



■ INTERNES HANDBUCH · DUG-IT WINPREP

Das komplette Handbuch zu DUGiT_WinPrep.ps1

Wie das Script aufgebaut ist, wie das interaktive Menü bedient wird, was jede der 21 Aufgaben macht, wie der parametergesteuerte Unattended-Lauf funktioniert — und was beim Ausführen unter einem Elevation-Konto zu beachten ist.

Skript DUGiT_WinPrep.ps1 Version 1.15 Stand 2026-09-15

Umfang 21 Aufgaben in 11 Kategorien, 10 Kapitel

01 Überblick & Aufbau	02 Angemeldeter Benutzer vs. Elevation-Konto
03 Interaktiver Modus	04 Aufgabenkatalog
05 Unattended-Modus	06 Alte Anpassungen entfernen
07 Logging, Prüfungen & Kunden-Report	08 Anhang: Ordner & Betrieb
09 Komfortabler Einsatz beim Kunden	10 Windows-Aktivierung reparieren

01 Überblick & Aufbau

Was das Script macht und wie die gut 2.200 Zeilen intern gegliedert sind — nützlich, wenn du selbst mal etwas ergänzen willst.

DUGiT_WinPrep entfernt Bloatware, deaktiviert Telemetrie/Copilot/Recall, härtet Sicherheitseinstellungen, konfiguriert Windows Defender, setzt Energieoptionen, wendet das

DUG-iT Look & Feel an (Taskleiste, Rechtsklick-Menü, Hintergrund, Edge) und installiert TeamViewer — auf einem frischen oder bereits genutzten Windows-11-Rechner beim Kunden. Es läuft entweder **interaktiv** über ein Pfeiltasten-Menü oder **unattended** ohne jede Rückfrage, z. B. per GPO oder Softwareverteilung.

! Ganz am Anfang, noch **vor** jeder Elevation, prüft ein Server-Check per `Win32_OperatingSystem.ProductType`, ob überhaupt ein Windows-11-**Client** läuft. Wird ein Windows Server oder Domain Controller erkannt, bricht das Script sofort mit Exit-Code 2 ab, ohne jede Änderung und ohne UAC-Prompt — das Script ist ausschließlich für Kunden-Clients gemacht.

Die acht Blöcke im Script

1 · Bootstrap	Server-Check (siehe oben), dann Selbst-Elevation (UAC-Prompt falls nötig), reicht alle Parameter beim Neustart mit Adminrechten durch.
2 · Konfiguration	Marke, Pfade (<code>C:\DUGITInstall</code>), Bloatware-Listen (MS-Apps, Drittanbieter, Xbox).
3 · Hilfsfunktionen	Registry-Helfer, <code>Apply-ToUserHive</code> (HKCU für andere Benutzer), Checklisten-UI, Logging.
4 · UI-Funktionen	Logo, Systeminfo-Übersicht, Rahmen-Box fürs Menü.
5 · Alte Anpassungen	Räumt Überbleibsel alter DUG-iT-Scripte und einer alten Lokal-GPO-Vorlage auf (Kapitel 06).
6 · Task-Definitionen	Der eigentliche Kern — 21 Aufgaben, jede mit Kategorie, Standardwert, Aktion und optionaler Undo-Routine (Kapitel 04).
7 · Menü	Pfeiltasten-Checkliste für die Aufgabenauswahl, mit Texteingabe-Fallback für Hosts ohne Pfeiltasten-Unterstützung.
8 · Hauptprogramm	Menü oder Unattended-Zweig, dann Ausführung, state.json speichern, Protokoll ausgeben.

Neue Anpassung ergänzen

Ein neuer Eintrag nach dem Muster `$Tasks['ID'] = @{ Cat=...; Default=...; IsHKCU=...; Name=...; Action={...}; Undo={...} }` im Block „Task-Definitionen“ reicht — Menü, `-ListTasks` und die Unattended-Parameter greifen den automatisch auf. `Undo` weglassen markiert eine Aufgabe als nicht rückgängig machbar (taucht dann beim Deinstallieren nicht auf). `IsHKCU=$true` markiert Aufgaben mit benutzerspezifischen Registry-Änderungen, die auch für andere angemeldete Benutzer übernommen werden (siehe Kapitel 02) — `Action / Undo` dürfen dafür ausschließlich `HKCU:` -Pfade verwenden.

02 Angemeldeter Benutzer vs. Elevation-Konto

Der wichtigste Stolperstein im Praxiseinsatz: Das Script läuft oft unter einem anderen Konto, als am Bildschirm angemeldet ist.

Wird „Als Administrator ausführen“ mit **anderen** Anmeldedaten bestätigt (z. B. einem separaten Admin-Konto, während am Rechner der Praxis-Mitarbeiter „Praxis“ angemeldet ist), zeigen `$env:USERNAME` und `$env:USERPROFILE` weiterhin auf das Admin-Konto — **nicht** auf den Menschen, der tatsächlich vor dem Bildschirm sitzt. Ohne Gegenmaßnahme würden Taskleiste, Rechtsklick-Menü, Hintergrundbild und Edge-Verknüpfung im falschen Profil landen.

WinPrep löst das automatisch: `Get-InteractiveUser` fragt stattdessen `Win32_ComputerSystem.UserName` ab, was zuverlässig den Menschen an der aktiven Konsolensitzung meldet. Der **vollqualifizierte** Name (z. B. `PRAXIS\Praxis`) wird über Windows' eigene Kontoauflösung (`NTAccount.Translate()`) in eine SID übersetzt — nicht über einen Namensvergleich mit Profilordnern, da Ordnername und Kontoname nach einem Profilkonflikt auseinanderlaufen können (Windows hängt dann z. B. `.PCNAME` an den Ordner, das Konto selbst bleibt aber sauber).

! Die SID-Auflösung nutzt bewusst den **vollqualifizierten** Kontonamen (mit Domänen-Präfix), nicht nur den bloßen Kontonamen: auf frisch domänenengebundenen Rechnern kann eine unqualifizierte Namensauflösung fehlschlagen, obwohl der Benutzer eindeutig existiert. Schlägt die Auflösung trotzdem fehl, erscheint jetzt eine gelbe Warnzeile mit der genauen Fehlermeldung, statt den Fehler still zu verschlucken.

