

# Split Archive mit 7z

---

Version: 2024-04-24, [felix.blasinger@fhstp.ac.at](mailto:felix.blasinger@fhstp.ac.at)

## Verwendetes Archivformat: 7z

---

### Warum 7z?

7-Zip wird für seine effiziente und deutliche Dateikompression geschätzt.

Darüber hinaus:

- Durch die modulare Open-Source-Architektur kann jede:r Nutzer:in mit dem Format arbeiten.
- Unterstützung von sicherer AES-256-Verschlüsselung.
- Beinahe unbegrenzt große Dateien können verarbeitet werden.
- Dateinamen werden in Unicode gespeichert.
- Dateien werden in einem einzelnen Stream komprimiert, um Redundanzen zu vermeiden.
- Archive können aufgeteilt werden.

## Verwendete Tools

---

### Windows

- 7-Zip, OpenSource, Download: [7-Zip Website](#)

### MacOS

*7-Zip gibt es für MacOS leider nur in einer Version für die Command Line. 7z-Archive können mit dem in MacOS integrierten Archiv-Tool geöffnet werden, es können aber keine 7z-Archive damit erstellt werden. Es gibt aber alternative Tools:*

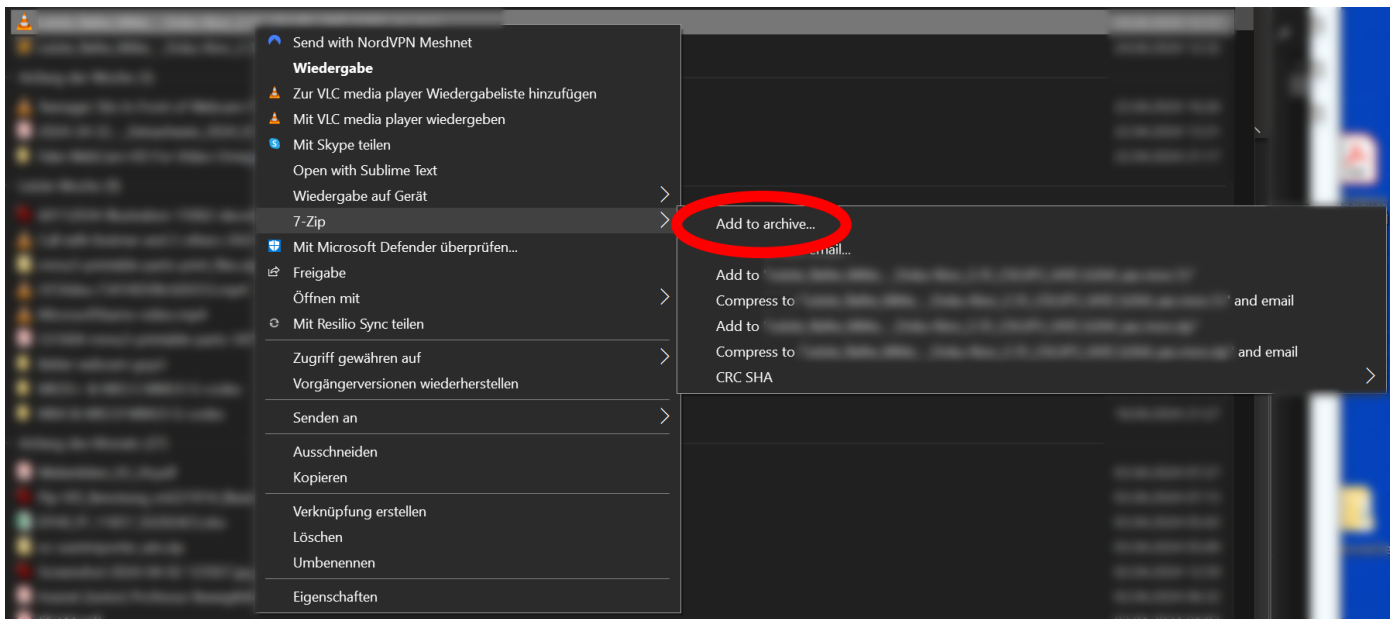
- Keka, OpenSource; Download: [Keka Website](#)

