

ROSE SWE Software Engineering

---

# ROSE SWE

<http://rose.rult.at>

<http://www.cfg2html.com>

---

## Enduser Program Documentation

This document provides end-user guidance for ROSE SWE software and services. It covers intended use, configuration, and operational behavior, and is intended for readers who administer or use the software in production environments.

The information in this document is provided “as is” and is subject to change without notice. ROSE SWE assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein or the usage of the program itself.

Last Updated: 21-Oct-2026

# Table of Contents

|                                                |   |
|------------------------------------------------|---|
| RMS_NG .....                                   | 3 |
| 1. Overview .....                              | 3 |
| 1.1. At a Glance .....                         | 3 |
| 1.2. Design Goals .....                        | 3 |
| 1.3. Status .....                              | 3 |
| 2. Quick start .....                           | 4 |
| 3. Command reference .....                     | 4 |
| 3.1. Example output .....                      | 5 |
| 4. Match types .....                           | 5 |
| 5. Quarantine .....                            | 5 |
| 5.1. Example .....                             | 5 |
| 6. Log files .....                             | 6 |
| 7. Exit codes .....                            | 6 |
| 8. Platform notes .....                        | 6 |
| 8.1. Linux .....                               | 6 |
| 8.2. Windows .....                             | 7 |
| 8.3. macOS (not yet provided!) .....           | 7 |
| 9. Troubleshooting .....                       | 8 |
| 9.1. "Database : ... not found" .....          | 8 |
| 9.2. "Error: magic line mismatch" .....        | 8 |
| 9.3. Scan appears to hang .....                | 8 |
| 9.4. Many "Variant" hits .....                 | 8 |
| 9.5. High thread counts slow things down ..... | 8 |
| 9.6. Terminal output is garbled .....          | 8 |
| 10. History .....                              | 8 |

# Table of Contents

---

## RMS\_NG

### 1. Overview

#### NOTE

RMS\_NG ROSE\_SWE Malware Scanner Next Generation — an ultra-fast, multi-platform, multithreaded virus and malware file signature scanner built for massive file collections.

RMS\_NG recursively scans a directory tree, compares every file against a signature database ( `virscan.mwdb` ), and reports, quarantines, or deletes files that match a known signature. It is designed for speed-first scanning of large archives, backups, research datasets, NAS volumes, and malware collections.

RMS\_NG was completely rewritten from the ground up using the newest technologies, leveraging multicore 64-bit CPUs and huge memory capacity to accelerate scanning. It also deliberately drops legacy technologies such as DOS and boot viruses to keep the design leaner and faster. RMS\_NG focuses on current threats: Windows, Android, macOS, and Linux malware, ransomware, Trojans, and rootkits.

For DOS, boot, and script viruses, use our classic **RMS** scanner. After a successful testing and concept phase, the intention is to merge **RMS** and **RMS\_NG** into one unified product.

#### 1.1. At a Glance

- Ultra-fast, multithreaded scanning
- 64-bit and multicore optimized
- Signature-based detection using `virscan.mwdb`
- Recursive directory-tree scanning
- Report, quarantine, or delete matched files
- Focused on modern malware: Windows, Android, macOS, and Linux
- Targets malware, ransomware, Trojans, and rootkits
- Legacy virus coverage handled by RMS
- Future roadmap: unified RMS + RMS\_NG product

#### 1.2. Design Goals

- Speed over bloat
- Modern threats over legacy targets
- Multicore scalability
- High memory utilization for faster matching
- Cleaner, leaner architecture
- Built to chew through very large file collections

#### 1.3. Status

Proof-of-concept. Early development. Bugs and defects are still possible. Not yet intended as a production security product.

 **CAUTION**

 **NOTICE:** This is a proof-of-concept in an early development stage. It may still contain bugs and other defects.  Additional features will also be added.

## 2. Quick start

You need two files in the same directory:

- `rms_ng` (or `rms_ng64.exe` on Windows)
- `virscan.mwdb` (the signature database)

```
# Report only (default)
rms_ng /path/to/scan

# Report and write a log file
rms_ng -log /path/to/scan

# Move all exact matches into quarantine
rms_ng -move /var/quarantine /path/to/scan

# Also act on variants
rms_ng -heur -move /var/quarantine /path/to/scan
```

