

Claude auf einen anderen PC umziehen

So nehmen Sie das Gedächtnis, die Regeln und die Skills mit — für
jedes Projekt

Version 2.4 · 27. Oktober 2026

Hans Tandt

Inhalt

1. Wozu dieses Handbuch dient.....	3
2. Wo Claude sein Gehirn aufbewahrt.....	3
2.1 Die fünf wichtigsten Teile.....	4
2.2 Die Teile, die man übersieht, und wozu sie dienen.....	5
2.3 Drei verschiedene Dinge, leicht zu verwechseln.....	6
2.4 Eine Sitzungsnotiz führen und jede Sitzung aktualisieren.....	6
3. Bevor Sie etwas kopieren.....	7
4. Alles mit zwei Prompts einsammeln.....	7
4.1 Die sieben Fragen des Exports.....	8
4.2 Was der Export auf dem Laufwerk hinterlässt.....	9
4.3 Prüfen, bevor Sie das Laufwerk abziehen.....	10
4.4 Er schreibt, was fehlt.....	10
5. Was Sie zurücklassen.....	11
6. Alles auf dem neuen PC wiederherstellen.....	11
6.1 Die vier Fragen des Imports.....	12
6.2 Er führt zusammen, er überschreibt nicht.....	12
6.3 Der Name des Gedächtnisordners wird ein anderer sein.....	13
6.4 Programme und Umgebungsvariablen.....	13
7. Prüfen, ob es funktioniert hat.....	14
7.1 Wenn noch etwas fehlt.....	15
8. Eine kurze Checkliste.....	15
9. Glossar.....	15

1. Wozu dieses Handbuch dient

Sie arbeiten seit Monaten mit Claude auf einem PC. Claude hat gelernt, wie Sie arbeiten, worum es in diesem Projekt geht und welche Fehler zu vermeiden sind. Jetzt brauchen Sie denselben Claude auf einem anderen PC.

Den Projektordner zu kopieren reicht nicht. Das meiste, was Claude gelernt hat, liegt woanders, und eine Kopie des Ordners lässt es zurück. Dieses Handbuch zeigt, was Sie mitnehmen müssen und wie Sie es wieder einrichten.

HINWEIS — Gilt für jedes Projekt

Nichts hier ist auf ein bestimmtes Projekt oder eine bestimmte Firma zugeschnitten. Gehen Sie bei einem Handbuch, einem Programm oder einem Ordner mit Dokumenten genauso vor.

HINWEIS — Wer das gemacht hat

Diese Prompts und dieses Handbuch hat Hans Tandt (hanstandt.be) erstellt. Sie sind kostenlos unter der MIT-Lizenz: Sie dürfen sie nutzen, kopieren und anpassen, solange Sie diese Zeile behalten.

2. Wo Claude sein Gehirn aufbewahrt

Claude speichert sein Wissen an zwei Orten. Nur einer davon reist mit, wenn Sie den Projektordner kopieren.



Der Großteil des Gehirns liegt rechts.

Eine Kopie des Projektordners nimmt nur die linke Seite mit. Die rechte Seite ist der Teil, der verloren geht.

Die rechte Seite liegt in Ihrem Benutzerprofil, in einem versteckten Ordner namens `.claude`. Unter Windows ist das `C:\Users\\.claude`. Alle Projekte auf dem PC teilen sich diesen Ordner, und genau deshalb ist er in einer Projektkopie nicht enthalten.

2.1 Die fünf wichtigsten Teile

Was	Was es enthält	Wo es liegt
Automatisches Gedächtnis	Was Claude über Sie und das Projekt herausgefunden hat, von Claude selbst geschrieben	<code>~/.claude/projects/<Projekt>/memory/</code>
Skills	Ihre Arbeitsweisen: wie man ein Handbuch erstellt, wie man Arbeit prüft, bevor man sie für fertig erklärt	<code>~/.claude/skills/</code>
Persönliche CLAUDE.md	Ihre Regeln für jedes Projekt auf diesem PC	<code>~/.claude/CLAUDE.md</code>
Projekt-CLAUDE.md	Die Regeln für dieses eine Projekt	Im Projektordner
Ihre Notizdatei	Entscheidungen und die Gründe dafür, von Ihnen geschrieben	Im Projektordner

FALLE — Skills gehen am leichtesten verloren

Skills liegen außerhalb des Projekts. Wenn Sie nur den Projektordner kopieren, bleibt jeder Skill, den Sie je geschrieben haben, auf dem alten PC. Es sind kleine Dateien, aber in ihnen steckt die meiste Arbeit — nehmen Sie sie also zuerst mit.