## Workflow: Archiv aufsplitten mit 7-Zip unter Windows

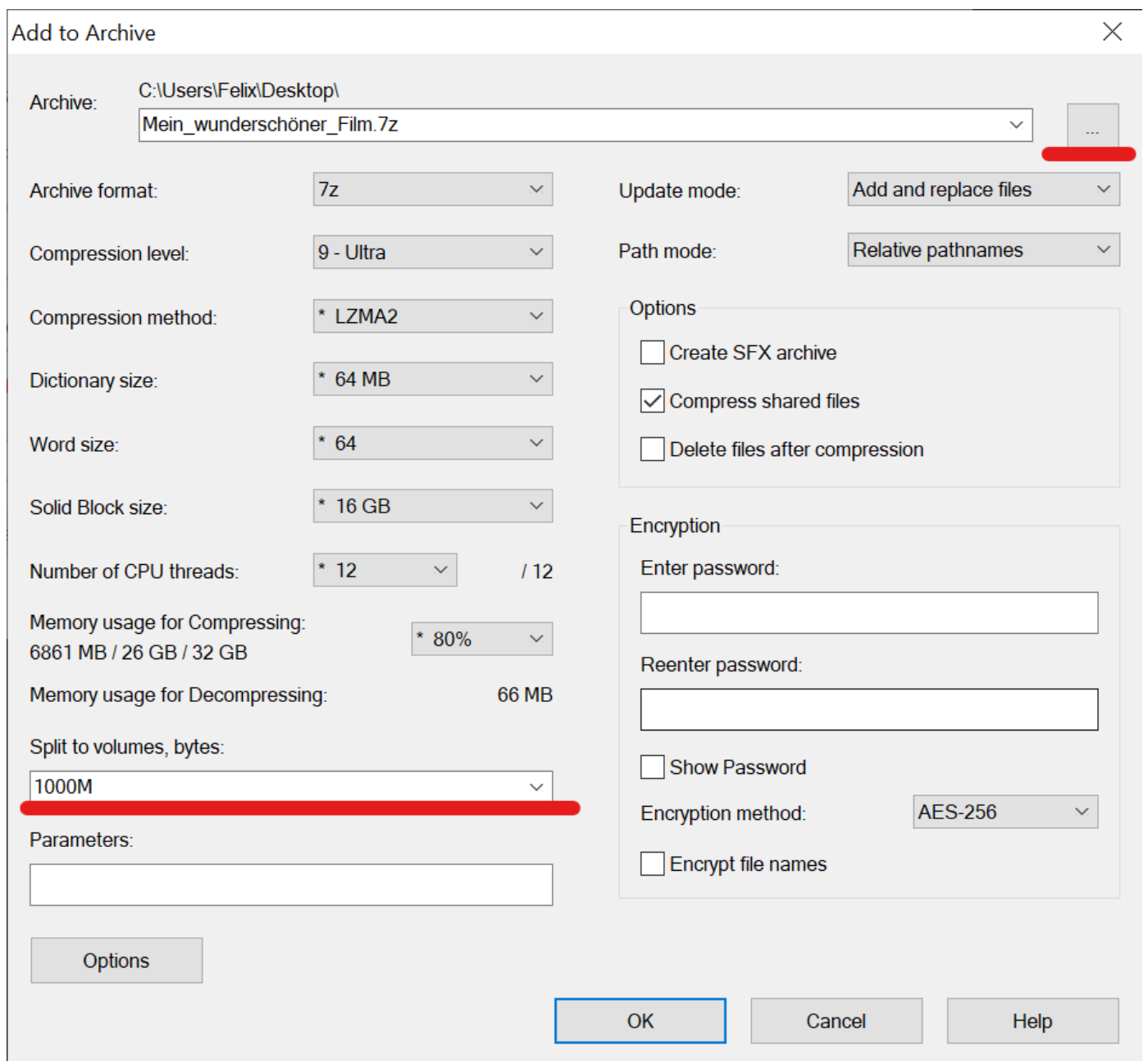
---

### Archiv erstellen

Nach der Installation ist 7-Zip mit einem Rechtsklick auf eine Dateiauswahl im Windows Explorer aufrufbar. Alternativ kann das Programm auch über das Startmenü gestartet werden.



