INTERNES HANDBUCH · DUG-IT DOWNLOADS-CLEANUP

Das komplette Handbuch zu Downloads-Cleanup

Wie ein einziges Skript den Downloads-Ordner (oder beliebige andere Ordner) bei jeder Anmeldung automatisch leert, wie Installation und Menü funktionieren, was die Löschlogik schützt — und wie Downloads-Cleanup genau wie WinPrep und NetMonitor über dl.dug-it.de beim Kunden zum Einsatz kommt.

Skript Cleanup-Downloads.ps1 Version 1.3 Stand 2026-10-07 Umfang 1 Skript, 8 Kapitel

01 Überblick & Architektur	02 Installation
03 Bedienung & Hauptmenü	04 Löschlogik & Schutz
05 Laufzeit-Artefakte	06 Deinstallation
07 Komfortabler Einsatz beim Kunden	08 Anhang: Ordner, Troubleshooting & Release

01 Überblick & Architektur

Was Downloads-Cleanup macht, warum es bei der Anmeldung statt beim Herunterfahren läuft und woraus es besteht.

Downloads-Cleanup leert einen oder mehrere Ordner — standardmäßig den Downloads-Ordner des jeweiligen Benutzers — automatisch bei **jeder Benutzeranmeldung**. Es ersetzt ein früheres GPO-Shutdown-Skript, das unzuverlässig war: Windows-Schnellstart (Herunterfahren ist dann kein echtes Herunterfahren), Timeouts beim Herunterfahren,

versteckte Dateien ohne `-Force`, falsche Löschreihenfolge und ein hart codierter Benutzerpfad. Zielumgebung ist Windows 10/11 mit Windows PowerShell 5.1.

Die Bausteine

Cleanup-Downloads.ps1	Das einzige Skript: Menü (läuft ohne Adminrechte), Installation/Deinstallation (öffnet dafür selbst ein erhöhtes Fenster per UAC) und die eigentliche Bereinigung (`-Run`, nur für die geplante Aufgabe).
Geplante Aufgabe	`DUG-iT Downloads-Cleanup`: Trigger bei Anmeldung (1 Minute Verzögerung), läuft im Kontext des jeweils angemeldeten Benutzers, versteckt.
Aufgabe (manuell)	`DUG-iT Downloads-Cleanup (manuell)`: ohne Trigger, nur zum Starten per Rechtsklick → Ausführen in der Aufgabenplanung — läuft als der angemeldete Benutzer selbst.
Cleanup-Downloads.json	Einstellungen im Installationsordner: `Folders` (zu leerende Ordner) und `Days` (Mindestalter in Tagen, 0 = alles).
Log pro Benutzer	`%LOCALAPPDATA%\DUG-iT\Downloads-Cleanup.log` — jeder Lauf protokolliert Start, Ergebnis je Ordner und Fehler.

Datenfluss

Die Installation (Kapitel 02) legt das Skript und die `Cleanup-Downloads.json` im Installationsordner ab und registriert die Aufgabe. Bei jeder Anmeldung startet die Aufgabe nach einer Minute `Cleanup-Downloads.ps1 -Run` im Kontext des Benutzers: Das Skript liest die Einstellungen, löst Platzhalter wie `{Downloads}` für genau diesen Benutzer auf, löscht nach den Regeln aus Kapitel 04 und schreibt das Log.

i **Warum bei der Anmeldung und nicht beim Herunterfahren?** Bei der Anmeldung läuft garantiert eine frische Benutzersitzung, die Dateien der letzten Sitzung liegen noch da, und es gibt kein Zeitlimit eines Herunterfahrvorgangs, das den Lauf abbrechen könnte. Außerdem läuft die Aufgabe im Kontext des angemeldeten Benutzers — es wird nur gelöscht, was dieser Benutzer selbst löschen darf.

i Genau wie WinPrep und NetMonitor läuft Downloads-Cleanup sowohl direkt lokal als auch über den gemeinsamen DUG-iT-Tool-Launcher (`dl.dug-it.de/wp.ps1` , Kapitel 07) — gleiches

Deployment-Prinzip, gleicher Server, gleiche Optik.

02 Installation

Menüpunkt 1 — ein kurzer Assistent in einem erhöhten Fenster; bei erneuter Installation werden die aktuell installierten Werte als Vorauswahl angeboten (Enter übernimmt jeweils den Wert in Klammern).

