** (for Raspberry Pi OS)
 - **mel_ubuntu_2x.0x** (for Ubuntu 2x.0x and similar Linux distributions)
-

6. Using MEL Editor

To open an existing file in MEL:

```
mel <file_name>
```

To start MEL **without opening a file**:

```
mel
```

This will open an empty editor where you can **manually create content** and later save it as a file.

7. Command-Line Options

MEL provides various command-line options for enhanced usability:

Command	Description
<code>mel -h</code> or <code>mel --help</code>	Displays help options
<code>mel -v</code> or <code>mel --version</code>	Shows the current version
<code>mel -b <file></code>	Creates a backup of the file (.bak) upon first save
<code>mel -l <line_number> <file></code>	Opens a file at a specific line number
<code>mel -w <width_marker_number> <file></code>	Sets a width marker for formatting

8. Multiple CLI Options

MEL allows using multiple options together:

```
mel -b -l 10 myfile.txt
```

This will:

1. Create a backup of `myfile.txt` upon the first save.
 2. Open it at **line 10**.
-

9. Content Loading Methods

9.1 Opening a File

MEL does not create new files automatically. However, you can start MEL and save your work manually.

9.2 Piping Data to MEL

```
cat myfile.txt | mel
```

9.3 Executing a Sequence of Commands and Sending Output to MEL

To execute multiple commands and pass the output to MEL, use **parentheses**:

```
(df -m ; ls -la; | inxi) | mel
```

9.4 Using `grep` to Filter Output Before Sending to MEL

```
(df -m | grep /boot/efi) | mel
```

This will:

1. Execute `df -m` (disk space usage).
2. Filter only lines containing `/boot/efi`.
3. Send the result to MEL.

Any commands that support programming channels are ready to use.

10. Keybindings

Keybinding	Action
Ctrl-Q	Exit, Exit, 3 times click Ctrl-Q if file was changed without saving
Ctrl-S	Save, requires input of file name, if file didn't exist
Ctrl-F	Search by pattern, Esc - exit from Search, Enter and Arrows to interact searching
Ctrl-N	Forward Search by pattern after Ctrl-F. Esc - exit from Search, works after Ctrl-F only
Ctrl-R	Backward Search by pattern after Ctrl-F. Esc - exit from Search, works after Ctrl-F only
Ctrl-J	Global replacement of \u0441haracter combinations, Input Search and Replace patterns, Esc to cancel, Enter to input
Ctrl-G	Go to line Number, requires input the line number
Ctrl-B	Hide/Show line numbering
Ctrl-E	Flip line upwards
Ctrl-D	Flip line downwards
Ctrl-C	Copy line
Ctrl-X	Cut line
Ctrl-V	Paste line
Ctrl-Z	Undo
Ctrl-Y	Redo
Ctrl-P	Pause mel (type "fg" to resume)
Ctrl-H	Toggle help screen
Home	Move the cursor to the beginning of the line
End	Move cursor to end of line
PgUp	Up page scroll
PgDn	Down page scroll

Keybinding	Action
Up	Move cursor up one position
Down	Move cursor down one position
Left	Move cursor left one position
Right	Move cursor right one position
Backspace	Delete character

12. Backup Mechanism

MEL automatically creates backups when the **first save operation occurs**.

```
mel -b myfile.txt
```

The backup file will be saved as `myfile.txt.bak`.

13. Integration with Large Language Models (LLMs)

MEL **supports various LLM models** through an **original framework**, allowing integration with AI-powered solutions for text processing and automation.

14. Uninstallation

```
sudo rm /usr/local/bin/mel
sudo rm /bin/mel
sudo rm /usr/bin/mel
```

Supports highlighting for

C,C++,Java,Bash,Mshell,Python,PHP,Javascript,JSON,XML,SQL,Ruby,Go

License: Public domain libre software GPL3,v.0.2.0, 2025

Coding: Igor Lukyanov, igor.lukyanov@appservgrid.com

For now, usage of UTF-8 is recommended.