2.2 Die Teile, die man übersieht, und wozu sie dienen

Diese fünf sind die wichtigsten. Das Profil enthält meist mehr, und was folgt, ist der Teil, den man übersieht, weil einem der Name nichts sagt. Jeder ist ein kleiner Ordner. Jeder ist Arbeit, die Sie bereits geleistet haben.

Was	Was es enthält	Was Sie ohne es verlieren
<code>rules/</code>	Persönliche Regeln nach Thema, aus Ihrer persönlichen <code>CLAUDE.md</code> ausgelagert, damit diese kurz bleibt	Sie haben dasselbe Gewicht wie die <code>CLAUDE.md</code> selbst. Nichts meldet, dass sie fehlen.
<code>agents/</code>	Von Ihnen geschriebene Subagents — ein Helfer mit eigenen Anweisungen für eine Art von Aufgabe	Es gibt sie nicht mehr. Keine Fehlermeldung; die Arbeit wird einfach auf dem gewöhnlichen, langsameren Weg erledigt.
<code>commands/</code>	Ihre eigenen Befehle, jeder eine gespeicherte Anweisung, die Sie oft verwenden	Eine Gewohnheit funktioniert nicht mehr. Sie tippen den Befehl, und nichts passiert.
<code>output-styles/</code>	Wie Antworten aussehen — Länge, Ton, wie viel erklärt wird	Claude klingt auf dem neuen PC anders, und Sie wissen nicht, warum.
<code>workflows/</code>	Aufgaben aus mehreren Schritten, gespeichert und als Ganzes ausgeführt	Meist das, was in dieser Tabelle am längsten dauert, um es aus dem Gedächtnis neu aufzubauen.
<code>plugins/</code>	Pakete mit Skills und Befehlen aus einem Marketplace	Diese werden installiert, nicht kopiert. Nehmen Sie die Liste mit, nicht den Ordner.
<code>.mcp.json</code>	Welche MCP-Server ein Projekt erreichen darf — die externen Werkzeuge, die es nutzen kann. Diese Datei liegt im Projekt, nicht im Profil	Das Projekt verliert seine Werkzeuge und wirkt ohne sichtbaren Grund defekt.

HINWEIS — Dieselben Namen gibt es auf beiden Seiten

Ein Projekt kann einen eigenen Ordner `.claude\` haben, mit `rules\`, `skills\`, `agents\`, `commands\` und `workflows\`, neben `.mcp.json` und `CLAUDE.local.md`. Diese reisen mit einer Kopie des Ordners mit. Die in Ihrem Benutzerprofil nicht. Wenn der Export beide auflistet, prüfen Sie, auf welcher Seite der Linie jeder davon liegt.

2.3 Drei verschiedene Dinge, leicht zu verwechseln

	Wer es schreibt	Wie lange es bleibt
Transkript	Automatisch aufgezeichnet	Wird nach einer Weile gelöscht. Verlassen Sie sich nicht darauf.
Automatisches Gedächtnis	Claude, für sich selbst	Bleibt, liegt aber außerhalb des Projekts
Sitzungsnotiz	Sie, mit Claudes Hilfe	Bleibt und reist mit dem Projekt mit

Die Sitzungsnotiz ist die einzige der drei, die Sie selbst in der Hand haben, die erhalten bleibt und die ein neuer Claude am ersten Tag lesen kann. Sie ist die wertvollste Datei der Übergabe — und die, an die niemand denkt.

2.4 Eine Sitzungsnotiz führen und jede Sitzung aktualisieren

Eine Sitzungsnotiz ist eine Datei im Projekt. Sie hält fest, was getan wurde, was entschieden wurde und warum, und was noch offen ist. Schreiben Sie sie in einfachen Sätzen, für einen Leser, der nicht dabei war.

Machen Sie es sich zur Gewohnheit: Bitten Sie Claude am Ende jeder Arbeitssitzung, die Sitzungsnotiz zu aktualisieren. Claude hat die ganze Sitzung vor sich und entwirft in einer Minute, wofür Sie zwanzig bräuchten. Lesen Sie dann, was es geschrieben hat — Sie sind derjenige, der weiß, ob es stimmt.

Gehört hinein	Bleibt draußen
Was sich geändert hat, und an welchem Datum	Alles, was Claude aus den Dateien selbst lesen kann
Was entschieden wurde, und der Grund dafür	Ein Protokoll des Gesprächs, Schritt für Schritt
Was versucht und verworfen wurde, und warum	Lob, Füllwörter oder eine Zusammenfassung dessen, was Sie schon wissen
Was noch offen ist, und der nächste Schritt	Alles, bei dem Sie nicht sicher sind, dass es stimmt

TIPP — Der Grund zählt mehr als die Entscheidung

„Wir verwenden LibreOffice für das PDF“ ist für sich genommen wenig wert. „Wir verwenden LibreOffice, weil Word hängen bleibt, wenn es unbeaufsichtigt läuft“ verhindert, dass der Nächste es rückgängig macht.