● Systeminfo-Ausgabe beim Start

```
User:      ADMIN-KONTO\vinko [Admin]
ACHTUNG: Script laeuft als 'vinko', angemeldet ist aber 'Praxis'.
          HKCU-Anpassungen (Aussehen, Edge-Verknuepfung etc.) werden automatisch
          auf 'Praxis' umgeleitet, nicht auf dieses Elevation-Konto.
```

Praktisch bedeutet das: Alle Aufgaben mit `IsHKCU=$true` (Taskleiste, Explorer, Hintergrund, Edge-Bereinigung, TeamViewer-Downloads-Ordner) werden über `Apply-ToUserHive` gezielt auf das Profil von „Praxis“ angewendet — per `reg load / reg unload` der `NTUSER.DAT`, oder direkt über `HKEY_USERS\<SID>`, falls der Benutzer bereits angemeldet ist. Das gilt für den interaktiven Lauf genauso wie für Unattended. Systemweite (HKLM-)Aufgaben wie RDP aktivieren, Defender konfigurieren oder Bloatware entfernen sind davon nicht betroffen — die gelten ohnehin maschinenweit.

Angemeldeten Benutzer abwählen

Bei Menüpunkt 1/3 zeigt die Checkliste „Betroffene Benutzer“ den angemeldeten Hauptbenutzer jetzt als eigenen, ebenso abwählbaren Eintrag — „<Benutzer> (angemeldeter Hauptbenutzer)“, Standard weiterhin angehakt (bisheriges Verhalten). Nur sichtbar, wenn das Script überhaupt unter einem anderen Konto läuft (`RunningAsOtherAccount`); läuft es unter dem angemeldeten Konto selbst, gibt es diese Unterscheidung gar nicht erst. Für Unattended-Läufe gibt es dafür den Schalter `-SkipInteractiveUser` (Kapitel 05) — analog zu `-SkipOtherUsers`, aber für den Hauptbenutzer statt für zusätzliche Profile.

- i Menüpunkt **2) Änderungen für andere Benutzer** ist etwas anderes: damit trägst du HKCU-Aufgaben gezielt für *weitere* lokale Profile auf derselben Maschine nach (z. B. ein zweiter Praxis-Mitarbeiter mit eigenem Konto), zusätzlich zur automatischen Umleitung auf den gerade angemeldeten Benutzer. Das dabei verwendete `Get-UserHives` listet außerdem nur Profile mit einer tatsächlich ladbaren Registry-Hive (live unter `HKEY_USERS` oder mit vorhandener `NTUSER.DAT`) — das Public-Profil („Öffentlich“) taucht dadurch gar nicht erst in der Auswahl auf, da es keine eigene Benutzer-Hive besitzt.

03 Interaktiver Modus

Der Normalfall bei einem Vor-Ort-Einsatz: Doppelklick oder Start aus der Konsole, dann per Menü führen.

Nach dem Start zeigt das Script Logo, Haftungshinweis (mit Enter bestätigen) und eine Systeminfo-Übersicht (OS-Version, Gerät, Aktivierungsstatus, lokale Benutzer, Geräte-Manager-Probleme, Zeitpunkt des letzten Laufs). Danach folgt das Hauptmenü:

TASTE	AKTION	WIRKUNG
1	Anpassungen anwenden	Standard-Installationslauf, alle Aufgaben nach Standardauswahl.
2	Änderungen für andere Benutzer	Nur HKCU-Aufgaben, gezielt für ausgewählte zusätzliche Profile (siehe Kapitel 02).
3	Anpassungen rückgängig machen	Nur was laut letztem Lauf (state.json) auch wirklich installiert ist, ist vorausgewählt.
4	Alte Anpassungen entfernen	Bereinigt Überbleibsel alter DUG-iT-Scripte/GPO (Kapitel 06).
5	Windows-Aktivierung	Untermenü: Reparaturversuch / Server-Erreichbarkeit prüfen / Key entfernen (Kapitel 10).
Q	Beenden	Ohne etwas auszuführen.

Bedienung der Aufgabenliste

Bei Menüpunkten 1–3 folgt die Aufgaben-Checkliste, gruppiert nach Kategorie, mit Ist-Zustand-Anzeige pro Zeile:

EINGABE	WIRKUNG
↑ / ↓	Zeile für Zeile navigieren.
Leertaste	Aktuelle Zeile an-/abwählen.
Enter	Mit aktueller Auswahl starten.
A / N	Alle Aufgaben an- bzw. abwählen.
L	Kategorie „DUG-iT Aussehen“ komplett umschalten.

Q / Esc Abbruch, ohne etwas auszuführen.

Jede Zeile trägt ein Tag, das den Zustand laut letztem Lauf zeigt: [installiert]

[nicht installiert] [entfernt] [FEHLER]. Beim Rückgängig-Menü wird eine Zeile gelb hervorgehoben, wenn sie zwar installiert ist, aber gerade abgewählt wurde — also bewusst so bleiben soll. Auf Hosts ohne Pfeiltasten-Unterstützung (z. B. der PowerShell-ISE) schaltet das Menü automatisch auf Texteingabe um: Nummern bzw. Bereiche wie 3-7 toggeln, Enter führt aus.

Nach der Aufgabenauswahl fragt das Script noch, ob eine ausführliche Ausgabe (Verbose) gewünscht ist — zeigt dann zusätzliche Detailzeilen (z. B. welche Registry-Werte genau gesetzt wurden) zu jeder Aufgabe.

04 Aufgabenkatalog

Alle 21 Aufgaben mit Standardauswahl (vorausgewählt bei Menüpunkt 1 bzw. bei einem Unattended-Lauf ohne Task-Parameter) und ob eine Rückgängig-Routine existiert. Referenz für Menü, -ListTasks und die Unattended-Parameter in Kapitel 05.

DATENSCHUTZ & TELEMETRIE

TEL01	Telemetrie, Diagnose, Online-Konto-Zwang aus (System)	An	Undo
TEL02	Werbe-ID, Tracking, Sprache, Freihand, Cloud-Inhalte aus	An	Undo

AI / COPILOT / RECALL

AI01	Copilot, Recall, AI Fabric, Edge-AI deaktivieren	An	Undo
-------------	--	----	------

BLOATWARE ENTFERNEN

BW01	Bloatware entfernen (MS, Drittanbieter, Teams Chat, Xbox)	An	Kein Undo
BW04	OneDrive deinstallieren (alle Benutzer)	An	Kein Undo

NETZWERK & SMB

NET01	Netzwerk-Härtung + SMB-Tuning + NIC-Energiesparen aus	An	Undo
NET02	WLAN + Bluetooth deaktivieren (nur Desktop)	Notebook: An	Undo

SICHERHEIT

SEC01	PS-Logging, SmartScreen an, Autoplay/Remote Assistance aus	An	Undo
SEC02	RDP aktivieren + NLA erzwingen, WinRM aktivieren	An	Kein Undo

! **SEC01** deaktiviert Windows Script Host (WSH) nicht mehr — frühere Versionen taten das als Härtung, was `slmgr.vbs` (Windows-Aktivierung, Kapitel 10) komplett lahmlegte, da das über WSH läuft. **SEC01** stellt jetzt aktiv sicher, dass WSH aktiviert **bleibt** — das korrigiert auch Kunden-PCs, auf denen eine ältere WinPrep-Version WSH bereits dauerhaft deaktiviert hatte. Bewusste Priorität: `slmgr` soll jederzeit zuverlässig funktionieren, auch manuell direkt in der Konsole, wichtiger als diese eine Härtungsmaßnahme.