Mit einem Klick auf `Add to archive...` öffnet sich ein Kontextmenü.

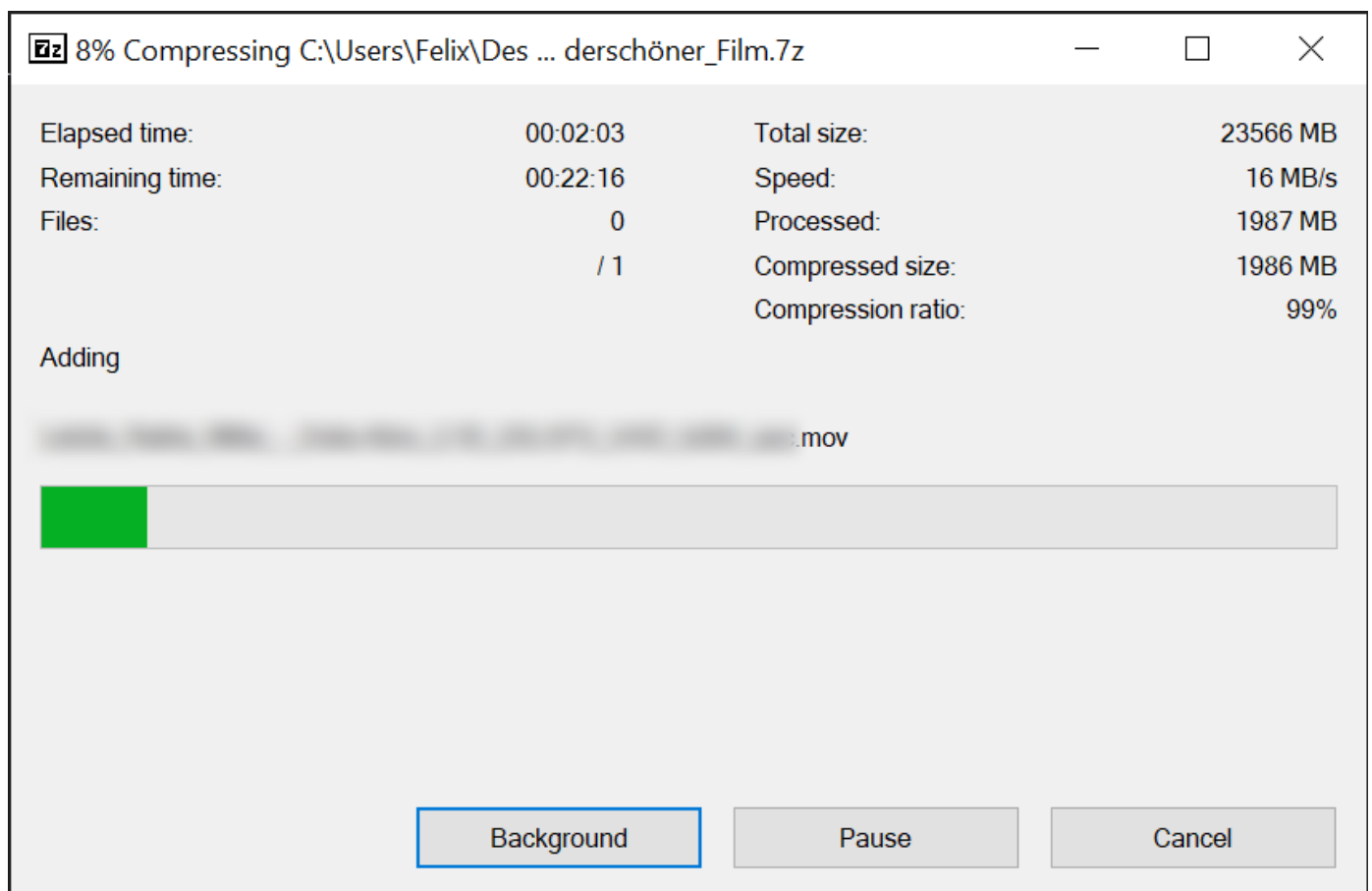


Mit dem Browse-Button [...] kann ein Zielpfad für das zu erstellende Archiv angegeben werden. Wichtig ist der Punkt `Split to volumes, bytes`. Dort muss angegeben werden, wie groß die einzelnen Archivteile werden sollen. Aus dem drop-down Menü können Voreinstellungen für klassische (Datenträger)größen ausgewählt werden. Es können aber auch alternative Werte eingegeben werden, wobei die Kürzel `M`, `G`, `T` (Mega-, Giga, Tera) (Beispielsweise `100M` -> 100MB Files) verwendet werden können.

Alle anderen Angaben können ohne Änderung übernommen werden. Gegebenenfalls kann das `Compression Level` angepasst werden, da dieses stark beeinflusst, wie lange der Archivierungsvorgang dauert. Ist bekannt, dass es sich um bereits stark komprimierte Dateien (Bsp: h264 etc.) handelt, kann das Level auf einen niedrigeren Wert gesetzt werden, da nicht zu erwarten ist, dass eine weitere wesentliche Reduktion der Dateigröße zu erwarten ist. Bei weniger stark komprimierten Dateien (Bsp. ProRes, DNxHD etc) kann ein hohes `Compression Level` durchaus zu Einsparungen in der Dateigröße führen, welche die längere Verarbeitungszeit durch eine kürzere Uploaddauer kompensieren.