1. **Installationspfad:** Standard ist `C:\Program Files\DUG-iT\Downloads-Cleanup`, anpassbar — bewusst unter Program Files, weil das Skript bei jeder Anmeldung im Kontext des jeweiligen Benutzers (auch von Administratoren) läuft und dort nur Administratoren schreiben dürfen. Bei erneuter Installation steht der aktuell installierte Pfad als Vorauswahl da; eine Installation aus V1.0–V1.2 unter `C:\DUGITInstall` wird dabei automatisch hierher verschoben.
2. **Ordner wählen:** Eine nummerierte Liste bietet den **Downloads-Ordner des angemeldeten Benutzers** als Option `[1]` an (mit aufgelöstem Pfad), dazu bisher konfigurierte weitere Ordner; `*` markiert die Vorauswahl. Eingabe: Nummern mit Komma (`1,2`), `A` für einen weiteren, selbst eingegebenen Ordner, Enter übernimmt die Vorauswahl.
   Mehrere Ordner sind möglich. Platzhalter und Umgebungsvariablen siehe Callout unten.
3. **Mindestalter:** "Ab wie vielen Tagen löschen" — Standard 0 (alles löschen) bzw. der aktuell installierte Wert, erlaubt sind 0 bis 3650.
4. **Zusammenfassung & Bestätigung:** Installationspfad, aufgelöste Ordner und Mindestalter werden angezeigt; erst nach `j` wird installiert.

Danach legt die Installation das Skript und die Einstellungsdatei im Installationsordner ab (Standard-NTFS-Rechte bleiben **unverändert**, es werden keine Rechte eingeschränkt), räumt eine ältere Installation an einem anderen Ort auf und registriert die geplante Aufgabe.

! Die Installation braucht **Administratorrechte** (geplante Aufgabe, Installationsordner). Das Menü selbst läuft **ohne** Adminrechte (siehe Kapitel 03); nur Installieren und Deinstallieren öffnen per UAC-Abfrage ein erhöhtes Fenster — wer keine Adminrechte hat, kann alles andere trotzdem nutzen. Die *Aufgabe selbst* läuft danach bewusst ohne Adminrechte: Principal ist die Gruppe

`BUILTIN\Users` (per SID `S-1-5-32-545`, sprachunabhängig) mit eingeschränktem Ausführungslevel.

i **Platzhalter in Ordnerpfaden:** `{Downloads}` ist der echte Downloads-Ordner des jeweiligen Benutzers (`shell:Downloads` — berücksichtigt Umleitungen, z. B. nach OneDrive) und wird zur Laufzeit pro Benutzer aufgelöst. Umgebungsvariablen wie `%USERPROFILE%` oder `%USERNAME%` sind erlaubt. In Vorauswahl und Zusammenfassung zeigt das Skript den aufgelösten Pfad des *am PC angemeldeten Benutzers* — bei jedem anderen Benutzer wird zur Anmeldung dessen eigener Pfad verwendet.

i **Angemeldeter Benutzer statt ausführendes Konto:** Läuft das Skript unter einem anderen Konto als dem des Benutzers am PC (z. B. Admin-Konto per UAC-Abfrage oder RDP), beziehen sich `{Downloads}` , `%USERPROFILE%` & Co., der Testlauf, "Jetzt bereinigen" und das Log trotzdem auf den **am PC angemeldeten Benutzer** — nicht auf das Admin-Konto. Der Name steht im Menükopf (`Angemeldet`) und in der Zusammenfassung. Ist niemand am PC angemeldet, gilt das ausführende Konto.

03 Bedienung & Hauptmenü

Rechtsklick → "Mit PowerShell ausführen" (oder über den Launcher) — **ohne Adminrechte**. Genau wie bei WinPrep und NetMonitor bleibt das Menü nach einer Aktion offen, bis `Q` gedrückt wird.

Cleanup-Downloads.ps1 — Hauptmenü

```
+------------------------------------------------------------------+
| Downloads-Cleanup - Status                                       |
+------------------------------------------------------------------+

Angemeldet     PRAXIS\praxis
Rechte         Standardbenutzer (Installieren/Deinstallieren fragt per UAC nach)
Aufgabe        installiert (DUG-iT Downloads-Cleanup)
Installiert in C:\Program Files\DUG-iT\Downloads-Cleanup
Letzter Lauf   07.10.2026 08:01 (laut Log)
Ordner
  - {Downloads}  ->  C:\Users\praxis\Downloads
Mindestalter   0 Tag(e)  (0 = alles loeschen)
```

```
+------------------------------------------------------------------------+
| Downloads-Cleanup V1.3                                                 |
+------------------------------------------------------------------------+

-- Einrichtung (Adminrechte per UAC) --
 1) Installieren / Einstellungen aendern
 2) Deinstallieren

-- Bereinigung (ohne Adminrechte) --
 3) Testlauf (zeigt nur an, was geloescht wuerde)
 4) Jetzt bereinigen (loescht wirklich, nach Rueckfrage)

-- Diagnose --
 5) Log anzeigen
 6) Einstellungen anzeigen

 Q) Beenden
```