! **SEC02** erkennt jetzt zusätzlich, wenn eine Domänen-GPO „Remoteverbindungen zulassen“ über einen eigenen Policy-Registry-Pfad (`HKLM:\SOFTWARE\Policies\...\Terminal Services`) erzwingt — dieser hat Vorrang vor dem lokalen Wert, den die Aufgabe (und auch der manuelle Schalter in den Windows-Einstellungen) setzt. In dem Fall bleibt RDP trotz Aufgabe und Gruppenmitgliedschaft aus, ganz ohne sichtbaren Fehler; die Aufgabe meldet das jetzt klar in Rot, statt es unkommentiert zu lassen. Nur Diagnose — eine bewusst gesetzte Domänen-Policy wird nicht überschrieben.

WINDOWS DEFENDER KONFIGURATION

DEF01	Defender: Scan-Performance + Standardaktionen	An	Undo
DEF02	Ausnahmen: Dampsoft (Zahnarzt-Software, Laufwerk F:)	Aus	Kein Undo
DEF03	Ausnahmen: Romexis (Dental-Röntgen-Software)	Aus	Kein Undo

WINDOWS UPDATE

WU01

Feature-Updates 365d, Quality-Updates 14d verzögern

An

Undo

ENERGIEOPTIONEN

PWR01

Höchstleistung, Schnellstart aus (Notebook: AC-Abfrage, Akku unangetastet)

An

Undo

DUG-IT AUSSEHEN

LOOK01

DUG-iT Clean Default (Taskleiste, Explorer, Rechtsklick, Desktop)

An

Undo

LOOK02

DUG-iT Clean Default: Edge-Bereinigung + Widgets aus (System)

An

Undo

LOOK03

Versteckte Dateien anzeigen (optional)

Aus

Undo

LOOK04Alle Desktopsymbole außer DUG-iT + Dieser PC/Papierkorb/Eigene
Dateien löschen (optional)

Aus

Kein Undo

! **LOOK01** fragt interaktiv jetzt per J/n nach, ob die Desktop-Symbole angepasst werden sollen (Dieser PC/Papierkorb/Eigene Dateien einblenden, Netzwerk/Systemsteuerung ausblenden) — Standard weiterhin „Ja“ (bisheriges Verhalten), analog zur AC-Abfrage bei **PWR01**. Unattended steuerbar per `-SkipDesktopIcons` (Kapitel 05). Zusätzlich setzt **LOOK01** jetzt `AllAppsViewMode=2`, damit „Alle Apps“ im neuen Startmenü (24H2+) wie gewohnt als Liste statt gruppiert nach Kategorie erscheint.

DUG-IT TOOLS

TOOL01

TeamViewer DUG-iT + Desktop-Verknüpfung

An

Undo

TOOL02

DUG-iT Support-Kontakt in Computerinformationen (überschreibt keine OEM-Daten)

An

Undo

SONSTIGES

SONST01

VDDS_MMI.INI prüfen + Berechtigungen setzen (Dental-Software)

Aus

Kein Undo

05 Unattended-Modus

Für Rollouts per GPO-Anmeldeskript, RMM-Task oder Fernwartung, bei denen niemand vor der Konsole sitzt, um Häkchen zu setzen.

Schnellstart

Der Regelfall — alle Standardaufgaben installieren, ohne jede Rückfrage:

● Windows PowerShell (als Administrator)

```
PS C:\DUGITInstall> .\DUGiT_WinPrep.ps1 -Unattended -ScriptMode INSTALL
```

Um vorab nachzusehen, welche Aufgaben-IDs die aktuell installierte Skriptversion kennt, bevor du eine Parameterzeile schreibst:

● Windows PowerShell (als Administrator)

```
PS C:\DUGITInstall> .\DUGiT_WinPrep.ps1 -ListTasks
```

```
TEL01    An (Standard)  Telemetrie, Diagnose, Online-Konto-Zwang aus (System)  (T
TEL02    An (Standard)  Werbe-ID, Tracking, Sprache, Freihand, Cloud-Inhalte aus
...
```

`-ListTasks` gibt nur den Katalog aus und beendet sich sofort — es wird nichts installiert oder verändert, auch nicht mit `-Unattended` kombiniert, und es ist keine Elevation nötig.

Parameter-Referenz

PARAMETER	WIRKUNG	STANDARD
-Unattended	Schaltet sämtliche interaktiven Punkte ab: Disclaimer-Bestätigung, Enter-Gate, Checklisten, Ja/Nein-Rückfragen. Ohne diesen Schalter bleibt das Skript auch mit gesetzten Task-Parametern interaktiv.	aus
-ScriptMode	Läuft als <code>INSTALL</code> oder <code>DEINSTALL</code> . Bei <code>DEINSTALL</code> laufen aus der Auswahl nur Aufgaben, die eine Undo-Routine besitzen und laut <code>state.json</code> aktuell installiert sind.	INSTALL

