

## Linux Mint: SHA256-Prüfdateien per Rechtsklick in Nemo erstellen & verifizieren

by <https://rbconnect.eu> Kontakt: [mail@rbconnect.eu](mailto:mail@rbconnect.eu) Stand: 10/2026

Wer große Dateimengen kopiert, Backups anlegt oder Downloads absichert, muss sich auf die Integrität der Daten verlassen können. Standardmäßig erfordert das Berechnen und Abgleichen von SHA256-Prüfsummen unter Linux oft den Umweg über das Terminal. In dieser Anleitung richten wir zwei maßgeschneiderte Nemo Actions ein. Damit erweiterst du das Kontextmenü deines Dateimanagers um zwei mächtige Funktionen:

1. Prüfdatei erstellen: Berechnet für beliebig viele markierte Dateien die Hashes und speichert sie ohne störende Pfadangaben.
2. Prüfdatei verifizieren: Gleichet die .sha256-Datei automatisch ab und warnt dich, falls Daten manipuliert wurden oder fehlen.

Beide Funktionen bieten einen grafischen Fortschrittsbalken mit geschätzter Restzeit und schließen Erfolgsmeldungen nach 10 Sekunden automatisch.

### Teil 1: Die Skripte anlegen

Nemo Actions rufen im Hintergrund Shell-Skripte auf. Diese speichern wir im lokalen Verzeichnis deines Benutzers.

#### Skript 1: Prüfdatei erstellen (checksum\_create.sh)

Öffne ein Terminal und erstelle die Datei mit folgendem Befehl:

```
nano ~/.local/share/nemo/actions/checksum_create.sh
```

Kopiere diesen vollständigen Code hinein:

```
#!/bin/bash
```

```
TARGET_DIR="$1"  
shift
```

```
cd "$TARGET_DIR" || exit 1  
rm -f "checksums.sha256"
```

```
# 1. Nur echte Dateien aus den Argumenten filtern und zählen  
VALID_FILES=(  
for arg in "$@"; do  
if [ -f "$arg" ]; then  
VALID_FILES+=("${basename "$arg}")"  
fi  
done
```

```
TOTAL_FILES=${#VALID_FILES[@]}
```

```
# Fehler ausgeben, falls keine Dateien markiert wurden  
if [ "$TOTAL_FILES" -eq 0 ]; then  
zenity --error --title="Fehler" --text="Es wurden keine gültigen  
Dateien für die Prüfsumme gefunden."  
exit 1  
fi
```

```

# 2. Die Berechnung starten und den Fortschrittsbalken füttern
count=0
START_TIME=$(date +%s)

(
for filename in "${VALID_FILES[@]}"; do
sha256sum "$filename" >> "checksums.sha256"

count=$((count + 1))
PERCENT=$((count * 100 / TOTAL_FILES))

# Restzeit berechnen
CURRENT_TIME=$(date +%s)
ELAPSED=$((CURRENT_TIME - START_TIME))

if [ $count -gt 0 ] && [ $ELAPSED -gt 0 ]; then
REMAINING_FILES=$((TOTAL_FILES - count))
TIME_PER_FILE_X100=$((ELAPSED * 100 / count))
ESTIMATED_REMAINING=$((REMAINING_FILES * TIME_PER_FILE_X100 / 100))
TIME_TEXT=" (Noch ca. ${ESTIMATED_REMAINING} Sek.)"
else
TIME_TEXT=" (Berechne Restzeit...)"
fi

if [ $count -eq $TOTAL_FILES ]; then
TIME_TEXT=""
fi

echo "# Datei $count von $TOTAL_FILES: $filename$TIME_TEXT"
echo "$PERCENT"
done
) | zenity --progress --title="Prüfsummen werden berechnet" --
text="Berechnung startet..." --percentage=0 --auto-close --no-cancel

# 3. Das finale Erfolgs-Pop-up
zenity --info --title="Prüfdatei fertig" --text="Die Datei
'checksums.sha256' wurde erfolgreich für $TOTAL_FILES Datei(en)
erstellt." --timeout=10

```

## **Skript 2: Prüfdatei verifizieren (checksum\_verify.sh)**

Erstelle das zweite Skript für die Gegenprüfung:

```
nano ~/.local/share/nemo/actions/checksum_verify.sh
```

Füge diesen Code ein:

```

#!/bin/bash
TARGET_DIR="$1"
CHECKSUM_FILE="$2"
cd "$TARGET_DIR" || exit 1