TASTE	AKTION	WIRKUNG
1	Installieren / Einstellungen ändern	Startet den Assistenten aus Kapitel 02 in einem erhöhten Fenster (UAC) — auch zum nachträglichen Ändern von Ordnern, Mindestalter oder Installationspfad.
2	Deinstallieren	Entfernt Aufgabe und Dateien (erhöhtes Fenster per UAC), siehe Kapitel 06.
3	Testlauf	Läuft die komplette Löschlogik durch, zeigt aber nur an (`[TEST] wuerde loeschen`), was gelöscht würde — es wird nichts verändert. Ohne Adminrechte, bezogen auf den angemeldeten Benutzer.
4	Jetzt bereinigen	Führt die Bereinigung sofort aus — **wirklich**, nach Anzeige von Benutzer, Ordnern und Mindestalter und einer Rückfrage (Standard: Nein). Ohne Adminrechte, nur wenn installiert.
5	Log anzeigen	Die letzten 40 Zeilen des Logs *des angemeldeten Benutzers*.

TASTE	AKTION	WIRKUNG
6	Einstellungen anzeigen	Rohinhalt der `Cleanup-Downloads.json`.
Q	Beenden	Schließt das Menü.

Ohne Menü (Parameter)

Für Rollout/RMM lässt sich alles auch direkt per Parameter erledigen — ohne Rückfragen:

PARAMETER	BEDEUTUNG
`-Install`	Ohne Menü und **ohne Rückfragen** installieren (fordert Adminrechte selbst an); nicht angegebene Werte werden durch die Standardwerte bzw. die aktuell installierten ersetzt.
`-Uninstall`	Ohne Menü deinstallieren.
`-InstallPath`	Installationsordner (Standard `C:\Program Files\DUG-iT\Downloads-Cleanup`).
`-Folders`	Zu leerende Ordner — als Liste (`'{Downloads}','%USERPROFILE%\Desktop\Temp'`) oder Semikolon-getrennt.
`-Days`	Mindestalter in Tagen, `0` = alles löschen (0 bis 3650).
`-WhatIf`	Testlauf auf der Kommandozeile, löscht nichts (mit `-Install` / `-Uninstall`: zeigt nur die Zusammenfassung, ändert nichts).
`-Run`	Nur für die geplante Aufgabe: führt die Bereinigung ohne Menü und ohne Ausgabe aus.

Windows PowerShell

```
PS> powershell -ExecutionPolicy Bypass -File .\Cleanup-Downloads.ps1 -Install -Folde
PS> powershell -ExecutionPolicy Bypass -File .\Cleanup-Downloads.ps1 -WhatIf
PS> powershell -ExecutionPolicy Bypass -File .\Cleanup-Downloads.ps1 -Uninstall
```

04

Löschlogik & Schutz

Was genau gelöscht wird, in welcher Reihenfolge — und was grundsätzlich nie angefasst wird.

Was erfasst wird

- **Versteckte und schreibgeschützte Dateien** werden mit erfasst (das alte Skript übersah versteckte Dateien ohne `-Force`); der Schreibschutz wird vor dem Löschen entfernt.
- **Alter einer Datei** = der *jüngere* Wert aus Erstell- und Änderungsdatum. Das ist wichtig bei entpackten ZIP-Archiven: sie behalten das alte Änderungsdatum aus dem Archiv, haben aber ein frisches Erstelldatum — und sollen mit `Days` > 0 nicht sofort wieder verschwinden.
- **Junctions und Symlinks** werden nicht verfolgt: es wird nur der Link selbst entfernt, nie der Inhalt des Ziels.

Reihenfolge

1. **Dateien:** alle Dateien, die älter als das Mindestalter sind.
2. **Junctions/Symlinks:** nur der Link, sofern älter als das Mindestalter.
3. **Leere Ordner, von innen nach außen:** tiefste Ebene zuerst; ein Ordner wird nur entfernt, wenn er nach Schritt 1 und 2 wirklich leer und älter als das Mindestalter ist. Der Wurzelordner selbst (z. B. Downloads) bleibt immer stehen.

Gesperrte Dateien

Lässt sich eine Datei nicht löschen (z. B. von einem Programm gesperrt), bricht der Lauf nicht ab: der Fehler wird mit Pfad und Meldung protokolliert (`FEHLER Datei: ...`), im Zähler am Ende des Ordners als `fehlgeschlagen=` gezählt, und die Datei wird beim **nächsten Lauf** (nächste Anmeldung) erneut versucht.

Was nie geleert wird

ORDNER	ANMERKUNG
`Laufwerksstamm`	z. B. `C:\`

ORDNER	ANMERKUNG
Benutzerprofile	`%USERPROFILE%`, jedes weitere Profil direkt unter `C:\Users` und `C:\Users` selbst.
Windows	`%SystemRoot%`
Programme / Daten	`%ProgramFiles%`, `%ProgramFiles(x86)%`, `%ProgramData%`
AppData	`%APPDATA%`, `%LOCALAPPDATA%`
Installationsordner	Skript und Einstellungen — auch dann, wenn er *innerhalb* eines geleerten Ordners liegt.