FALLE — Lassen Sie die Notiz nicht zum Tagebuch werden

Eine Notiz, die niemand liest, ist so gut wie keine. Beschränken Sie sie auf Entscheidungen, Gründe und offene Punkte. Verschieben Sie erledigte Details in die eigenen Dokumente des Projekts.

3. Bevor Sie etwas kopieren

Erledigen Sie zuerst diese drei Dinge auf dem Quell-PC. Sie dauern zehn Minuten und entscheiden darüber, wie viel Sie wirklich mitnehmen.

1. Sorgen Sie dafür, dass das Projekt eine `CLAUDE.md` hat. Fehlt sie, bitten Sie Claude, `/init` auszuführen, lesen Sie dann, was es geschrieben hat, und korrigieren Sie es. Diese Datei erklärt einem neuen Claude das Projekt.
2. Aktualisieren Sie die Sitzungsnotiz. Bitten Sie Claude, sie auf den neuesten Stand zu bringen, und lesen Sie sie dann. Auf dem neuen PC wird sie als Erstes gelesen, also muss sie stimmen.
3. Bitten Sie Claude, sein Gedächtnis aufzuräumen. Öffnen Sie `/memory`, entfernen Sie Veraltetes und führen Sie doppelt Geschriebenes zusammen. Nehmen Sie keine veralteten Fakten auf den neuen PC mit.

TIPP — Warum zuerst aufräumen

Ein falscher Fakt, der auf den neuen PC mitreist, ist schlimmer als ein fehlender. Sie werden ihm vertrauen, und er wird falsch sein.

4. Alles mit zwei Prompts einsammeln

Zwei Prompts erledigen die Arbeit. Es sind einfache Textdateien, gemacht zum Einfügen in Claude Code — keine Dokumente zum Lesen am Bildschirm. Dieses Handbuch beschreibt deren Version 3.2.

Datei	Ausführen auf	Was sie tut
<code>1_EXPORT_PROMPT.txt</code>	Dem Quell-PC	Stellt sieben Fragen und arbeitet dann fünf Aufgaben ab: Sie erfasst das Projekt, sammelt alles, was außerhalb liegt, schreibt, was fehlt, kopiert alles und prüft das Ergebnis
<code>2_IMPORT_PROMPT.txt</code>	Dem neuen PC	Stellt vier Fragen und arbeitet dann sieben Aufgaben ab: Sie liest, was der Export auf dem Laufwerk hinterlassen hat, legt jeden Teil an die richtige Stelle, weist nach, dass er angekommen ist, und geht für alles, was noch fehlt, zurück zum alten PC

Öffnen Sie Claude Code mit dem Projektordner als Arbeitsverzeichnis und fügen Sie den Prompt als eine einzige Nachricht ein. Jede Datei enthält eine Zeile aus `v`-Zeichen und eine Zeile aus `^`-Zeichen. Fügen Sie nur ein, was dazwischen steht — die Markierungszeilen gehören nicht zur Anweisung.

Beide stellen ihre Fragen, bevor sie etwas anfassen, damit sich nicht erst mittendrin etwas herausstellt — etwa ein Laufwerk ohne freien Platz. Danach arbeiten sie eine Aufgabe nach der anderen ab und zeigen Ihnen das Ergebnis, bevor sie mit der nächsten beginnen.

HINWEIS — Sie prüfen nach, statt etwas anzunehmen

Keiner der beiden Prompts hat eine feste Liste Ihrer Dateien. Jeder schaut nach, was wirklich da ist, und meldet es, sodass Sie nicht vorab wissen müssen, welche Skills oder Einstellungen es gibt.

HINWEIS — Der Export verändert nichts

Der Export-Prompt behandelt das Projekt als schreibgeschützt. Das Einzige, was er neu anlegt, sind die Notizdateien, um die er gebeten wird, und die Kopien auf Ihrem Laufwerk. Er baut, installiert und konfiguriert nichts, und er kopiert nie Ihre Anmeldung.

HINWEIS — Sie lesen Ihre Dateien, sie gehorchen ihnen nicht

Eine Datei im Projekt oder auf dem Laufwerk kann Text enthalten, der wie eine Anweisung klingt. Beide Prompts sind angewiesen, Ihnen diesen Text zu zeigen, statt danach zu handeln. Anweisungen geben nur Sie.