Mit einem Klick auf `OK` startet der Archivierungsprozess. Je nach gewählter Kompressionsstärke und verfügbarer CPU-Leistung dauert der Prozess.

## Abwarten...



- Note: Die Komprimierung der Datei durch 7-Zip beeinflusst die Qualität nicht. Die später entpackte Datei entspricht exakt dem Original.

- Note: Ist jemand sehr in Zeitnot und will während des Archivierungsprozesses bereits mit dem Upload von einzelnen Teilen starten, kann er oder sie das machen. Dabei ist darauf zu achten, dass die entsprechenden Split-Files bereits vollständig geschrieben wurden (also wenn es schon ein `Archivname.7z.003` gibt, kann ich das `Archivname.7z.002` bereits hochladen). Bei `Archivname.7z.001` muss allerdings bis zum Ende des Archivierungsprozesses gewartet werden, da in dieser laufend Metadaten aktualisiert werden.

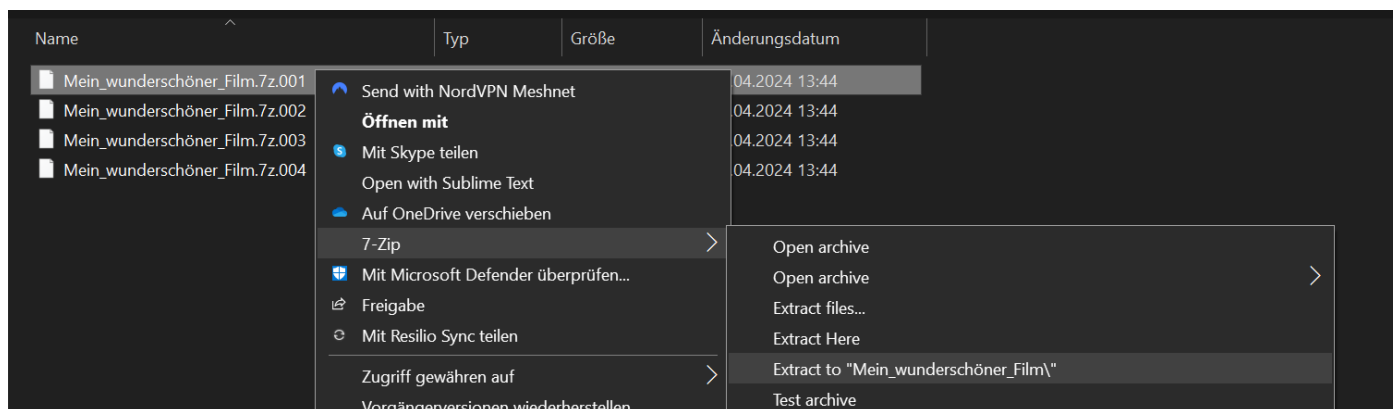
## Ergebnis

Name	Typ	Größe
Mein_wunderschöner_Film.7z.001	001-Datei	102 4
Mein_wunderschöner_Film.7z.002	002-Datei	102 4
Mein_wunderschöner_Film.7z.003	003-Datei	102 4
Mein_wunderschöner_Film.7z.004	004-Datei	31 1