---

## 3. Command reference

Usage: `rms_ng` [options] <directory>

Options:

|                                 |                                                           |
|---------------------------------|-----------------------------------------------------------|
| <code>-log</code>               | Print found files only; also writes a log file            |
| <code>-report</code>            | Print every file; no log file                             |
| <code>-heur</code>              | Show and act on variants → false positives                |
| <code>-maxheur</code>           | Maximum heuristics on → produces a lot of false positives |
| <code>-err</code>               | Show file/dir read/access errors                          |
| <code>-threads &lt;N&gt;</code> | Number of worker threads (default: auto → all cores)      |
| <code>-del</code>               | Delete exact matches                                      |
| <code>-copy &lt;path&gt;</code> | Copy matches into quarantine <path>                       |
| <code>-move &lt;path&gt;</code> | Move matches into quarantine <path>                       |
| <code>-?</code>                 | Show help                                                 |

| Option                          | Description                                                                                                        |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <i>(none)</i>                   | Print found files only.                                                                                            |
| <code>-log</code>               | Print found files and write a timestamped log file in the current directory.                                       |
| <code>-report</code>            | Print <b>every</b> scanned file, marked <code>OK</code> or with the match text. Slower output; no log file.        |
| <code>-heur</code>              | Enable heuristic mode. Variants are shown and acted on. Without this flag, variants are silently ignored.          |
| <code>-err</code>               | Show read/access errors that would otherwise be silently counted.                                                  |
| <code>-threads &lt;N&gt;</code> | Number of worker threads. Default: number of logical CPUs. Maximum: 64.                                            |
| <code>-del</code>               | Delete exact matches. Combined with <code>-heur</code> , also deletes variants. 🙌                                  |
| <code>-copy &lt;path&gt;</code> | Copy matches into quarantine <code>&lt;path&gt;</code> . Combined with <code>-heur</code> , also copies variants.  |
| <code>-move &lt;path&gt;</code> | Move matches into quarantine <code>&lt;path&gt;</code> . Combined with <code>-heur</code> , also moves variants. 🙌 |



NOTE

`-del` , `-copy` and `-move` are **mutually exclusive** — pick one.

3.1. Example output

```
Database : /home/user/bin/virscan.mwdb
Created  : 2026-10-06T15:50:46
Entries  : 374.559
Scanning : /home/user/drv
Threads  : 24

/home/user/drv/worm1.exe : Worm.1234.458752
/home/user/drv/worm2.exe : Worm.1234.458752
/home/user/drv/suspect.bin : Variant of Worm.9999.12345678

--- Statistics ---
Files scanned      : 40.640
Bytes scanned     : 54632.70 MB
Exact matches     : 40.639
Variants          : 1
Total found       : 40.640
Elapsed time      : 00:03:11
```

While scanning, a **status line** at the bottom of the terminal shows the file currently being processed. On non-interactive terminals (pipes, redirected output, `TERM=dumb` ) this line is suppressed automatically so logs stay clean.

4. Match types

| Result             | Meaning                                                                         | Action                                                                                       |
|--------------------|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| <b>Exact match</b> | The file is identical to a known entry in the database.                         | Reported. Acted on with <code>-del</code> , <code>-copy</code> , or <code>-move</code> .     |
| <b>Variant</b>     | The file resembles a known entry but is not identical. Usually a modified copy. | Hidden by default. Reported and acted on only with <code>-heur</code> .                      |
| <b>No match</b>    | The file is not known.                                                          | Silently skipped ( <code>-log</code> ) or shown as <code>OK</code> ( <code>-report</code> ). |

Variants are never acted on unless you explicitly enable `-heur` — this prevents accidental deletion of legitimate files.

5. Quarantine

`-copy` and `-move` place files in a **flat** directory with the database name prepended. The original directory structure is not preserved.

| Situation                      | Target filename                                                                           |
|--------------------------------|-------------------------------------------------------------------------------------------|
| Exact match                    | <code>&lt;DB-name&gt;.&lt;original&gt;</code>                                             |
| Variant ( <code>-heur</code> ) | <code>&lt;DB-name&gt;.Variant.&lt;original&gt;</code>                                     |
| Name collision                 | <code>&lt;base&gt;_1.&lt;text&gt;</code> , <code>&lt;base&gt;_2.&lt;text&gt;</code> , ... |

5.1. Example

```
Original file   : /home/user/drv/sub/worm.exe
Database entry  : Worm.1234
```

```
Quarantine folder : /var/quarantine
Result             : /var/quarantine/Worm.1234.worm.exe
```

If `Worm.1234.worm.exe` already exists, the next file is stored as `Worm.1234.worm_1.exe`, then `Worm.1234.worm_2.exe`, and so on.

---

## 6. Log files

With `-log`, RMS\_NG writes a timestamped log file in the current directory:

```
rms_ng_20261006_153012.log
```

Format: `rms_ng_<yyyymmdd>_<hhnnss>.log`.

The log contains the same content as the console output (header, hits, statistics). The status line is not written to the log.

---

## 7. Exit codes

| Code | Meaning                                                                                    |
|------|--------------------------------------------------------------------------------------------|
| 0    | Scan completed successfully.                                                               |
| 1    | Fatal error — database missing/corrupt, invalid arguments, or inaccessible scan directory. |

The exit code does **not** indicate whether matches were found. Parse the statistics or the log file for that.

---

## 8. Platform notes

RMS\_NG runs on 64-bit Linux, Windows, and macOS. Multicore CPUs and non-mechanical hard disk highly recommended!

