

Claude verhuizen naar een andere pc

Hoe u het geheugen, de regels en de skills meeneemt — voor elk project

Versie 2.4 · 27 oktober 2026

Hans Tandt

Inhoud

1. Waarvoor dit dient.....	3
2. Waar Claude zijn brein bewaart.....	3
2.1 De vijf belangrijkste onderdelen.....	4
2.2 De delen die mensen overslaan, en waarvoor elk dient.....	5
2.3 Drie verschillende dingen die u makkelijk verwacht.....	6
2.4 Houd een sessienotitie bij, en werk ze elke sessie bij.....	6
3. Voordat u iets kopieert.....	7
4. Alles verzamelen met twee prompts.....	7
4.1 De zeven vragen van de export.....	8
4.2 Wat de export op de schijf achterlaat.....	8
4.3 Controleer voordat u loskoppelt.....	9
4.4 De export schrijft wat ontbreekt.....	9
5. Wat u moet achterlaten.....	10
6. Alles terugzetten op de nieuwe pc.....	11
6.1 De vier vragen van de import.....	11
6.2 De prompt voegt samen, hij overschrijft niet.....	12
6.3 De naam van de geheugenmap zal anders zijn.....	12
6.4 Programma's en omgevingsvariabelen.....	13
7. Controleer of het gelukt is.....	13
7.1 Als er nog iets ontbreekt.....	14
8. Een korte controlelijst.....	14
9. Verklarende woordenlijst.....	15

1. Waarvoor dit dient

U werkt al maanden met Claude op één pc. Claude heeft geleerd hoe u werkt, waar dit project over gaat en welke fouten het moet vermijden. Nu hebt u diezelfde Claude nodig op een andere pc.

De projectmap kopiëren is niet genoeg. Het meeste van wat Claude heeft geleerd, staat ergens anders, en een kopie van de map laat dat achter. Deze handleiding toont wat u moet meenemen en hoe u het terugzet.

OPMERKING — Werkt voor elk project

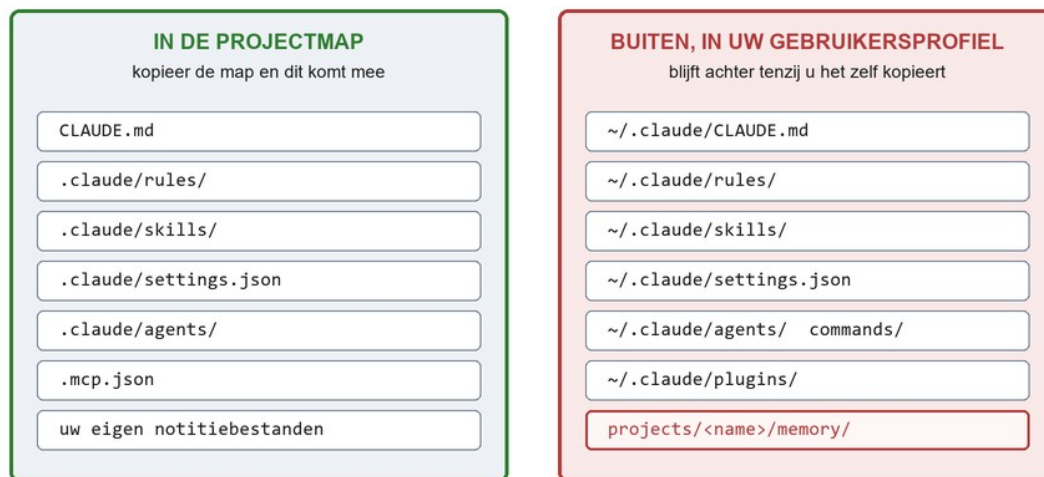
Niets hier is specifiek voor één project of één bedrijf. Volg dezelfde stappen voor een handleiding, een programma of een map met documenten.

OPMERKING — Wie dit maakte

Deze prompts en deze handleiding zijn gemaakt door Hans Tandt (hanstandt.be). Ze zijn gratis onder de MIT-licentie: u mag ze gebruiken, kopiëren en aanpassen, zolang u deze regel laat staan.

2. Waar Claude zijn brein bewaart

Claude bewaart wat het weet op twee plaatsen. Slechts één daarvan reist mee als u de projectmap kopieert.



Het grootste deel van het brein staat rechts.