Am Ende des Archivierungsprozesses liegen im Zielordner eine Reihe von `Dateiname.7z.***`-Files, die bis auf das letzte File (außer es handelt sich um eine sehr passend ausgewählte Split-Size) die angegebene Größe aufweisen.

## Archiv entpacken

Das Entpacken des Archives kann später dann einfach wieder über das Kontextmenü erfolgen. Dabei vergleicht das Tool auch die beim Verpacken errechnete Checksumme des Files und warnt, sollte ein Archivteil korrupt sein.



- Note: Das reine entpacken funktioniert unter MacOS auch mit dem integrierten Archivierungstool.

## Extra: Checksumme erstellen

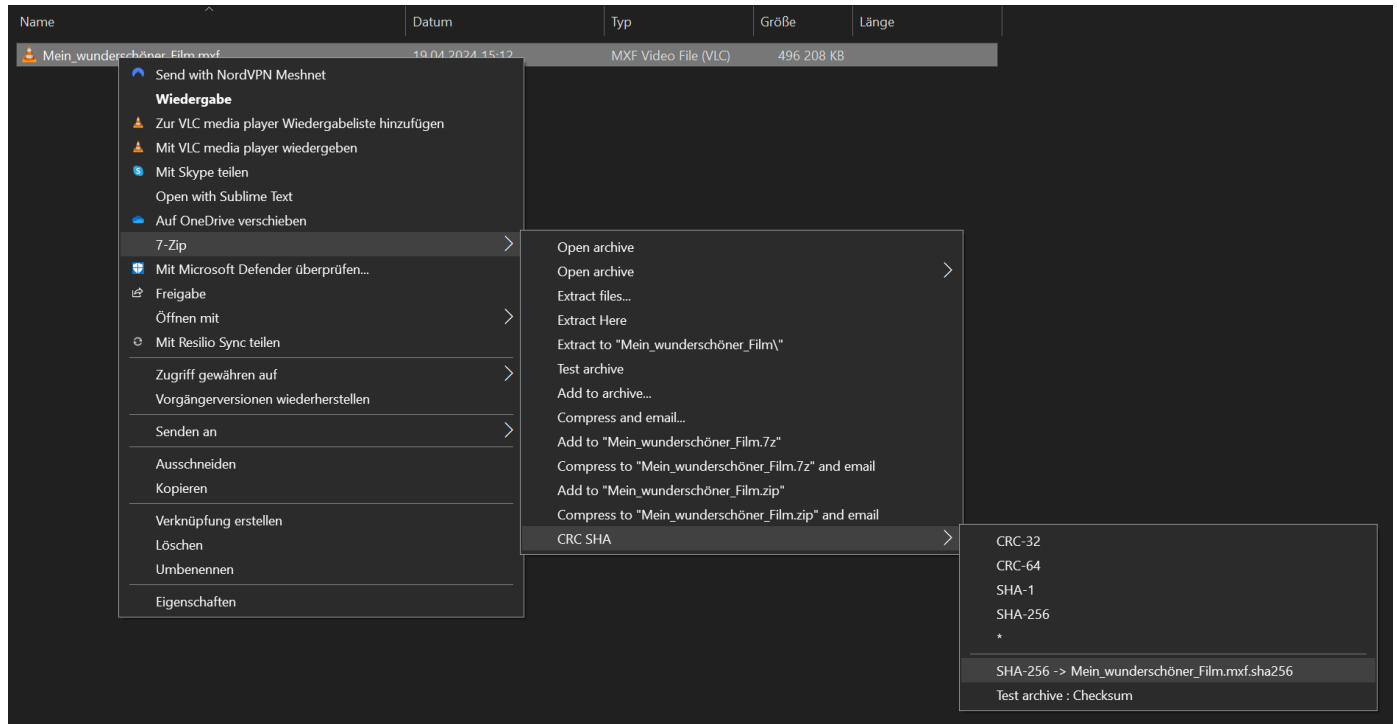
### Warum?