PARAMETER	WIRKUNG	STANDARD
-SkipOtherUsers	HKCU-Aufgaben werden nur auf den interaktiv angemeldeten Benutzer angewendet, nicht auf weitere lokale Profile auf derselben Maschine.	aus
-SkipInteractiveUser	HKCU-Aufgaben werden NICHT auf den angemeldeten Hauptbenutzer angewendet — nur relevant, wenn das Script unter einem ANDEREN Konto läuft (Kapitel 02). Interaktiv entspricht das dem Abwählen des Hauptbenutzer-Eintrags in der „Betroffene Benutzer“-Checkliste.	aus
-SkipDesktopIcons	Desktop-Symbole (Dieser PC/Papierkorb/Netzwerk/Systemsteuerung, Aufgabe LOOK01) NICHT anpassen — überlässt es dem Windows-Standard. Interaktiv entspricht das der J/n-Abfrage bei LOOK01 (Kapitel 04).	aus
-DetailedLog	Schreibt zusätzliche Diagnosezeilen ins Transcript-Log — sinnvoll, wenn ein Lauf per Fernzugriff geprüft werden muss, ohne die Konsole live zu sehen.	aus
-NotebookFullPower	Setzt PWR01 (Höchstleistung) auf einem Notebook auch ohne aktive Netzteil-Erkennung, statt die interaktive AC-Abfrage zu überspringen.	aus
-OnlyTasks	Kommagetrennte Liste von Task-IDs. Es laufen ausschließlich diese — die Standardauswahl des Katalogs wird komplett ignoriert.	leer
-ExcludeTasks	Kommagetrennte Liste von Task-IDs, die aus der Standardauswahl entfernt werden. Wird ignoriert, wenn -OnlyTasks gesetzt ist.	leer
-IncludeTasks	Kommagetrennte Liste von Task-IDs, die zusätzlich zur Standardauswahl aktiviert werden — z. B. eine Aufgabe, die standardmäßig aus ist. Wird ignoriert, wenn -OnlyTasks gesetzt ist.	leer

PARAMETER	WIRKUNG	STANDARD
-ListTasks	Gibt den vollständigen Aufgabenkatalog der aktuellen Skriptversion aus und beendet sich, ohne etwas zu installieren.	aus
-UpdateUrl	Überschreibt die eingebaute Update-Check-Adresse (<code>\$script:UpdateCheckUrl</code> , aktuell <code>https://dl.dug-it.de/winprep/VERSION.txt</code>). Ist die dort hinterlegte Version neuer, bricht ein Unattended-Lauf mit Exit-Code 2 ab — siehe Kapitel 09.	eingebaut
-SkipUpdateCheck	Lässt den Update-Check aus, auch wenn eine Adresse konfiguriert ist — z. B. für Testläufe ohne Internet.	aus

Bei `-OnlyTasks` , `-ExcludeTasks` und `-IncludeTasks` sind Kommas, Semikolons und Leerzeichen als Trenner erlaubt, Groß-/Kleinschreibung spielt keine Rolle — `te101,ai01` und `TEL01 AI01` sind gleichwertig.

Wie die Auswahl aufgelöst wird

Die drei Task-Parameter wirken nicht gleichzeitig, sondern in einer festen Reihenfolge:

- 1 **Ist `-OnlyTasks` gesetzt?** Dann läuft ausschließlich diese Liste. Alles andere in diesem Abschnitt entfällt.
- 2 **Sonst: Standardauswahl aus dem Katalog.** Jede Aufgabe startet mit ihrem eigenen Vorgabewert — Spalte „Standard“ in Kapitel 04.
- 3 **`-ExcludeTasks` anwenden.** Genannte IDs werden aus der Standardauswahl entfernt.
- 4 **`-IncludeTasks` anwenden.** Genannte IDs werden zusätzlich aktiviert, auch wenn ihr Standard „aus“ war.

! Eine unbekannte ID in `-OnlyTasks` / `-ExcludeTasks` / `-IncludeTasks` erzeugt nur eine rote Warnzeile in der Konsole — der Lauf bricht dadurch **nicht** ab. Bei Tippfehlern vorher mit `-ListTasks` die aktuell gültigen IDs prüfen.

Exit-Codes

Für RMM-Tools/Aufgabenplanung, die das Ergebnis auswerten wollen, ohne das Transcript-Log zu parsen. Der Bootstrap-Schritt (Kapitel 01) wartet auf den elevierten Kindprozess und spiegelt dessen Code, das gilt also auch beim ursprünglichen, nicht-elevierten Aufruf.

CODE BEDEUTUNG

- | | |
|---|--|
| 0 | Alle ausgewählten Aufgaben erfolgreich — oder legitim nichts zu tun (z. B. <code>DEINSTALL</code> ohne vorherige Installation). |
| 1 | Teilfehler — mindestens eine Aufgabe ist fehlgeschlagen, der Rest lief durch. |
| 2 | Abbruch, nichts ausgeführt — Q/Esc, leere Auswahl, Elevation abgelehnt/fehlgeschlagen, Update-Check hat eine veraltete Version blockiert, oder der Server-Check (Kapitel 01) hat einen Windows Server erkannt. |

Beispiele aus der Praxis

Vollinstallation, Standardumfang

Der Rollout-Regelfall — alle standardmäßig aktiven Aufgaben, still im Hintergrund.

● Windows PowerShell

```
PS> .\DUGiT_WinPrep.ps1 -Unattended -ScriptMode INSTALL
```

Ohne TeamViewer und ohne VDDS-Prüfung

Für Maschinen ohne Fernwartungs-Anforderung oder ohne Dental-Software.

● Windows PowerShell

```
PS> .\DUGiT_WinPrep.ps1 -Unattended -ScriptMode INSTALL -ExcludeTasks "TOOL01,SONST01"
```

Praxis-PC mit Dampsoft: Standard plus Defender-Ausnahme

`DEF02` ist standardmäßig aus (nicht jeder Kunde hat Dampsoft) — hier gezielt dazuschalten.

● Windows PowerShell

```
PS> .\DUGiT_WinPrep.ps1 -Unattended -ScriptMode INSTALL -IncludeTasks "DEF02"
```

Nur Datenschutz & Sicherheit, sonst nichts

Wenn ein Kunde ausdrücklich nur die Härtung will, ohne Bloatware-Entfernung oder Branding.

● Windows PowerShell

```
PS> .\DUGiT_WinPrep.ps1 -Unattended -ScriptMode INSTALL -OnlyTasks "TEL01,TEL02,AIO1"
```

Rückbau vor Rückgabe/Verkauf des Geräts

Deinstalliert alles, was laut `state.json` aktuell installiert ist und eine Undo-Routine hat.

● Windows PowerShell

```
PS> .\DUGiT_WinPrep.ps1 -Unattended -ScriptMode DEINSTALL
```

Einzelplatz-Rechner: nur den angemeldeten Benutzer anfassen, ausführlich loggen

Kombinierbar mit jedem der obigen Muster.

● Windows PowerShell

```
PS> .\DUGiT_WinPrep.ps1 -Unattended -ScriptMode INSTALL -SkipOtherUsers -DetailedLog
```

Nur systemweite Aufgaben, kein Profil anfassen

Wenn HKCU-Änderungen (Taskleiste, Desktop-Symbole, Startmenü) grundsätzlich niemanden betreffen sollen — z. B. auf einem Gerät, das noch nicht dem Endnutzer zugeordnet ist.