Een kopie van de projectmap neemt alleen de linkerkant mee. De rechterkant is het deel dat mensen kwijtraken.

De rechterkant staat in uw gebruikersprofiel, in een verborgen map met de naam `.claude`. Onder Windows is dat `C:\Users\<uw naam>\.claude`. Alle projecten op de pc delen die map, en precies daarom zit ze niet in een kopie van het project.

2.1 De vijf belangrijkste onderdelen

Wat	Wat erin staat	Waar het staat
Automatisch geheugen	Wat Claude over u en het project heeft uitgezocht, door Claude zelf geschreven	<code>~/.claude/projects/<project>/memory/</code>
Skills	Uw werkwijzen: hoe u een handleiding bouwt, hoe u werk controleert voordat u zegt dat het klaar is	<code>~/.claude/skills/</code>
Persoonlijke CLAUDE.md	Uw regels voor elk project op die pc	<code>~/.claude/CLAUDE.md</code>
Project-CLAUDE.md	De regels voor dit ene project	In de projectmap
Uw notitiebestand	Beslissingen en de redenen erachter, door u geschreven	In de projectmap

VALKUIL — Skills raakt u het makkelijkst kwijt

Skills staan buiten het project. Kopieert u alleen de projectmap, dan blijft elke skill die u ooit schreef op de oude pc. Het zijn kleine bestanden en er zit het meeste werk in, dus neem ze als eerste mee.

2.2 De delen die mensen overslaan, en waarvoor elk dient

Die vijf zijn het belangrijkste. Het profiel bevat meestal meer, en wat volgt is het deel dat mensen overslaan, omdat de naam hun niets zegt. Elk is een kleine map. Elk is werk dat u al gedaan hebt.

Wat	Wat erin staat	Wat u zonder verliest
<code>rules/</code>	Persoonlijke regels per onderwerp, uit uw persoonlijke <code>CLAUDE.md</code> gehaald om die kort te houden	Ze wegen even zwaar als <code>CLAUDE.md</code> zelf. Niets meldt dat ze verdwenen zijn.
<code>agents/</code>	Subagents die u schreef — een helper met eigen instructies voor één soort taak	Ze bestaan niet meer. Er komt geen foutmelding; het werk gebeurt gewoon op de gewone, tragere manier.
<code>commands/</code>	Uw eigen opdrachten, elk een opgeslagen instructie die u vaak gebruikt	Een gewoonte werkt niet meer. U typt ze en er gebeurt niets.
<code>output-styles/</code>	Hoe antwoorden eruitzien — lengte, toon, hoeveel uitleg	Claude klinkt anders op de nieuwe pc en u weet niet waarom.
<code>workflows/</code>	Taken van meerdere stappen, opgeslagen en als één geheel uitgevoerd	Meestal het onderdeel in deze tabel dat het langst duurt om uit het hoofd opnieuw op te bouwen.
<code>plugins/</code>	Pakketten met skills en opdrachten uit een marketplace	Die worden geïnstalleerd, niet gekopieerd. Neem de lijst mee, niet de map.
<code>.mcp.json</code>	Welke MCP-servers een project mag bereiken — de externe tools die het kan gebruiken. Dit bestand staat in het project, niet in het profiel	Het project verliest zijn tools en lijkt zonder zichtbare reden kapot.

OPMERKING — Dezelfde namen bestaan aan beide kanten

Een project kan een eigen map `.claude\` hebben met `rules\`, `skills\`, `agents\`, `commands\` en `workflows\`, naast `.mcp.json` en `CLAUDE.local.md`. Die reizen mee met een kopie van de map. Die in uw gebruikersprofiel niet. Als de export beide opsomt, kijk dan aan welke kant van de lijn elk ervan staat.

2.3 Drie verschillende dingen die u makkelijk verwacht

	Wie het schrijft	Hoe lang het blijft
Transcript	Automatisch opgenomen	Na een tijd verwijderd. Reken er niet op.
Automatisch geheugen	Claude, voor zichzelf	Blijft, maar staat buiten het project
Sessienotitie	U, met hulp van Claude	Blijft, en reist mee met het project

De sessienotitie is de enige van de drie die u zelf in de hand hebt, die blijft bestaan en die een nieuwe Claude op de eerste dag kan lezen. Het is het waardevolste bestand van de overdracht, en het is het bestand dat niemand eraan denkt te schrijven.