Da Dateien bei Kopiervorgängen in seltenen Fällen auch beschädigt werden können, ist es eine recht gute Idee, eine Checksumme der Originalversion zu erstellen.

Eine Checksumme errechnet aus dem Inhalt (unabhängig vom Dateinamen) einer Datei eine Zeichenfolge, die sich wesentlich verändert, wenn sich in der Datei auch nur ein einzelnes Byte verändert. So können Kopierfehler schnell entdeckt werden und Dateien auf ihre Unversehrtheit überprüft werden.

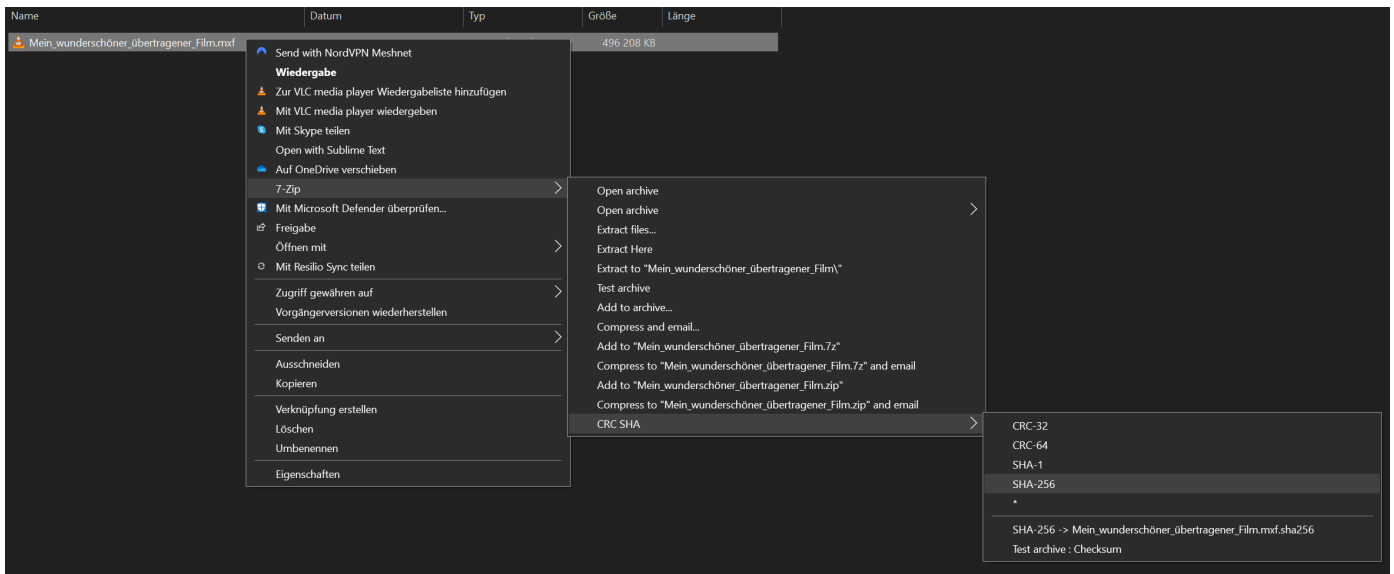
## Checksumme meiner Ausgangsdatei erstellen

Eine Checksumme für eine Datei kann sehr einfach über das 7-Zip Kontextmenü erstellt werden:



Dieser Befehl erstellt eine Datei mit der SHA-256 Checksumme am Speicherort der gewählten Datei. Dieses wenige Bytes große File dient später als Grundlage, wenn ich nach einer Übertragung die Unversehrtheit meines Files überprüfen möchte.