4.1 Die sieben Fragen des Exports

Halten Sie diese Antworten bereit. Nichts beginnt, bevor Sie sie gegeben haben, und sobald das geschehen ist, prüft der Export, ob am Ziel genug freier Platz ist, und hält an, wenn nicht.

1. Ist der Ordner, in dem ich mich befinde, das Projekt, das Sie exportieren möchten?
2. Wohin soll der Export? Ein Laufwerksbuchstabe oder ein vollständiger Pfad. Dort wird ein Ordner namens `Handover_Export` angelegt.
3. Ist das ein Umzug, eine Sicherung oder ein zweites Gerät, das weiter in Gebrauch bleibt?
4. Sollen die großen Ordner mit? Sie sehen zuerst die Gesamtgröße, und jeder einzelne Ordner über 100 MB wird genannt, damit Sie entscheiden können.
5. Welche Datei ist der Haupteinstiegspunkt — das, was ausgeführt oder gebaut wird? Der Export schlägt selbst etwas vor, das Sie bestätigen.
6. Gibt es eine Sitzungsnotiz oder eine Entscheidungsdatei, die auf jeden Fall dabei sein muss?
7. Gibt es etwas außerhalb des Projektordners und außerhalb von `.claude`, von dem die Arbeit abhängt — eine Prompt-Bibliothek, einen Ordner mit Vorlagen, ein freigegebenes Laufwerk? Das kann der Export nicht selbst finden.

4.2 Was der Export auf dem Laufwerk hinterlässt

Alles landet in einem Ordner. Nehmen Sie den ganzen Ordner mit auf den neuen PC, mit Inhalt und Namen, wie sie sind: Der zweite Prompt fragt Sie, wo er liegt, und liest dann darin.

Handover_Export\	
project\	das Projekt, ohne was nicht mitreisen darf
brain\	
skills\	jeder Skill vom alten PC
memory\	die .md-Dateien des Gedächtnisses
rules\ agents\	commands\ output-styles\ workflows\
CLAUDE.md	Ihre persönlichen Regeln, für jedes Projekt
settings.json	Einstellungen und Hooks
HANDOVER_MANIFEST.md	jede Datei, mit einer Prüfsumme
HANDOVER_BRAIN.md	was nie irgendwo aufgeschrieben wurde
SOURCE_FACTS.md	die alten Pfade, Programme und Variablen
robocopy_log.txt	der Nachweis, was kopiert wurde

Was der Export erzeugt. Dieser Ordner ist die Übergabe.

Die drei Notizdateien	Wozu sie dient
HANDOVER_MANIFEST.md	Jede Datei im Projekt mit ihrer Größe und einer Prüfsumme, damit Sie auf dem neuen PC nachweisen können, dass Sie die richtige Version haben und unterwegs nichts verloren ging
HANDOVER_BRAIN.md	Entscheidungen und ihre Gründe, mündlich Vereinbartes, das nie aufgeschrieben wurde, nie beantwortete Fragen. Lesen Sie diese Datei selbst — sie ist der Teil, den keine Ordnerkopie enthält
SOURCE_FACTS.md	Der alte Projektpfad, der alte Name des Gedächtnisordners, die Programme, die Umgebungsvariablen und was zusammengeführt statt ersetzt werden muss. Der Import-Prompt liest diese Datei

HINWEIS — Es ist ein Ordner, keine ZIP-Datei

Hier ist nichts komprimiert. Der Import-Prompt liest `SOURCE_FACTS.md` direkt vom Laufwerk, und die Prüfung vor dem Abziehen öffnet kopierte Dateien, um nachzuweisen, dass sie vollständig angekommen sind — beides geht mit einem einfachen Ordner leichter, und eine ZIP-Datei, die unbemerkt beschädigt ist, liefert Ihnen gar nichts. Wenn Sie den Export über ein Netzwerk senden, statt ihn mitzunehmen, packen Sie den ganzen Ordner `Handover_Export` für den Transport in eine ZIP-Datei und entpacken Sie ihn, bevor Sie den Import ausführen.

4.3 Prüfen, bevor Sie das Laufwerk abziehen

Der Export endet mit einer Prüfung. Lesen Sie sie, statt sie nur zu überfliegen. Solange das Laufwerk noch im Quell-PC steckt, können Sie befehlen, was sie findet; sind Sie einmal gegangen, ist dieses Laufwerk vielleicht die einzige Kopie, die Sie haben.

FALLE — Eine Kopie, die Erfolg meldet, ist kein Beweis

Ein Kopierprogramm kann Erfolg melden und trotzdem eine abgeschnittene Datei geschrieben haben. Der Export öffnet zwei oder drei der kopierten Dateien — eine `SKILL.md` und eine `.md`-Datei aus dem Gedächtnis — und bestätigt, dass sie vollständig sind, nicht leer und nicht abgeschnitten.