2.4 Houd een sessienotitie bij, en werk ze elke sessie bij

Een sessienotitie is één bestand in het project. Het legt vast wat er gedaan is, wat er besloten is en waarom, en wat nog openstaat. Schrijf het in gewone zinnen, voor een lezer die er niet bij was.

Maak er een gewoonte van: vraag Claude aan het einde van elke werksessie de sessienotitie bij te werken. Claude heeft de hele sessie voor zich en schrijft in een minuut wat u twintig minuten zou kosten. Lees daarna wat het schreef — u bent degene die weet of het klopt.

Zet erin	Laat weg
Wat er veranderde, en op welke datum	Alles wat Claude uit de bestanden zelf kan lezen
Wat er besloten werd, en de reden daarvoor	Een verslag van het gesprek, stap voor stap
Wat er geprobeerd en verworpen werd, en waarom	Lof, opvulling, of een samenvatting van wat u al weet
Wat nog openstaat, en de volgende stap	Alles waarvan u niet zeker weet dat het klopt

TIP — De reden telt meer dan de beslissing

“We gebruiken LibreOffice voor de pdf” is op zich weinig waard. “We gebruiken LibreOffice omdat Word vastloopt als het zonder toezicht draait” voorkomt dat de volgende het terugdraait.

VALKUIL — Laat de notitie geen dagboek worden

Een notitie die niemand leest, is hetzelfde als geen notitie. Beperk ze tot beslissingen, redenen en open punten. Verplaats afgewerkte details naar de eigen documenten van het project.

3. Voordat u iets kopieert

Doe eerst deze drie dingen op de bron-pc. Ze duren tien minuten en ze bepalen hoeveel u echt meekrijgt.

1. Zorg dat het project een `CLAUDE.md` heeft. Is er geen, vraag Claude dan `/init` uit te voeren, lees wat het schreef en verbeter het. Dit is het bestand dat het project uitlegt aan een nieuwe Claude.
2. Werk de sessienotitie bij. Vraag Claude ze bij te werken en lees ze daarna. Op de nieuwe pc wordt dit als eerste gelezen, dus het moet kloppen.
3. Vraag Claude zijn geheugen op te ruimen. Open `/memory`, verwijder wat verouderd is en voeg samen wat dubbel staat. Neem geen verouderde feiten mee naar de nieuwe pc.

TIP — Waarom eerst opruimen

Een fout feit dat meereist naar de nieuwe pc is erger dan een ontbrekend feit. U zult het vertrouwen, en het zal fout zijn.

4. Alles verzamelen met twee prompts

Twee prompts doen het werk. Het zijn gewone tekstbestanden, bedoeld om in Claude Code te plakken — geen documenten om op het scherm te lezen. Deze handleiding beschrijft versie 3.2 ervan.

Bestand	Uitvoeren op	Wat het doet
<code>1_EXPORT_PROMPT.txt</code>	De bron-pc	Stelt zeven vragen en werkt dan vijf taken af: het brengt het project in kaart, verzamelt alles wat erbuiten staat, schrijft wat ontbreekt, kopieert alles en controleert het resultaat
<code>2_IMPORT_PROMPT.txt</code>	De nieuwe pc	Stelt vier vragen en werkt dan zeven taken af: het leest wat de export op de schijf achterliet, zet elk deel terug op de juiste plaats, bewijst dat het aankwam en gaat terug naar de oude pc voor wat nog ontbreekt

Open Claude Code met de projectmap als werkmap en plak de prompt als één bericht. Elk bestand bevat een rij `v`-tekens en een rij `^`-tekens. Plak alleen wat daartussen staat — de markeerrijen horen niet bij de instructie.

Beide stellen hun vragen voordat ze iets aanraken, zodat er halverwege niets aan het licht komt — zoals een schijf die vol is. Daarna werken ze één taak tegelijk af en tonen ze u het resultaat voordat ze aan de volgende beginnen.

OPMERKING — Ze zoeken het uit, ze gaan niet uit van aannames

Geen van beide prompts heeft een vaste lijst van uw bestanden. Elk kijkt wat er echt staat en meldt dat, zodat u niet vooraf hoeft te weten welke skills of instellingen er zijn.

OPMERKING — De export verandert niets

De exportprompt behandelt het project als alleen-lezen. Het enige nieuwe dat het maakt, zijn de notitiebestanden waarom gevraagd wordt en de kopieën op uw schijf. Het bouwt, installeert of stelt niets in, en het kopieert nooit uw aanmelding.