### WARNING

**Memory requirement:** Rule of thumb:

- Linux: `virscan.mwdb` size \* 15, so with a 90 MB database you should have around 1.35 GB of free memory available, which makes the program unsuitable for e.g. small SOC systems.
- Windows: `virscan.mwdb` size \* 11, so with a 98 MB database you should have around 1.04 GB of free memory available

### 8.1. Linux

**Requirements:** glibc 2.34 or newer (Ubuntu 22.04, Debian 12, Fedora 35+, RHEL 9+, SLES/openSUSE 15.6) for threading support. Glibc 2.34 was released on August 1, 2021, making it about 5 years old as of late 2026. Since newer versions of glibc are backward-compatible, any distribution that shipped with version 2.34 or later will support software compiled against it.

```
$ ldd -v rms_ng
linux-vdso.so.1 (0x00007188551bd000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x0000718854e00000)
/lib64/ld-linux-x86-64.so.2 (0x00007188551bf000)

Version information:
/home/ralph/exe/rms_ng:
    libc.so.6 (GLIBC_2.34) => /lib/x86_64-linux-gnu/libc.so.6
    libc.so.6 (GLIBC_2.2.5) => /lib/x86_64-linux-gnu/libc.so.6
    /lib/x86_64-linux-gnu/libc.so.6:
        ld-linux-x86-64.so.2 (GLIBC_2.2.5) => /lib64/ld-linux-x86-64.so.2
        ld-linux-x86-64.so.2 (GLIBC_2.3) => /lib64/ld-linux-x86-64.so.2
        ld-linux-x86-64.so.2 (GLIBC_PRIVATE) => /lib64/ld-linux-x86-64.so.2
```

### Running:

```
./rms_ng /path/to/scan
./rms_ng -threads 8 /path/to/scan
./rms_ng -heur -move /var/quarantine /path/to/scan
```

**Thread count:** default is the number of logical CPUs ( `nproc` ). Override with `-threads N` .

**Permissions:** to use `-del` or `-move` , the user must have write access to the files being scanned and to the quarantine directory.

## 8.2. Windows

**Requirements:** Windows 10 or 11, 64-bit.

**Running (Command Prompt or PowerShell):**

```
rms_ng.exe C:\Data
rms_ng.exe -threads 8 C:\Data
rms_ng.exe -heur -move D:\Quarantine C:\Data
```

### Notes:

- Paths with spaces must be quoted: `rms_ng.exe "C:\My Data"` .
- Long paths (> 260 characters) require the Windows long-path opt-in or a short path prefix (`\\?\` ).
- To use `-del` or `-move` on system directories, run as Administrator.
- If you want to scan the C: drive use "C:\\" (with backslash)

## 8.3. macOS (not yet provided!)

**Requirements:** macOS 12 (Monterey) or newer, Intel or Apple Silicon.

### Running:

```
./rms_ng ~/Documents
./rms_ng -threads $(sysctl -n hw.ncpu) ~/Documents
./rms_ng -heur -move ~/Quarantine ~/Downloads
```

### Notes:

- On first run, macOS may show a security warning. Allow the binary under **System Settings** → **Privacy & Security**.
- If the binary was downloaded from the internet, clear the quarantine flag manually:

```
xattr -d com.apple.quarantine ./rms_ng
```

- `-threads` defaults to 4 on macOS; for best performance, specify it explicitly using `$(sysctl -n hw.ncpu)`.

---

## 9. Troubleshooting

### 9.1. "Database : ... not found"

RMS\_NG looks for `virscan.mwdb` in this order:

1. The current working directory
2. The directory containing the `rms_ng` binary
3. Any directory in `$PATH` / `%PATH%`

Place `virscan.mwdb` next to the binary or run RMS\_NG from the directory that contains it.

### 9.2. "Error: magic line mismatch"

The database was created with a different version, or it is corrupt. Re-download `virscan.mwdb` from a trusted source.

### 9.3. Scan appears to hang

RMS\_NG is likely stuck on a slow network mount, a device with I/O errors, or an enormous single file. Check the status line — it shows the file currently being processed. Use `-err` to see read errors.

### 9.4. Many "Variant" hits

If `-heur` produces a large number of variant reports, the database was built with a lenient signature threshold. This is expected for some environments; check whether the variants are genuinely suspicious before acting on them.

### 9.5. High thread counts slow things down

On spinning disks and network mounts, `-threads 1` or `-threads 2` often performs better than higher counts. On NVMe SSDs, use the default (all CPUs).

### 9.6. Terminal output is garbled

The status line uses terminal control sequences. If you see raw escape codes, your terminal does not support them. RMS\_NG detects this and disables the status line automatically when `TERM=dumb` (Linux) or when output is redirected to a file.

## 10. History

```
1.3.2 - 21.10.2026 - small enhancements. First release as a standalone program
1.2   - 18.10.2026 - Option -maxheur, ESC should stop a running scan
1.1   - 17.10.2026 - Added key file (rms_ng.key)
1.0   - 01.10.2026 - Initial public release
```