Er bestätigt außerdem Folgendes, bevor Sie gehen — und wenn etwas fehlt, ist jetzt der Moment, es herauszufinden:

- Jede der drei Notizdateien ist vorhanden, mit ihrer Größe
- `project\`, `brain\skills\` und `brain\memory\` sind alle angekommen
- Nichts, was nicht mitreisen darf, ist auf dem Laufwerk gelandet — vor allem nicht `.credentials.json`
- Anzahl und Gesamtgröße der Dateien stimmen mit dem überein, was der Kopiervorgang selbst gemeldet hat
- Eine einfache Liste von allem auf dem Laufwerk, damit Sie die ganze Ernte auf einen Blick sehen

4.4 Er schreibt, was fehlt

Kapitel 3 bittet Sie, `CLAUDE.md` und die Sitzungsnotiz in Ordnung zu bringen, bevor Sie beginnen. Das macht nicht jeder, und eine dünne Sitzungsnotiz reist genauso schlecht wie gar keine.

Deshalb beschwert sich der Export nicht mehr nur. Seine dritte Aufgabe bietet an, das Fehlende zu schreiben, aus der Sitzung heraus, in der Sie schon sind: eine `CLAUDE.md`, falls keine existiert, und einen neuen, datierten Eintrag für die Sitzungsnotiz. Er zeigt Ihnen zuerst jede Zeile und schreibt nichts, bevor Sie zustimmen.

HINWEIS — Er ergänzt nur

Er darf eine Datei anlegen, die es nicht gibt, oder etwas an eine bestehende anhängen. Er wird nie eine Zeile, die Sie geschrieben haben, umschreiben oder entfernen. Sagen Sie Nein, lässt er die Datei in Ruhe und macht weiter — eine Lücke, die Sie kennen, ist in Ordnung.

TIPP — Das lohnt sich auch, wenn Sie nicht umziehen

Führen Sie die dritte Aufgabe des Export-Prompts für ein Projekt aus, das genau dort bleibt, wo es ist, und Sie erhalten eine `CLAUDE.md` und eine Sitzungsnotiz, die geschrieben wurden, solange die Arbeit noch frisch war. Das ist es wert, ob Sie umziehen oder nicht.

5. Was Sie zurücklassen

Manches darf nie kopiert werden. Es ist entweder privat, oder es wird auf dem neuen PC automatisch neu erstellt, und eine Kopie verursacht Fehler.

Nicht kopieren	Warum
<code>.credentials.json</code>	Ihre Anmeldung. Melden Sie sich stattdessen auf dem neuen PC an. Legen Sie sie nie auf einen USB-Stick.
API-Schlüssel, Tokens, Passwörter	Nie als Wert kopiert. Der Export nennt sie und schreibt SECRET - set by hand on the new PC. Sie richten sie dort selbst ein.
<code>.venv</code> , <code>node_modules</code>	Für einen bestimmten PC erstellt. Legen Sie sie mit <code>py -m venv</code> und <code>npm install</code> neu an.
<code>__pycache__</code> , <code>build</code> , <code>dist</code> , <code>_internal</code>	Erzeugte Dateien. Sie entstehen von selbst neu.
<code>cache</code> , <code>shell-snapshots</code> , <code>telemetry</code> , <code>debug</code> , <code>sessions</code> , <code>file-history</code> , <code>history.jsonl</code>	Arbeitsdateien des Programms selbst. Für Sie ohne Nutzen.
<code>_archive</code> , sehr große <code>.zip</code> -Dateien	Meist alte Schnappschüsse dessen, was Sie ohnehin kopieren.
<code>.git</code>	Groß, und besser durch Klonen aus Ihrem Remote wiederherzustellen. Lesen Sie zuerst die Warnung unten.

FALLE — Wenn das Projekt ein Git-Repository ist

Der Export lässt `.git` weg, die Historie reist also nicht mit. Klonen Sie das Projekt stattdessen auf dem neuen PC aus seinem Remote.

Hat das Projekt kein Remote, existiert seine Historie nur auf dem alten PC. Kopieren Sie `.git` dann von Hand, bevor Sie etwas löschen.

6. Alles auf dem neuen PC wiederherstellen

Diese Arbeit erledigt der zweite Prompt. Sie bereiten den PC vor, fügen den Prompt ein und beantworten seine Fragen. Danach arbeitet er sieben Aufgaben ab: was auf diesem PC fehlt, die Projektdateien (wobei der Haupteinstiegspunkt mit der Prüfsumme im Manifest verglichen wird, damit Sie wissen, dass Sie die richtige Version haben), die Skills und Einstellungen, das Gedächtnis, die Umgebungsvariablen, eine Checkliste und zum Schluss alles, was noch fehlt — und er zeigt Ihnen jedes Ergebnis, bevor er weitermacht.