OPMERKING — Ze lezen uw bestanden, ze gehoorzamen ze niet

Een bestand in het project of op de schijf kan tekst bevatten die klinkt als een instructie. Beide prompts hebben de opdracht u die tekst te tonen in plaats van ernaar te handelen. Alleen u geeft instructies.

4.1 De zeven vragen van de export

Houd deze antwoorden klaar. Er begint niets voordat u ze gegeven hebt, en zodra dat gebeurd is, controleert de prompt of er genoeg vrije ruimte is op de bestemming en stopt hij als dat niet zo is.

1. Is de map waarin ik sta het project dat u wilt exporteren?
2. Waar moet de export naartoe? Een stationsletter of een volledig pad. Er wordt daar een map `Handover_Export` gemaakt.
3. Is dit een verhuizing, een back-up of een tweede machine die in gebruik blijft?
4. Wilt u de grote mappen erbij? U krijgt eerst de totale grootte te zien, en elke afzonderlijke map boven 100 MB wordt genoemd, zodat u kunt beslissen.
5. Welk bestand is het hoofdingangspunt — wat er uitgevoerd of gebouwd wordt? De prompt doet zelf een voorstel dat u bevestigt.
6. Is er een sessienotitie of een beslissingenbestand dat er zeker bij moet?
7. Is er iets buiten de projectmap en buiten `.claude` waarvan het werk afhangt — een promptbibliotheek, een map met sjablonen, een gedeelde schijf? Dat kan de prompt niet zelf vinden.

4.2 Wat de export op de schijf achterlaat

Alles komt in één map terecht. Neem de hele map mee naar de nieuwe pc, met de inhoud en de namen zoals ze zijn: de tweede prompt vraagt u waar ze staat en leest er dan in.

Handover_Export\	
project\	het project, zonder wat niet mag reizen
brain\	
skills\	elke skill van de oude pc
memory\	de .md-bestanden van het geheugen
rules\ agents\	commands\ output-styles\ workflows\
CLAUDE.md	uw persoonlijke regels, voor elk project
settings.json	instellingen en hooks
HANDOVER_MANIFEST.md	elk bestand, met een controlesom
HANDOVER_BRAIN.md	wat nooit ergens werd opgeschreven
SOURCE_FACTS.md	de oude paden, programma's en variabelen
robocopy_log.txt	bewijs van wat er gekopieerd werd

Wat de export oplevert. Deze map is de overdracht.

De drie notitiebestanden	Waarvoor het dient
HANDOVER_MANIFEST.md	Elk bestand in het project met zijn grootte en een controlesom, zodat u op de nieuwe pc kunt bewijzen dat u de juiste versie hebt en dat er onderweg niets verloren ging
HANDOVER_BRAIN.md	Beslissingen en hun redenen, dingen die mondeling werden afgesproken en nooit opgeschreven, vragen die nooit beantwoord werden. Lees dit bestand zelf — het is het deel dat geen enkele kopie van een map bevat
SOURCE_FACTS.md	Het oude projectpad, de oude naam van de geheugenmap, de programma's, de omgevingsvariabelen en wat samengevoegd moet worden in plaats van vervangen. De importprompt leest dit bestand

OPMERKING — Het is een map, geen zipbestand

Niets hier is gecomprimeerd. De importprompt leest `SOURCE_FACTS.md` rechtstreeks van de schijf, en de controle voordat u de schijf loskoppelt opent gekopieerde bestanden om te bewijzen dat ze volledig aankwamen — beide gaan makkelijker met een gewone map, en een zipbestand dat licht beschadigd is, levert u helemaal niets op. Verstuurt u de export over een netwerk in plaats van hem mee te nemen, zip dan de hele map `Handover_Export` voor de reis en pak hem uit voordat u de import uitvoert.

4.3 Controleer voordat u loskoppelt

De export eindigt met een controle. Lees die echt, in plaats van ze vluchtig te bekijken. Zolang de schijf nog in de bron-pc zit, kunt u herstellen wat ze vindt; bent u eenmaal weg, dan is die schijf misschien de enige kopie die u hebt.

VALKUIL — Een kopie die succes meldt, is geen bewijs

Een kopieerprogramma kan succes melden en toch een afgekapt bestand geschreven hebben. De export opent twee of drie van de gekopieerde bestanden — een `SKILL.md` en een `.md`-bestand uit het geheugen — en bevestigt dat ze volledig zijn, niet leeg en niet afgebroken.