● Windows PowerShell

```
PS> .\DUGiT_WinPrep.ps1 -Unattended -ScriptMode INSTALL -SkipOtherUsers -SkipInterac
```

Menüpunkt 4 — räumt Überbleibsel der alten DUG-iT-Skripte (V1/V2-Härtung) und einer alten Lokal-GPO-Härtungsvorlage auf, bevor die neuen Aufgaben aus Kapitel 04 sauber greifen können. Läuft in dieser festen Reihenfolge:

- 1 V2-Schäden entfernen.** Löscht `hlsvchost.exe` / `hard.exe`, korrigiert einen manipulierten `DiagTrack`-Dienstpfad, entfernt die zugehörige Firewall-Blockregel und die per Policy erzwungene SmartScreen-Deaktivierung.
Setzt außerdem den Besitzer von `svchost.exe` zurück auf TrustedInstaller, falls V2s Hardlink-Trick ihn auf „Administratoren“ geändert hatte.
- 2 Härtung zurücksetzen.** LSA-Einstellungen (`RunAsPPL`, `RestrictAnonymous*`), Firewall-Profile auf „nicht konfiguriert“, Kontorichtlinien (`net accounts`), Windows Script Host wieder aktiviert, `LocalAccountTokenFilterPolicy` entfernt, ExecutionPolicy zurückgesetzt, WinRM-Listener bei Bedarf wiederhergestellt.
- 3 Aktivierungsdienste prüfen.** Reaktiviert `spssvc` / `LicenseManager` / `wlidsvc` / `WaaSMedicSvc`, falls deaktiviert, und meldet den aktuellen Aktivierungsstatus.
- 4 Alte lokale Gruppenrichtlinie entfernen.** Sichert `registry.pol` (Machine + User) und `gpt.ini` nach `DUGITInstall\GPO_Backup_<Zeitstempel>`, löscht sie, dann `gpupdate /force`.
Bekannter Konflikt der alten Vorlage: SmartScreen und Defender-Echtzeitschutz waren dort per Policy zwangsdeaktiviert — widerspricht Kapitel 04. Als Sicherheitsnetz werden diese Policy-Werte zusätzlich direkt aus der Registry entfernt, falls gpupdate die Historie nicht sauber zurückrollt.
- 5 Alte Windows-Update-Verzögerung entfernen.** Räumt Werte unter dem falschen, nicht-autoritativen Pfad `PolicyState` auf.
- 6 Alte NIC-Energiesparen-Registry entfernen.** Sucht Netzwerkadapter mit direkt gesetztem `PnPCapabilities=24` und entfernt den Wert.
- 7 Alte SMB-/Cortana-/Store-Policies zurücksetzen.** SMB-Autodisconnect, Cortana-Firewallregeln, Store-Auto-Update-Policy, IE-„Suggested Sites“.

Jeder Schritt schreibt einen kondensierten Log-Eintrag (Anzahl Funde/Reparaturen) statt einer Zeile pro Registry-Wert — für den Überblick reicht das, Details stehen bei aktivem Verbose-Modus zusätzlich im Transcript-Log.

07 Logging, Prüfungen & Kunden-Report

Was nach jedem Lauf in `C:\DUGITInstall` liegt, was vorab geprüft wird, und wie das eigene Branding ohne Kundendaten anzutasten reinkommt.

Transcript-Log

Ab dem allerersten Bootstrap-Schritt läuft `Start-Transcript` mit — das Log fängt also auch dann alles ab, wenn das Script mitten im Lauf abstürzt. Dateiname:

`WinPrep_<JJJJMMTT_hhmmss>.log`. Am Ende steht zusätzlich ein kompaktes Protokoll auf der Konsole: Zeitstempel, Task-ID, Status (`[OK]` / `[!!]`), Ergebnis und ggf. Fehlerdetail pro Zeile.

state.json

Pro Task-ID wird der **Ist-Zustand** gespeichert, nicht nur ob der letzte Lauf erfolgreich war — das macht das Menü (Kapitel 03) und die Deinstall-Vorauswahl erst möglich:

WERT	BEDEUTUNG
------	-----------

INSTALLED	Aufgabe wurde in einem <code>INSTALL</code> -Lauf erfolgreich ausgeführt.
-----------	---

REMOVED	Aufgabe wurde in einem <code>DEINSTALL</code> -Lauf erfolgreich rückgängig gemacht.
---------	---

FEHLER	Letzter Versuch (Install oder Deinstall) ist fehlgeschlagen — unabhängig von der Richtung.
--------	--

Werte aus früheren Läufen, die im aktuellen Lauf gar nicht ausgewählt waren, bleiben unangetastet erhalten — `state.json` wird also gemerged, nicht überschrieben.

Update-Check

Direkt nach der Systeminfo prüft das Script, ob unter `$script:UpdateCheckUrl` (aktuell `https://dl.dug-it.de/winprep/VERSION.txt`, überschreibbar per `-UpdateUrl`) eine neuere Version als die lokal laufende hinterlegt ist. Kein Internet oder die Adresse nicht

erreichbar: nur eine graue Konsolenzeile, der Lauf geht normal weiter — das darf nie blockieren. Ist eine neuere Version verfügbar: interaktiv nur eine Warnung (Enter = trotzdem fortfahren), unattended bricht der Lauf mit Exit-Code 2 ab (Kapitel 05), außer bei `- SkipUpdateCheck` . Lädt **nicht** automatisch nach — siehe Kapitel 09 für den eigentlichen Update-Weg.

Vorab-Prüfung

Direkt danach prüft `Get-PreflightWarnings` still im Hintergrund: freier Speicherplatz auf `c:` , ausstehender Neustart (Component-Based-Servicing/Windows-Update-Marker), laufendes Windows-Servicing (`TrustedInstaller`). Reine Warnung, kein Abbruch — taucht in der Konsole und im Kunden-Report auf, landet aber bewusst **nicht** in `state.json` .

Kunden-Report

Am Ende jedes Laufs entsteht zusätzlich zum Transcript-Log ein kurzer HTML-Bericht (`WinPrep_Report_<JJJJMMTT_hhmmss>.html` in `C:\DUGITInstall`): Ergebnis-Zusammenfassung, Vorab-Prüfung-Hinweise falls vorhanden, und eine Tabelle aller ausgeführten Aufgaben mit Status. Zum Dalassen beim Kunden oder für die eigene Doku — unabhängig vom Transcript-Log, das eher fürs Troubleshooting gedacht ist.