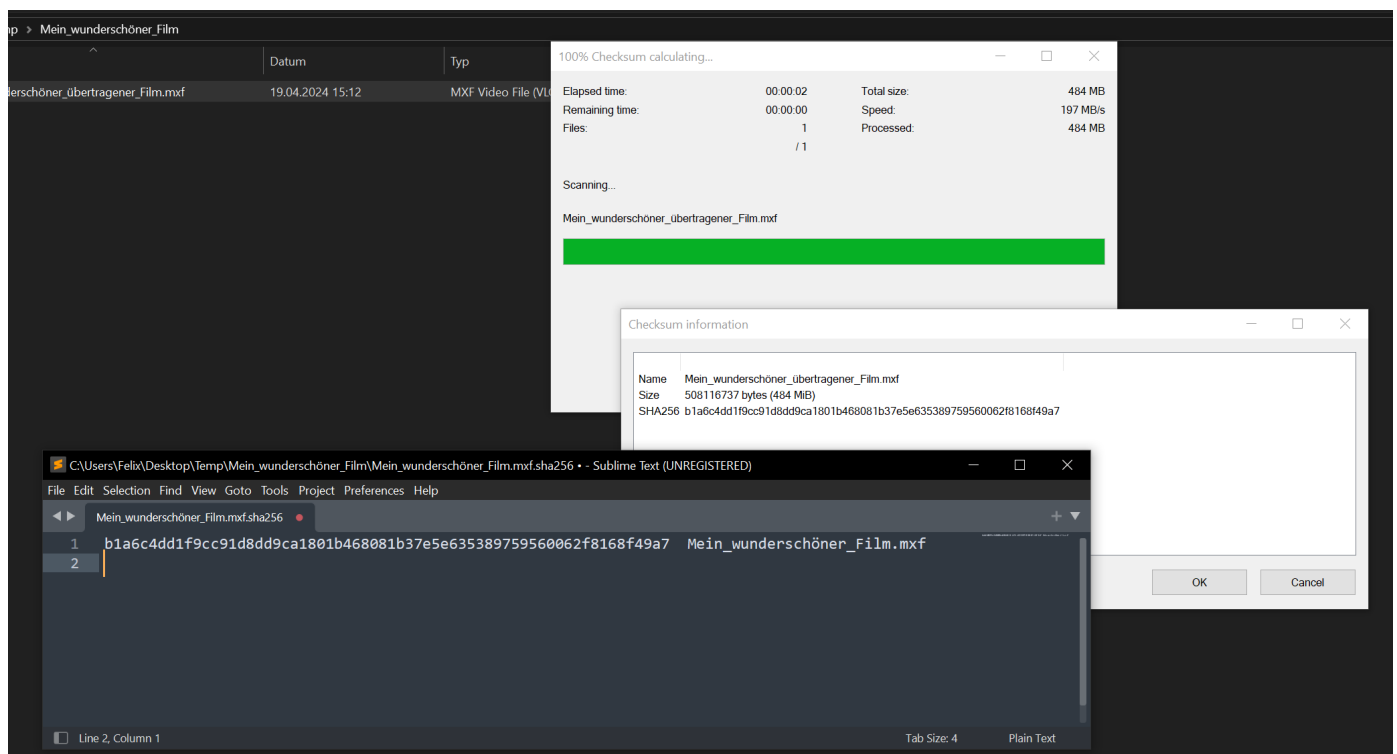
## Checksumme der kopierten Datei erstellen



Nach der Übertragung kann ich ebenso über das Kontextmenü eine Checksumme meiner Datei anzeigen lassen.

Diese Checksumme vergleiche ich dann mit dem Wert der Checksumme meiner Ausgangsdatei (Dateinamen spielen dabei keine Rolle).

## Checksummen vergleichen



Erfreulicherweise stimmen in diesem Beispiel die Checksummen überein, was bedeutet, dass es bei der Übertragung keine bösen Überraschungen gab.

- Note: weicht die kopierte Datei auch nur minimal vom Original ab, entscheidet sich die Checksumme wesentlich
- Note: Es ist unwesentlich, mit welchem Tool die Checksumme erstellt wurde. Sofern der selbe Algorithmus (in diesem Fall SHA-256) verwendet und richtig implementiert wurde, kommt für

dasselbe File immer diesselbe Summe raus.