Hij bevestigt ook het volgende voordat u vertrekt — en als er iets ontbreekt, is dit het moment om dat te ontdekken:

- Elk van de drie notitiebestanden is aanwezig, met zijn grootte
- `project\`, `brain\skills\` en `brain\memory\` zijn alle drie aangekomen
- Niets wat niet mag reizen, staat op de schijf — vooral `.credentials.json` niet
- Het aantal bestanden en de totale grootte komen overeen met wat het kopiëren zelf meldde
- Een gewone lijst van alles op de schijf, zodat u de hele oogst in één keer ziet

4.4 De export schrijft wat ontbreekt

Hoofdstuk 3 vraagt u `CLAUDE.md` en de sessienotitie in orde te brengen voordat u begint. Mensen doen dat niet altijd, en een magere sessienotitie reist net zo slecht als helemaal geen.

Daarom klaagt de export niet langer alleen. Zijn derde taak biedt aan te schrijven wat ontbreekt, vanuit de sessie waarin u al zit: een `CLAUDE.md` als die er niet is, en één nieuw, gedateerd item voor de sessienotitie. Hij toont u eerst elke regel en schrijft niets voordat u akkoord geeft.

OPMERKING — Hij voegt alleen toe

Hij mag een bestand aanmaken dat niet bestaat, of iets toevoegen aan een bestand dat wel bestaat. Hij zal nooit een regel die u schreef herschrijven of verwijderen. Zegt u nee, dan laat hij het bestand met rust en gaat hij verder — een leemte die u kent, is geen probleem.

TIP — Dit is het enige dat de moeite waard is, ook als u niet verhuist

Voer de derde taak van de exportprompt uit op een project dat precies blijft waar het is, en u krijgt een `CLAUDE.md` en een sessienotitie die geschreven zijn terwijl het werk nog vers was. Dat is de moeite waard, of u nu verhuist of niet.

5. Wat u moet achterlaten

Sommige dingen mogen nooit gekopieerd worden. Ze zijn privé, of ze worden op de nieuwe pc automatisch opnieuw aangemaakt en kopiëren veroorzaakt fouten.

Niet kopiëren	Waarom
<code>.credentials.json</code>	Uw aanmelding. Meld u op de nieuwe pc opnieuw aan. Zet dit nooit op een usb-stick.
API-sleutels, tokens, wachtwoorden	Nooit als waarde gekopieerd. De export noemt ze en schrijft SECRET - set by hand on the new PC. U stelt ze daar zelf in.
<code>.venv</code> , <code>node_modules</code>	Gebouwd voor één pc. Maak ze opnieuw aan met <code>py -m venv</code> en <code>npm install</code> .
<code>__pycache__</code> , <code>build</code> , <code>dist</code> , <code>_internal</code>	Gegenereerde bestanden. Ze komen vanzelf terug.
<code>cache</code> , <code>shell-snapshots</code> , <code>telemetry</code> , <code>debug</code> , <code>sessions</code> , <code>file-history</code> , <code>history.jsonl</code>	Werkbestanden van het programma zelf. Voor u van geen nut.
<code>_archive</code> , zeer grote <code>.zip</code> -bestanden	Meestal oude momentopnamen van wat u al kopieert.
<code>.git</code>	Groot, en beter terug te zetten door te klonen vanaf uw remote. Lees eerst de waarschuwing hieronder.

VALKUIL — Als het project een git-repository is

De export laat `.git` weg, dus de geschiedenis reist niet mee. Kloon het project op de nieuwe pc in plaats daarvan vanaf zijn remote.

Heeft het project geen remote, dan bestaat de geschiedenis alleen op de oude pc.

Kopieer `.git` dan met de hand voordat u iets wist.

6. Alles terugzetten op de nieuwe pc

De tweede prompt doet dit werk. U bereidt de pc voor, plakt de prompt en beantwoordt de vragen. Daarna werkt de prompt zeven taken af: wat er op deze pc ontbreekt, de projectbestanden (waarbij het hoofdingangspunt wordt vergeleken met de controlesom in het manifest, zodat u weet dat u de juiste versie hebt), de skills en instellingen, het geheugen, de omgevingsvariabelen, een controlelijst, en tot slot wat nog ontbreekt — en u krijgt elk resultaat te zien voordat de volgende taak begint.