Eigenes Hintergrundbild

Datei `Hintergrund.jpg` (oder `.png` / `.bmp`) unter `C:\DUGITInstall` ablegen — wird von den Aussehen-Aufgaben automatisch für Desktop **und** An-/Abmeldebildschirm übernommen. Liegt keine Datei dort, bleibt es bei Windows-Blau (Desktop) und Windows Spotlight (Anmeldung). Keine weitere Konfiguration nötig — die Datei wird bei jedem Lauf automatisch erkannt.

08 Anhang: Ordner & Betrieb

Kurzreferenz für den Ordner `C:\DUGITInstall` und ein paar Betriebs-Eigenheiten, die beim Troubleshooting relevant werden können.

DATEI / MUSTER	INHALT
----------------	--------

<code>DUGiT_WinPrep.ps1</code>	Das Script selbst — erwarteter Ablageort für den lokalen Aufruf.
--------------------------------	--

WinPrep_*.log	Ein vollständiges Transcript pro Lauf, auch bei Absturz.
state.json	Ist-Zustand pro Task-ID, siehe Kapitel 07.
Hintergrund.*	Optionales eigenes Hintergrundbild, siehe Kapitel 07.
GPO_Backup_*	Sicherung der alten <code>registry.pol</code> / <code>gpt.ini</code> , angelegt von Kapitel 06 vor dem Löschen.

Berechtigungen auf DUGITInstall

Bei jedem Lauf setzt das Script `icacls ... /grant *S-1-5-32-545:(OI)(CI)(M)` auf den Ordner — das öffnet Lesen/Schreiben/Löschen für die Gruppe „Benutzer“ (nicht nur Administratoren), vererbt sich auch auf Dateien, die erst später dort landen, und ist idempotent. Nötig, weil z. B. eine vom Elevation-Konto heruntergeladene TeamViewer-Datei sonst vom eigentlichen, nicht-administrativen Benutzer nicht geöffnet werden könnte.

Selbst-Elevation

Läuft das Script ohne Adminrechte, startet es sich selbst über `Start-Process powershell.exe -Verb RunAs` neu und reicht dabei alle gesetzten Parameter (inklusive der Unattended-Task-Auswahl) als Kommandozeile durch. Der ursprüngliche, nicht-elevierte Aufruf **wartet** auf den elevierten Kindprozess (`-Wait -PassThru`) und beendet sich dann mit dessen Exit-Code (Kapitel 05) — wichtig für den Ein-Zeiler-Einsatz in Kapitel 09, sonst würde ein aufrufendes RMM-Tool immer nur den Exit-Code dieses Bootstrap-Schritts sehen. – `ListTasks` braucht keine Elevation und wird deshalb nicht mit durchgereicht, sondern direkt nach dem Neustart mit Adminrechten ausgewertet, sobald der Aufgabenkatalog geladen ist.

Mark-of-the-Web

Wurde die Datei per Netzwerk/Download kopiert, trägt sie unter Umständen ein Mark-of-the-Web (Zone.Identifier), das „Rechtsklick > Mit PowerShell ausführen“ inkonsistent blockieren kann. Das Script entsperrt sich deshalb beim Start selbst über `Unblock-File` .

Wie das Script tatsächlich auf einen Kunden-PC kommt — vom Ein-Zeiler für den Vor-Ort-Termin bis zur unbeaufsichtigten Verteilung.

Alles hier basiert auf **dl.dug-it.de** — einem internen Download-Server (mehrere Projekte, WinPrep liegt unter `/winprep/`), der bei jedem `git push` ins WinPrep-Repo spätestens 10 Minuten später automatisch die aktuelle Version ausliefert. Kein manuelles Kopieren von USB-Sticks mehr nötig, sofern der Kunden-PC Internetzugang hat.

Der Ein-Zeiler (Regelfall für einen Vor-Ort-Termin)

PowerShell **als Administrator** öffnen (wichtig — siehe Hinweis unten), dann:

```
● Windows PowerShell (als Administrator)
```

```
PS> iex(irm https://dl.dug-it.de/wp.ps1)
```

`wp.ps1` ist der generische DUG-iT-Tool-Launcher (`DUGiT_Tools_Launcher.ps1`) — zeigt bei mehreren verfügbaren Tools eine nummerierte Auswahl, aktuell mit nur einem Eintrag (WinPrep) startet er ohne Rückfrage direkt durch. Er lädt dafür

`DUGiT_WinPrep_Bootstrap.ps1` nach, der wiederum die aktuelle `DUGiT_WinPrep.ps1` nach `C:\DUGITInstall` herunterlädt und startet — das normale interaktive Menü (Kapitel 03) erscheint, als hätte man die Datei lokal dabei gehabt. Weitere DUG-iT-Tools kommen künftig als zusätzlicher Menüpunkt im selben Launcher dazu, ohne dass sich diese URL ändert. Bewusst ohne Pipe (`|`) geschrieben — auf deutschen Tastaturen braucht `|` AltGr+7, (`/`) nur Shift+8/9. Die klassische Pipe-Form funktioniert genauso: `irm https://dl.dug-it.de/wp.ps1 | iex`.

! `https://` davor ist Pflicht, nicht nur Stil: ohne Schema geht `Invoke-WebRequest` erst auf Port 80, Caddy leitet von dort per **308 (Permanent Redirect)** auf `https` weiter — und Windows PowerShell 5.1 (altes .NET-Framework-`HttpWebRequest`, nicht das `HttpClient` aus PS7) folgt 308-Redirects nicht zuverlässig, das endet in einer `WebException`. Mit `https://` davor tritt das Problem gar nicht erst auf.

! Diese Kurzform funktioniert außerdem nur zuverlässig, wenn PowerShell schon **als Administrator** läuft — per `iex` ausgeführter Code hat keinen eigenen Dateipfad, wodurch die Selbst-Elevation von `DUGiT_WinPrep.ps1` (Kapitel 08) fehlschlagen kann. `wp.ps1` selbst braucht dafür keine Elevation, nur der Weiterstart danach.

Mit Parametern (Unattended, Task-Auswahl, ...)