1. Schließen Sie das Laufwerk mit `Handover_Export` an.
2. Legen Sie den Ordner an, in dem das Projekt liegen soll. Leer ist in Ordnung.
3. Öffnen Sie Claude Code mit diesem Ordner als Arbeitsverzeichnis. Das ist wichtig: So legt Claude den Gedächtnisordner unter dem richtigen neuen Namen an.
4. Fügen Sie `2_IMPORT_PROMPT.txt` als eine einzige Nachricht ein, ohne die Markierungszeilen.

TIPP — Überlassen Sie die kniffligen Schritte dem Prompt

Das Gedächtnis zurückzuspielen heißt, einen Ordner zu finden, dessen Name sich mit dem Pfad geändert hat. Die Einstellungen wiederherzustellen heißt, zwei Dateien zusammenzuführen, statt eine zu ersetzen. Genau solche Schritte überlässt man besser einer Maschine, statt sie am Ende eines langen Tages von Hand zu erledigen.

6.1 Die vier Fragen des Imports

Er stellt diese vier Fragen, liest dann `SOURCE_FACTS.md` und zeigt Ihnen den alten Projektpfad, den alten Namen des Gedächtnisordners, die Programme und Variablen, die Sie brauchen, und alles, was zusammengeführt statt ersetzt werden muss. Stellt sich heraus, dass der Zielordner nicht leer ist, listet er dessen Inhalt auf und hält an, um Sie zu fragen.

1. Wo liegt `Handover_Export`? Ein Laufwerksbuchstabe oder ein vollständiger Pfad.
2. Ist der Ordner, in dem ich mich befinde, der Ort, an dem das Projekt liegen soll?
3. Ist dieser PC neu, oder hat er bereits Claude-Skills und -Einstellungen, die es wert sind, behalten zu werden?
4. Soll ich die fehlenden Programme installieren oder nur auflisten?

6.2 Er führt zusammen, er überschreibt nicht

Der neue PC hat vielleicht schon eine eigene Claude-Einrichtung. Dateien einfach komplett zu ersetzen, würde die Berechtigungen und Hooks wegwerfen, die dort entstanden sind. Der Prompt ist so geschrieben, dass er das nicht tut.

Was	Wie es behandelt wird
Skills	Ordner, also liegen sie nebeneinander. Ein Skill mit demselben Namen wird nie überschrieben: Sie sehen beide — Datum, Größe und ein eventuelles Änderungsprotokoll — und entscheiden.
Persönliche <code>CLAUDE.md</code>	Eine einzelne Datei. Hat dieser PC keine, wird sie hineinkopiert. Hat er eine, werden beide vollständig gezeigt, und eine zusammengeführte Fassung wird vorgeschlagen. Nichts wird geschrieben, bevor Sie zustimmen.
<code>settings.json</code>	Eine einzelne Datei, genauso behandelt: beide Fassungen, die Schlüssel, die sich unterscheiden, und eine vorgeschlagene Zusammenführung zu Ihrer Freigabe.
<code>rules\</code> , <code>agents\</code> , <code>commands\</code> , <code>output-styles\</code> , <code>workflows\</code>	Datei für Datei kopiert. Was hier schon existiert, wird übersprungen, und Sie erfahren, welche.
Gerätegebundene Dateien in einem Skill	<code>node_modules</code> , <code>__pycache__</code> und <code>profile.json</code> bleiben absichtlich zurück. Sie gehören zum alten PC — <code>profile.json</code> ist die Datei, die festlegt, wer dort Dokumente unterzeichnet.
Plugins	Nicht kopiert. Er listet auf, welche Plugins und welche Marketplaces der alte PC hatte, und gibt Ihnen den Befehl, um jedes hier zu installieren. Plugins werden installiert, nicht kopiert.

6.3 Der Name des Gedächtnisordners wird ein anderer sein

Dies ist der Schritt, der am häufigsten schiefgeht. Der Gedächtnisordner ist nach dem vollständigen Pfad des Projekts benannt. Legen Sie das Projekt woanders ab, ändert sich der Name.



Kopieren Sie die Gedächtnisdateien in den Ordner, der nach dem NEUEN Pfad benannt ist, nicht nach dem alten.