1. Sluit de schijf met `Handover_Export` aan.
2. Maak de map aan waarin het project komt. Leeg is prima.
3. Open Claude Code met die map als werkmap. Dit is belangrijk: zo maakt Claude de geheugenmap aan onder de juiste nieuwe naam.
4. Plak `2_IMPORT_PROMPT.txt` als één bericht, zonder de markeerrijen.

TIP — Laat de prompt het priegelwerk doen

Het geheugen terugzetten betekent een map vinden waarvan de naam met het pad veranderde. De instellingen terugzetten betekent twee bestanden samenvoegen in plaats van er één te vervangen. Dat zijn precies de stappen die u beter aan een machine overlaat dan ze zelf te doen aan het einde van een lange dag.

6.1 De vier vragen van de import

De prompt stelt deze vier vragen, leest dan `SOURCE_FACTS.md` en toont u het oude projectpad, de oude naam van de geheugenmap, de programma's en variabelen die u nodig hebt, en alles wat samengevoegd moet worden in plaats van vervangen. Blijkt de doelmap niet leeg te zijn, dan toont de prompt wat erin staat en stopt hij om het u te vragen.

1. Waar staat `Handover_Export`? Een stationsletter of een volledig pad.
2. Is de map waarin ik sta de plaats waar het project moet komen?
3. Is deze pc nieuw, of heeft hij al Claude-skills en -instellingen die het bewaren waard zijn?
4. Moet ik de ontbrekende programma's installeren, of ze alleen opsommen?

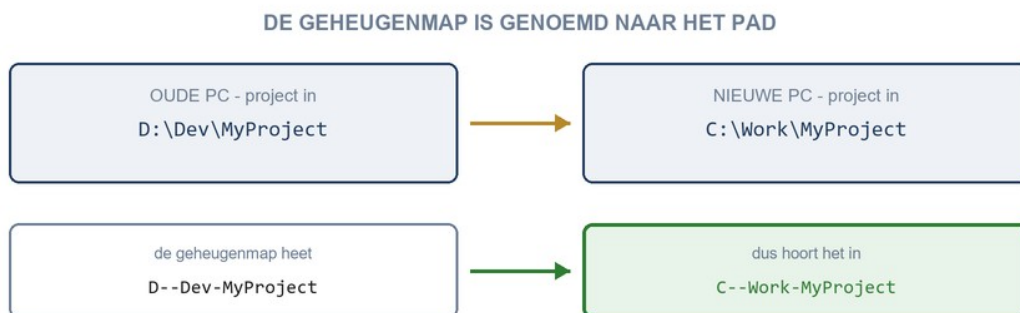
6.2 De prompt voegt samen, hij overschrijft niet

De nieuwe pc kan al een eigen Claude-configuratie hebben. Bestanden in hun geheel vervangen zou de machtigingen en hooks die daar zijn opgebouwd weggooien. De prompt is zo geschreven dat hij dat niet doet.

Wat	Hoe het behandeld wordt
Skills	Mappen, dus ze staan naast elkaar. Een skill met dezelfde naam wordt nooit overschreven: u ziet beide (datums, groottes, wijzigingslogboek) en kiest.
Persoonlijke CLAUDE.md	Eén bestand. Heeft deze pc er geen, dan wordt het gekopieerd. Anders ziet u beide volledig, met een voorstel tot samenvoeging. Niets wordt geschreven zonder uw akkoord.
settings.json	Eén bestand, net zo behandeld: beide versies, de sleutels die verschillen en een samenvoeging ter goedkeuring.
rules\, agents\, commands\, output-styles\, workflows\	Bestand per bestand gekopieerd. Wat hier al bestaat, wordt overgeslagen, en u hoort welke.
Machinegebonden bestanden in een skill	node_modules, __pycache__ en profile.json blijven bewust achter: ze horen bij de oude pc. profile.json zegt wie daar documenten ondertekent.
Plugins	Niet gekopieerd. De prompt somt op welke plugins en marketplaces de oude pc had, en geeft de opdracht om elk ervan hier te installeren. Plugins worden geïnstalleerd, niet gekopieerd.

6.3 De naam van de geheugenmap zal anders zijn

Dit is de stap die het vaakst misgaat. De geheugenmap is genoemd naar het volledige pad van het project. Zet het project ergens anders en de naam verandert.