`iex(irm ...)` kann keine Parameter entgegennehmen — dafür den Bootstrap erst als Datei speichern, dann wie gewohnt mit Parametern aufrufen (identische Parameter wie `DUGiT_WinPrep.ps1` selbst, siehe Kapitel 05):

● Windows PowerShell (als Administrator)

```
PS> iwr https://dl.dug-it.de/wp.ps1 -OutFile boot.ps1
PS> .\boot.ps1 -Unattended -ExcludeTasks "TOOL01,SONST01"
```

Ohne Internet auf dem Kunden-PC

Der Ein-Zeiler braucht Internetzugang auf dem Zielgerät. Ohne Internet bleibt der klassische Weg: `DUGiT_WinPrep.ps1` selbst (z. B. per USB-Stick) nach `C:\DUGITInstall` kopieren und direkt starten — der Update-Check (Kapitel 07) schlägt dann einfach lautlos fehl und blockiert nichts.

Handbuch unterwegs

Dieses Handbuch liegt ebenfalls auf dl.dug-it.de, zum Nachschlagen ohne lokale Kopie:

- dl.dug-it.de/winprep/DUGiT_WinPrep_Handbuch.html
- dl.dug-it.de/winprep/DUGiT_WinPrep_Handbuch.pdf

10 Windows-Aktivierung reparieren

Menüpunkt 5 — ein Untermenü mit drei einzeln startbaren, unterschiedlich riskanten Schritten statt einem einzigen automatischen Ablauf. Grundprinzip: der Reparaturversuch darf niemals mehr kaputt machen als reparieren.

PUNKT	WIRKUNG
-------	---------

1	Reparaturversuch — komplett zerstörungsfrei, siehe unten.
---	---

2	Server-/Firewall-Erreichbarkeit prüfen — reine Diagnose, verändert nichts.
---	--

- 3 Hinterlegten Key entfernen (Reset) — bewusst destruktiv, siehe unten.
- 4 Windows Script Host sicherstellen — minimaler Einzelzweck-Punkt, siehe unten.

1) Reparaturversuch

Zeigt zuerst eine Übersicht beider möglichen Key-Quellen: den **lokal installierten** Key (Edition, Status, sowie der von Windows selbst nur auf 5 Zeichen begrenzte Teil-Key — keine Einschränkung dieses Scripts) und einen eventuell im **BIOS/UEFI** hinterlegten OEM-Key (ACPI/MSDM-Tabelle, häufig bei Fertig-PCs), diesen **vollständig**, da er ohnehin für die OEM-Wiederherstellung gedacht und kein Geheimnis ist.

Beispielausgabe

```
-- Lokal installierter Key --
Edition:  Windows 11 Pro
Teil-Key:  T83GX   (Windows zeigt fuer einen installierten Key nie mehr als die le
Status:    Unlizenziert
HINWEIS:  Das ist Microsofts oeffentlicher Platzhalter-Key (Windows 10/11 Pro) -
kein echter Kauf-/OEM-Key. Damit aktiviert Windows NIE, ausser ueber einen eigen
KMS-Host im Netzwerk - es wird der echte Lizenzkey benoetigt.
-- BIOS/UEFI-OEM-Key --
AAAAA-BBBBBB-CCCCC-DDDDD-EEEEEE
```

! Der installierte Teil-Key wird gegen Microsofts öffentlich dokumentierte **Generic/KMS-Client-Setup-Keys** abgeglichen (alle Windows-10/11-Client-Editionen, Quelle: learn.microsoft.com/windows-server/get-started/kms-client-activation-keys). Ein Treffer bedeutet: installiert ist nur Windows' eigener Platzhalter, kein echter Kauf-Key — der aktiviert grundsätzlich nie ohne eigenen KMS-Host im Netzwerk.

Danach folgt der eigentliche Ablauf:

- 1 **Bereits aktiviert?** Dann sofort Abbruch mit grüner Meldung — keine weiteren Schritte.
- 2 **Online-Aktivierung versuchen** (`slmgr /ato`). Nutzt ausschließlich den bereits installierten Key. Löscht oder ersetzt an dieser Stelle nichts.

3

Nur falls danach immer noch überhaupt kein Key installiert ist (z. B. Fehlercode `0xC004F213` „kein Product Key gefunden“): Der BIOS/OEM-Key wird zum Einspielen angeboten — ausschließlich nach expliziter J/n-Bestätigung, und ausschließlich in diesem „gar kein Key da“-Fall. Ist bereits ein Key installiert (und sei es nur der Generic/KMS-Client-Platzhalter), wird dieser Schritt komplett übersprungen — ein vorhandener Key wird nie automatisch überschrieben.

4

Bleibt die Aktivierung erfolglos: Angebot, den offiziellen Windows-Aktivierungs-Troubleshooter zu öffnen (`ms-settings:activation`) — Microsofts eigene geführte Problembehandlung, u. a. für Fälle mit digitaler Lizenz nach Hardwarewechsel via Microsoft-Konto-Verknüpfung.

Exakte Fehlermeldung statt „nicht aktiviert“

Erfolg/Misserfolg wird zuverlässig, sprachunabhängig per WMI-Status geprüft (`Get-WindowsActivationInfo`) — ein reiner Textvergleich wäre bei deutscher vs. englischer Windows-Installation fragil. Zusätzlich fängt `Invoke-Slmgr` die komplette Konsolenausgabe von `slmgr.vbs` ab (statt sie wie in früheren Versionen nach `Out-Null` zu schicken) und zeigt sie direkt an — bei jedem der drei `slmgr`-Aufrufe (`/ato`, `/ipk`, `/upk`), egal ob erfolgreich oder nicht. Landet zusätzlich im Detail-Feld des Protokolls, taucht also auch im Kunden-Report auf.

! `slmgr.vbs` läuft über `cscript.exe` (Windows Script Host) — frühere WinPrep-Versionen deaktivierten WSH über `SEC01` (Kapitel 04) standardmäßig als Härtung, was `cscript` komplett lahmlegte. Muss-Anforderung: `slmgr` soll jederzeit zuverlässig funktionieren, auch bei einem direkten manuellen Aufruf. Deshalb hält `SEC01` WSH jetzt selbst dauerhaft aktiviert (korrigiert auch ältere, bereits WSH-deaktivierte Kunden-PCs); `Invoke-Slmgr` prüft zusätzlich vor jedem eigenen Aufruf und aktiviert WSH bei Bedarf ebenfalls dauerhaft — kein Zurücksetzen danach mehr, z. B. falls `SEC01` gezielt ausgeschlossen wurde oder eine Domänen-GPO WSH unabhängig davon deaktiviert.