Der Prompt erledigt das in seiner vierten Aufgabe. Er sucht den Ordner für den Pfad dieses PCs, bestätigt, dass es nicht der alte Name aus `SOURCE_FACTS.md` ist, kopiert die `.md`-Dateien hinein, liest danach `MEMORY.md` zurück und sagt Ihnen, wie viele Einträge angekommen sind. Er listet auch jeden Gedächtniseintrag auf, der hier jetzt falsch ist — ein alter Laufwerksbuchstabe, ein alter Pfad, eine Notiz über den Quell-PC — und überlässt Ihnen die Entscheidung.

So machen Sie es von Hand, oder prüfen die Arbeit des Prompts:

1. Starten Sie Claude Code einmal im Projektordner. Dadurch wird der Ordner mit dem richtigen Namen angelegt.
2. Schauen Sie in `C:\Users\<Ihr Name>\.claude\projects\` nach und suchen Sie den neuen Namen.
3. Kopieren Sie die `.md`-Dateien vom alten PC in den Ordner `memory\` darin.
4. Starten Sie Claude Code neu und fragen Sie: „Was weißt du noch über dieses Projekt?“

FALLE — In den alten Namen zu kopieren bewirkt nichts

Ein Ordner, der nach dem alten Pfad benannt ist, wird nie wieder gelesen. Die Dateien liegen dort, sehen richtig aus und sind für Claude unsichtbar.

6.4 Programme und Umgebungsvariablen

Dass die Dateien angekommen sind, heißt noch nicht, dass das Projekt läuft. Zwei Dinge reisen nie mit: installierte Programme und Umgebungsvariablen. Beide müssen Sie von Hand einrichten.

Die erste Aufgabe des Import-Prompts meldet, welche Programme und Pakete fehlen, und gibt für jedes den genauen Befehl aus. Die fünfte Aufgabe listet jede Umgebungsvariable des alten PCs auf, sagt, ob sie hier schon gesetzt ist und auf welchen Wert, und gibt Ihnen den `setx`-Befehl mit dem für diesen PC angepassten Pfad.

FALLE — Eine neue Variable braucht ein neues Terminal

`setx` schreibt die Variable, aber ein Fenster, das schon offen ist, sieht sie nie. Schließen Sie Claude Code danach und öffnen Sie es erneut.

7. Prüfen, ob es funktioniert hat

Nehmen Sie nichts an. Fragen Sie nach und vergleichen Sie mit dem, was Sie mitgebracht haben.

FALLE — Neu starten, bevor Sie etwas prüfen

Das Gedächtnis wird nur beim Start einer Sitzung gelesen. Schließen Sie Claude Code zuerst und öffnen Sie es erneut, sonst antwortet eine völlig korrekte Einrichtung so, als erinnere sie sich an nichts.

Fragen Sie Claude	Sie sollten erhalten
„Was weißt du noch über dieses Projekt?“	Die Fakten vom alten PC, keine leere Antwort
„Welche Skills hast du?“	Dieselbe Liste wie auf dem alten PC
„Welche Anweisungsdateien hast du geladen?“ (<code>/context</code>)	Ihre Projekt- und Ihre persönliche <code>CLAUDE.md</code>
Führen Sie den projekteigenen Build oder Test aus	Dasselbe Ergebnis wie auf dem alten PC

Der Import-Prompt schließt mit vier Listen: was kopiert wurde, was übersprungen wurde und warum, was noch Ihre Entscheidung braucht und was gar nicht möglich war. Lesen Sie sie, bevor Sie dem neuen PC vertrauen.

TIPP — Behalten Sie den Stick, bis das bestanden ist

Löschen Sie den Quell-PC oder den Stick erst, wenn jede Zeile oben richtig beantwortet ist. Bis dahin ist es die einzige Kopie, die Sie haben.

7.1 Wenn noch etwas fehlt

Betrachten Sie die Checkliste nicht als Ende. Kam eine Antwort falsch zurück oder läuft das Projekt nicht, sind Sie noch nicht fertig — und der alte PC steht noch da und kann antworten. Die siebte Aufgabe des Import-Prompts nutzt das.

1. Sie nennt genau, was ihr fehlt — eine bestimmte Datei, ein Skill, eine Einstellung, ein Fakt.
2. Sie teilt das auf: was der alte PC noch liefern kann, und was nur Sie hier tun können (Programm installieren, Plugin hinzufügen, Variable setzen).
3. Für die erste Gruppe schreibt sie Ihnen einen Prompt zum Einfügen in Claude Code auf dem ALTEN PC, mit dem Namen des Projekts und genau dem, was sie braucht.
4. Sie bringen die Antwort zurück, die Checkliste läuft erneut — so oft wie nötig.

FALLE — Deshalb wird noch nichts gelöscht

Die ganze Schleife setzt voraus, dass der alte PC noch da ist. Behalten Sie ihn und das Laufwerk, bis die Checkliste fehlerfrei zurückkommt. Beide Prompts sagen das jetzt in ihrer letzten Zeile.