Kopieer de bestanden naar de NIEUWE naam. De oude wordt nooit meer gelezen.

Kopieer de geheugenbestanden naar de map met de naam van het NIEUWE pad, niet het oude.

De prompt regelt dit in zijn vierde taak. Hij zoekt de map voor het pad van deze pc, bevestigt dat het niet de oude naam uit SOURCE_FACTS.md is, kopieert de .md-bestanden erin, leest daarna MEMORY.md terug en vertelt u hoeveel items er aankwamen. Hij somt ook geheugenitems op die hier nu fout zijn — een oude stationsletter, een oud pad, een notitie over de bron-pc — en laat u beslissen.

Om het met de hand te doen, of om het werk van de prompt te controleren:

1. Start Claude Code één keer in de projectmap; zo ontstaat de map met de juiste naam.
2. Kijk in `C:\Users\<uw naam>\.claude\projects\` en zoek de nieuwe naam.
3. Kopieer de `.md`-bestanden van de oude pc naar de map `memory\` daarin.
4. Start Claude Code opnieuw en vraag: “wat weet je nog over dit project?”

VALKUIL — Kopiëren naar de oude naam doet niets

Een map die naar het oude pad genoemd is, wordt nooit meer gelezen. De bestanden staan daar, zien er correct uit en zijn onzichtbaar voor Claude.

6.4 Programma's en omgevingsvariabelen

Dat de bestanden aangekomen zijn, betekent nog niet dat het project draait. Twee dingen reizen nooit mee: geïnstalleerde programma's en omgevingsvariabelen. Beide moet u met de hand op hun plaats zetten.

De eerste taak van de importprompt meldt welke programma's en pakketten ontbreken en toont voor elk de exacte opdracht. De vijfde taak somt elke omgevingsvariabele van de oude pc op, zegt of ze hier al ingesteld is en op welke waarde, en geeft u de `setx`-opdracht met het pad aangepast voor deze pc.

VALKUIL — Een nieuwe variabele vraagt een nieuwe terminal

`setx` schrijft de variabele, maar een venster dat al open is, zal ze nooit zien. Sluit Claude Code daarna en open het opnieuw.

7. Controleer of het gelukt is

Ga niet uit van aannames. Vraag het, en vergelijk met wat u meebracht.

VALKUIL — Start opnieuw voordat u iets controleert

Het geheugen wordt alleen gelezen wanneer een sessie start. Sluit Claude Code eerst en open het opnieuw, anders antwoordt een volledig correcte configuratie alsof ze zich niets herinnert.

Vraag aan Claude	U zou moeten krijgen
“Wat weet je nog over dit project?”	De feiten van de oude pc, geen leeg antwoord
“Welke skills heb je?”	Dezelfde lijst als op de oude pc
“Welke instructiebestanden heb je geladen?” (<code>/context</code>)	Uw project- en uw persoonlijke <code>CLAUDE.md</code>
Voer de eigen build of test van het project uit	Hetzelfde resultaat als op de oude pc

De importprompt sluit af met vier lijsten: wat er gekopieerd werd, wat er overgeslagen werd en waarom, wat nog uw beslissing vraagt, en wat helemaal niet kon. Lees die voordat u de nieuwe pc vertrouwt.

TIP — Houd de stick tot dit in orde is

Wis de bron-pc of de stick niet voordat elke rij hierboven correct beantwoord is. Tot dan is het de enige kopie die u hebt.

7.1 Als er nog iets ontbreekt

Zie de controlelijst niet als het einde. Kwam een antwoord fout terug, of draait het project niet, dan bent u nog niet klaar — en de oude pc staat er nog, en kan nog antwoorden. De zevende taak van de importprompt maakt daar gebruik van.

1. Ze vertelt u precies wat ze nodig heeft en niet heeft — een bepaald bestand, een skill, een instelling, een feit.
2. Ze splitst dat in tweeën: wat de oude pc u nog kan geven, en wat alleen u hier kunt doen (een programma installeren, een plugin toevoegen, een variabele instellen).
3. Voor de eerste groep schrijft ze een prompt die u in Claude Code op de OUDE pc plakt, met de naam van het project en precies wat ze nodig heeft.
4. U brengt het antwoord terug en de controlelijst draait opnieuw. Doorloop die lus zo vaak als nodig is.