```

```

if [ ! -f "$CHECKSUM_FILE" ]; then
zenity --error --title="Fehler" --text="Die ausgewählte Datei existiert
nicht."
exit 1
fi

TOTAL_FILES=$(grep -c ' ' "$CHECKSUM_FILE")

if [ "$TOTAL_FILES" -eq 0 ]; then
zenity --error --title="Fehler" --text="Keine gültigen SHA256-Einträge
in der Datei gefunden."
exit 1
fi

FAILED_LOG=$(mktemp)
MISSING_LOG=$(mktemp)

count=0
START_TIME=$(date +%s)

(
while read -r line; do
[ -z "$line" ] && continue

echo "$line" > "${CHECKSUM_FILE}.tmp"
filename=$(echo "$line" | awk '{print $2}' | sed 's/^\s*//')
count=$((count + 1))
PERCENT=$((count * 100 / TOTAL_FILES))

CURRENT_TIME=$(date +%s)
ELAPSED=$((CURRENT_TIME - START_TIME))
if [ $count -gt 1 ] && [ $ELAPSED -gt 0 ]; then
REMAINING_FILES=$((TOTAL_FILES - count))
TIME_PER_FILE_X100=$((ELAPSED * 100 / count))
ESTIMATED_REMAINING=$((REMAINING_FILES * TIME_PER_FILE_X100 / 100))
TIME_TEXT=" (Noch ca. ${ESTIMATED_REMAINING} Sek.)"
else
TIME_TEXT=" (Berechne Restzeit...)"
fi
[ $count -eq $TOTAL_FILES ] && TIME_TEXT=""

echo "# Prüfe Datei $count von $TOTAL_FILES: $filename$TIME_TEXT"
echo "$PERCENT"

if ! sha256sum --check "${CHECKSUM_FILE}.tmp" > /dev/null 2>&1; then
if [ ! -f "$filename" ]; then
echo "- $filename (Fehlt!)" >> "$MISSING_LOG"
else
echo "- $filename (Beschädigt/Geändert!)" >> "$FAILED_LOG"
fi
fi

```

```

rm -f "${CHECKSUM_FILE}.tmp"
done < "$CHECKSUM_FILE"
) | zenity --progress --title="Prüfsummen werden verifiziert" --
text="Prüfung startet..." --percentage=0 --auto-close --no-cancel

ERRORS=""
if [ -s "$FAILED_LOG" ]; then
ERRORS="$ERRORS\n\n<b>Defekte Dateien:</b>\n$(cat "$FAILED_LOG")"
fi
if [ -s "$MISSING_LOG" ]; then
ERRORS="$ERRORS\n\n<b>Fehlende Dateien:</b>\n$(cat "$MISSING_LOG")"
fi

rm -f "$FAILED_LOG" "$MISSING_LOG"

if [ -z "$ERRORS" ]; then
zenity --info --title="Prüfung erfolgreich" --text="Alle $TOTAL_FILES
Dateien sind intakt und fehlerfrei!" --timeout=10
else
zenity --error --title="Prüfung fehlgeschlagen!" --text="Achtung! Es
wurden Fehler gefunden:$ERRORS"
fi

```

### **Skripte ausführbar machen**

Damit das System die Skripte starten darf, musst du die Rechte im Terminal anpassen:

```

chmod +x ~/.local/share/nemo/actions/checksum_create.sh
chmod +x ~/.local/share/nemo/actions/checksum_verify.sh

```

### **Teil 2: Das Nemo-Kontextmenü erweitern**

Damit die Menüpunkte im Rechtsklick-Menü auftauchen, legen wir nun die .nemo\_action-Konfigurationsdateien an.

```

nano ~/.local/share/nemo/actions/checksum_create.nemo_action

```

#### **Inhalt einfügen:**

```

[Nemo Action]
Active=true
Name=Prüfdatei erstellen (SHA256)
Comment=Erstellt eine SHA256-Prüfdatei für alle markierten Elemente
Exec=<checksum_create.sh %P %F>
Selection=notnone
Extensions=any
Quote=double

```

## Aktion 2: Menüpunkt fürs Verifizieren

```
nano ~/.local/share/nemo/actions/checksum_verify.nemo_action
```

### Inhalt einfügen:

```
[Nemo Action]
Active=true
Name=Prüfdatei verifizieren (SHA256)
Comment=Überprüft alle in dieser Datei gelisteten Hashes
Exec=<checksum_verify.sh %P %F>
Selection=s
Extensions=sha256;SHA256;
Quote=double
```

## Teil 3: Aktivierung & Anwendung

Um die Konfiguration einzulesen, starte den Nemo-Dateimanager einmal im Terminal komplett neu mit: **nemo -q**

So nutzt du die neuen Funktionen:

- **Hashes generieren:** Markiere eine oder mehrere beliebige Dateien → *Rechtsklick* → „**Prüfdatei erstellen (SHA256)**“. Es erscheint ein Ladebalken mit Restzeit. Nach Abschluss liegt eine saubere checksums.sha256 im selben Ordner.
- **Hashes prüfen:** Mache einen *Rechtsklick* direkt auf eine .sha256-Datei → „**Prüfdatei verifizieren (SHA256)**“. Stellt das Skript fest, dass eine Datei manipuliert oder gelöscht wurde, listet ein rotes Fehler-Pop-up die betroffenen Dateien exakt auf.