8. Eine kurze Checkliste

- Projektordner, ohne `.venv`, `node_modules`, `build`, `dist`, `__pycache__`
- Die Sitzungsnotiz, am Tag des Kopierens auf den neuesten Stand gebracht
- Projekt-CLAUDE.md und Ihre Notizdateien (bereits im Ordner)
- Der ganze Ordner `~/.claude/skills/`
- `~/.claude/CLAUDE.md` und `~/.claude/rules/`, falls vorhanden
- `~/.claude/settings.json`
- `~/.claude/agents/`, `commands/`, `output-styles/`, `workflows/`, falls vorhanden
- Die `.md`-Dateien aus `~/.claude/projects/<Projekt>/memory/`
- Eine schriftliche Liste der installierten Plugins — die Liste, nicht der Ordner
- Eine schriftliche Liste der Umgebungsvariablen und installierten Programme
- `HANDOVER_MANIFEST.md`, `HANDOVER_BRAIN.md` und `SOURCE_FACTS.md`
- `.git`, nur wenn das Projekt kein Remote zum Klonen hat
- Niemals: `.credentials.json`

9. Glossar

Begriff	Bedeutung
Automatisches Gedächtnis	Was Claude über Sie und das Projekt für sich notiert.
Sitzungsnotiz	Eine Datei im Projekt, in der Sie festhalten, was entschieden wurde und warum. Sie schreiben sie, und sie reist mit dem Ordner mit.
CLAUDE.md	Eine Datei, die Sie schreiben, um Claude zu sagen, wie es arbeiten soll. Eine pro Projekt, plus eine persönliche.
Skill	Ein Ordner mit einer schriftlich festgehaltenen Arbeitsweise, mit einer <code>SKILL.md</code> darin.
Plugin	Ein Satz Skills und Befehle aus einem Marketplace. Wird installiert, nicht kopiert.

Begriff	Bedeutung
Handover_Export	Der Ordner, den der Export-Prompt auf Ihrem Laufwerk anlegt: Projekt, Gehirn und drei Notizdateien.
SOURCE_FACTS.md	Die Datei, die der Export schreibt und der Import liest: alte Pfade, Programme und Variablen.
Benutzerprofil	Ihr eigener Ordner auf dem PC: C:\Users\<Ihr Name>. Enthält den versteckten Ordner .claude.
Transkript	Aufzeichnung einer Sitzung, nach einiger Zeit gelöscht.
Umgebungsvariable	In Windows gespeicherte Einstellung, keine Datei. Reist nicht mit.
Hook	Ein Befehl, den Claude zu einem festgelegten Zeitpunkt von selbst ausführt, eingerichtet in settings.json.
MCP-Server	Externes Werkzeug, das ein Projekt nutzen darf (in .mcp.json).

Versionsgeschichte: Version 1.0 — 10. September 2026 — Erste Fassung, die einen Übergabe-Prompt für ein einzelnes Projekt durch eine allgemeine Methode ersetzt. Version 2.0 — 11. September 2026 — An Version 3.0 der beiden Prompts angepasst: die Fragen, die jeder stellt, was der Export auf dem Laufwerk hinterlässt, Zusammenführen statt Überschreiben, Git-Historie, Programme und Variablen. Version 2.1 — 11. September 2026 — Alles, was die beiden Prompts behandeln, wird jetzt erklärt, mit dem Grund, es mitzunehmen: die Profilordner, die bisher nur aufgelistet waren (2.2), die Prüfung vor dem Abziehen (4.3) und die Regel, dass die Prompts Ihre Dateien lesen, ohne ihnen zu gehorchen. Version 2.2 — 26. Oktober 2026 — An die Prompts Version 3.1 angepasst: Der Export schreibt jetzt eine fehlende CLAUDE.md oder Sitzungsnotiz, statt sich nur darüber zu beschweren (4.4), er fragt nach Ordnern außerhalb des Projekts (4.1), der Import kann für Fehlendes zum alten PC zurückgehen (7.1), und nichts darf gelöscht werden, bevor das bestanden ist. Version 2.3 — 26. Oktober 2026 — Sagt in 4.2 deutlich, dass der Export ein Ordner und keine ZIP-Datei ist, und wann das Zippen trotzdem richtig ist. Version 2.4 — 27. Oktober 2026 — An die Prompts Version 3.2 angepasst: Passwörter, Schlüssel und Tokens werden genannt, aber nie auf das Laufwerk geschrieben (Kapitel 5). Wer es gemacht hat, und die MIT-Lizenz (Kapitel 1). Niederländische und deutsche Fassung.