VALKUIL — Daarom wordt er nog niets verwijderd

De hele lus hangt ervan af dat de oude pc er nog is om het te vragen. Bewaar hem, en bewaar de schijf, tot de controlelijst zonder fouten terugkomt. Beide prompts zeggen dat nu in hun laatste regel.

8. Een korte controlelijst

- Projectmap, zonder `.venv`, `node_modules`, `build`, `dist`, `__pycache__`
- De sessienotitie, bijgewerkt op de dag dat u kopieert
- Project-CLAUDE.md en uw notitiebestanden (staan al in de map)
- De hele map `~/.claude/skills/`
- `~/.claude/CLAUDE.md` en `~/.claude/rules/`, als ze bestaan
- `~/.claude/settings.json`
- `~/.claude/agents/`, `commands/`, `output-styles/`, `workflows/`, als ze bestaan
- De `.md`-bestanden uit `~/.claude/projects/<project>/memory/`
- Een geschreven lijst van de geïnstalleerde plugins — de lijst, niet de map
- Een geschreven lijst van omgevingsvariabelen en geïnstalleerde programma's
- `HANDOVER_MANIFEST.md`, `HANDOVER_BRAIN.md` en `SOURCE_FACTS.md`
- `.git`, alleen als het project geen remote heeft om van te klonen
- Nooit: `.credentials.json`

9. Verklarende woordenlijst

Term	Betekenis
Automatisch geheugen	Notities die Claude voor zichzelf schrijft over u en het project.
Sessienotitie	Een bestand in het project waarin u vastlegt wat er besloten werd en waarom. U schrijft het, en het reist mee met de map.
CLAUDE.md	Een bestand dat u schrijft om Claude te vertellen hoe het moet werken. Eén per project, plus een persoonlijk bestand.
Skill	Een map met een beschreven werkwijze, met een <code>SKILL.md</code> erin.
Plugin	Een set skills en opdrachten uit een marketplace. Wordt geïnstalleerd, niet gekopieerd.
Handover_Export	De map die de exportprompt op uw schijf aanmaakt. Ze bevat het project, het brein en drie notitiebestanden.
SOURCE_FACTS.md	Het bestand dat de export schrijft en de import leest: oude paden, programma's en variabelen.
Gebruikersprofiel	Uw eigen map op de pc: <code>C:\Users\<uw naam></code> . Bevat de verborgen map <code>.claude</code> .
Transcript	De opname van één sessie. Wordt na een tijd automatisch verwijderd.
Omgevingsvariabele	Een instelling die in Windows wordt bewaard, niet in een bestand. Ze reist niet mee met een kopie.
Hook	Een opdracht die Claude zelf uitvoert op een vast moment, ingesteld in <code>settings.json</code> .
MCP-server	Een externe tool die een project mag bereiken, vermeld in <code>.mcp.json</code> .

Versiegeschiedenis: Versie 1.0 — 10 september 2026 — Eerste versie, die een overdrachtprompt voor één project vervangt door een algemene methode. Versie 2.0 — 11 september 2026 — In lijn gebracht met versie 3.0 van de twee prompts: de vragen die elk stelt, wat de export op de schijf achterlaat, samenvoegen in plaats van overschrijven, git-geschiedenis, programma's en variabelen. Versie 2.1 — 11 september 2026 — Alles wat de twee prompts behandelen, wordt nu uitgelegd, met de reden om het mee te nemen: de profielmappen die alleen maar opgesomd werden (2.2), de controle voordat u loskoppelt (4.3), en de regel dat de prompts uw bestanden lezen zonder ze te gehoorzamen. Versie 2.2 — 26 oktober 2026 — In lijn gebracht met de prompts versie 3.1: de export schrijft nu een ontbrekende `CLAUDE.md` of sessienotitie in plaats van er alleen over te klagen (4.4), hij vraagt naar mappen die buiten het project staan (4.1), de import kan naar de oude pc teruggaan voor wat ontbreekt (7.1), en er mag niets verwijderd worden voordat dat in orde is. Versie 2.3 — 26 oktober 2026 — Zegt in 4.2 duidelijk dat de export een map is en geen zipbestand, en wanneer zippen toch de juiste keuze is. Versie 2.4 — 27 oktober 2026 — Prompts 3.2: wachtwoorden, sleutels en tokens worden genoemd, nooit op de schijf gezet (hoofdstuk 5). Auteur en MIT-licentie (hoofdstuk 1); Nederlandse en Duitse versie.