● Beispiel: kein lokaler Key → BIOS-Key gefunden & eingespielt

```
Online-Aktivierung versuchen (slmgr /ato) ...
```

```
-> Weiterhin nicht aktiviert.
```

```
Genaue slmgr-Meldung: Aktivierungsfehler: 0xC004F213 Es wurde kein Product Key fuer die Aktivierung von Windows gefunden. Verwenden Sie "slmgr.vbs /ipk" zur Eingabe eines gueltigen Product Keys.
```

```
Kein Product Key installiert - suche nach einem im BIOS/UEFI hinterlegten  
OEM-Key (haeufig bei Fertig-PCs vorhanden) ...
```

```
-> BIOS/OEM-Key gefunden: W269N-WFGWX-YVC9B-4J6C9-T83GX
```

```
Diesen Key einspielen und Aktivierung erneut versuchen? (J/n):
```

```
slmgr-Meldung (Key einspielen): Product Key wurde erfolgreich installiert.
```

```
-> Aktivierung mit BIOS-Key erfolgreich.
```

```
Genaue slmgr-Meldung: Produkt wurde erfolgreich aktiviert.
```

Grobe Einordnung des Fehlercodes

`Get-ActivationErrorCategory` sucht in der `slmgr`-Ausgabe nach einem Fehlercode (`0x.....`) und ordnet ihn einer von acht groben Kategorien zu, jeweils mit kurzem Hinweis zum nächsten Schritt — erscheint direkt unter der genauen Meldung, nur bei Fehlschlag. Quelle: Microsofts eigene Support-Dokumentation. Kategorien bewusst nach Relevanz für DUG-iT-Kunden-PCs gewichtet (Einzelplatz-Retail/OEM/Digital-Lizenz, keine Firmen-Volumenlizenz) — MAK/KMS-Host-spezifische Codes sind deshalb nur als eine Sammelkategorie drin.

KATEGORIE	BEISPIEL-CODES	HINWEIS
Kein Key installiert	<code>0xC004F213</code>	BIOS/OEM-Key-Suche oben nutzen oder Original-Kaufkey eingeben.
Netzwerk/Firewall	<code>0x8007232B</code> , <code>0xC004F074</code> , <code>0x800706BA</code> u. a.	Aktivierungsserver nicht erreichbar — Menüpunkt 2 „Server-/Firewall-Erreichbarkeit prüfen" nutzen.
Key ungültig / falsche Edition	<code>0xC004F050</code> , <code>0xC004C001</code>	Tippfehler oder Key passt nicht zur installierten Edition (z. B. Pro-Key auf Home).
Key gesperrt	<code>0xC004C003</code> , <code>0xC004F051</code>	Microsoft hat den Key gesperrt (mehrfach verwendet/gestohlen gemeldet) — neuen Key besorgen.

KATEGORIE	BEISPIEL-CODES	HINWEIS
Nicht als Original erkannt	0x8004FE21 , 0xC004F064 , 0xC004F065	Manipulierte Systemdateien/unlizenzierte Sprachpakete — im Zweifel Neuinstallation erwägen.
Digitale Lizenz nach Hardware-Wechsel	0x803FA067 , 0x803F7001 , 0x803F8001	Nach Mainboard-Tausch o. ä. — Troubleshooter mit dem ursprünglichen Microsoft-Konto nutzen.
Keine Berechtigung	0x80070005	Ohne Adminrechte — sollte durch die Selbst-Elevation dieses Scripts eigentlich nie auftreten.
Volumenlizenz (KMS/MAK)	0xC004C020 , 0xC004F039 u. a.	Typisch für Firmen-Volumenlizenzen — bei Einzelplatz-PCs unwahrscheinlich, deutet auf einen falsch installierten KMS-Client-Key hin.

Unbekannte, nicht in dieser Liste enthaltene Codes fallen auf einen generischen Hinweis mit Link zur [Microsoft-Dokumentation](#) zurück, statt stillschweigend nichts anzuzeigen.

2) Server-/Firewall-Erreichbarkeit prüfen

Reine Diagnose, unabhängig vom Reparaturversuch einzeln startbar (z. B. um gegenüber Kunde/IT ein Firewall-Problem konkret zu belegen). Prüft per `Test-NetConnection` auf Port 443, ob die für die Online-Aktivierung nötigen Microsoft-Server erreichbar sind:

- `activation.sls.microsoft.com`
- `validation.sls.microsoft.com`
- `crl.microsoft.com` (Zertifikatsprüfung — wird gerne übersehen, blockiert aber genauso die Aktivierung)

Ist mindestens einer nicht erreichbar, deutet das auf Firewall/Proxy/DNS als eigentliche Ursache für eine fehlgeschlagene Online-Aktivierung hin — unabhängig von einem eventuell vorhandenen, aber ungültigen Key.

3) Hinterlegten Key entfernen (Reset)

! Bewusst destruktiv und komplett getrennt vom Reparaturversuch: entfernt per `slmgr /upk` den aktuell installierten Product Key. Windows ist danach **garantiert** deaktiviert, bis von Hand ein gültiger Key neu eingetragen wird (über den Reparaturversuch oder `slmgr /ipk <Key>`). Nur sinnvoll, wenn der richtige Key griffbereit ist.

Verlangt zur Bestätigung die exakte, **großgeschriebene** Eingabe `ENTFERNEN` — ein einfaches J/n reicht bewusst nicht, „entfernen“ kleingeschrieben ebenfalls nicht. Jede andere Eingabe bricht ohne jede Änderung ab. Wird nirgends automatisch aufgerufen, auch nicht vom Reparaturversuch. Auch hier zeigt `Invoke-Slmgr` die genaue Bestätigungsmeldung von `slmgr /upk` an (typischerweise „Product Key wurde erfolgreich deinstalliert.“).

4) Windows Script Host sicherstellen

Minimaler Einzelzweck-Punkt, unabhängig von `SEC01` (Kapitel 04) und vom vollen Reparaturversuch einzeln startbar: zeigt nur den aktuellen WSH-Status (aktiviert / deaktiviert / nicht gesetzt — Windows-Standard ist dabei aktiviert) und aktiviert bei Bedarf, ohne gleich eine komplette Aktivierung oder Aufgabe mitzustarten. Sinnvoll z. B. direkt nach einem manuellen `slmgr`-Aufruf, der noch fehlschlägt, oder um den Status schnell zu prüfen, ohne extra durch Punkt 1 zu gehen.

DUG-iT WinPrep · internes Handbuch

state.json und Transcript-Log liegen nach jedem Lauf in DUGITInstall