Geschützt sind jeweils genau diese Ordner; Unterordner davon (z. B. `%APPDATA%\Foo`) dürfen konfiguriert werden. Ein geschützter oder nicht vorhandener Ordner in der Liste wird übersprungen und im Log vermerkt (`UEBERSPRUNGEN (geschuetzter Pfad)` bzw. `(nicht vorhanden)`), die übrigen Ordner werden normal bearbeitet.

> ! Gelöscht wird **dauerhaft** — es gibt keinen Papierkorb. Vor der ersten Installation oder bei jeder Änderung der Ordnerliste deshalb zuerst den **Testlauf** (Menüpunkt 3) nutzen: er zeigt Datei für Datei, was beim echten Lauf verschwinden würde.

05 Laufzeit-Artefakte

Alles, was Downloads-Cleanup auf dem Kunden-PC anlegt.

ARTEFAKT	INHALT
`Cleanup-Downloads.ps1`	Kopie des Skripts im Installationsordner — das führt die Aufgabe aus.
`Cleanup-Downloads.json`	Einstellungen: `Folders` (Liste) und `Days` (Zahl).
`Aufgabe "DUG-iT Downloads-Cleanup"`	Trigger: bei Anmeldung, 1 Minute Verzögerung. Principal: Gruppe `BUILTIN\Users` (SID `S-1-`

ARTEFAKT	INHALT
	`5-32-545`), eingeschränktes Ausführungslevel — läuft im Kontext des jeweiligen Benutzers. Läuft auch im Akkubetrieb, wird bei verpasstem Start nachgeholt, maximal eine Instanz gleichzeitig, Zeitlimit 30 Minuten, Fenster versteckt.
`Aufgabe "DUG-iT Downloads-Cleanup (manuell)"`	Zum Starten von Hand (Aufgabenplanung → Rechtsklick → Ausführen). **Kein Trigger**, Principal ist der bei der Installation *am PC angemeldete Benutzer* selbst (interaktiv, ohne gespeichertes Kennwort, eingeschränktes Ausführungslevel) — nie SYSTEM. Läuft nur, solange dieser Benutzer angemeldet ist; gleiche Aktion und Einstellungen wie die Anmelde-Aufgabe.
`%LOCALAPPDATA%\DUG-iT\Downloads-Cleanup.log`	Log **pro Benutzer**; ab 1 MB wird es zu `Downloads-Cleanup.log.old` rotiert.

Einstellungsdatei

C:\Program Files\DUG-iT\Downloads-Cleanup\Cleanup-Downloads.json

```
{
    "Folders":  [
                    "{Downloads}",
                    "%USERPROFILE%\\Desktop\\Temp"
                ],
    "Days":  0
}
```

Log-Beispiel

Downloads-Cleanup.log

```
2026-10-07 08:01:12  Start: Benutzer=praxis Days=0 Cutoff=07.10.2026 08:01:12
2026-10-07 08:01:13    FEHLER Datei: C:\Users\praxis\Downloads\Scan.pdf - Der Proze
```

```
2026-10-07 08:01:13    C:\Users\praxis\Downloads -> geloescht=42 fehlgeschlagen=1
2026-10-07 08:01:13  Ende
```

i Das Log liegt im Profil des Benutzers, für den der Lauf stattfand. Menüpunkt 5 zeigt das Log des **am PC angemeldeten Benutzers** — auch dann, wenn das Skript unter einem Admin-Konto läuft. Manuell: `C:\Users\<Name>\AppData\Local\DUG-iT\Downloads-Cleanup.log`. Die Statuszeile `Letzter Lauf` im Menükopf wird aus diesem Log abgeleitet (braucht keine Adminrechte).

06 Deinstallation

Menüpunkt 2 — entfernt die Einrichtung vollständig.

Nach einer Rückfrage (`j` / `n`, Standard `n`) entfernt die Deinstallation beide geplanten Aufgaben (Anmeldung und manuell) sowie `Cleanup-Downloads.ps1` und `Cleanup-Downloads.json` im Installationsordner. Der Installationsordner selbst wird nur entfernt, wenn er danach leer ist — liegt dort etwas anderes, bleibt der Ordner samt Inhalt bestehen.

! Die **Logs bleiben bewusst erhalten** (sie liegen pro Benutzer in dessen Profil, nicht im Installationsordner) — so lässt sich auch nach der Deinstallation noch nachvollziehen, was zuletzt gelöscht wurde. Bereits gelöschte Dateien kommen natürlich nicht zurück.

07 Komfortabler Einsatz beim Kunden

Genau dieselbe Infrastruktur wie WinPrep und NetMonitor — derselbe Server, derselbe Launcher, dieselbe Handhabung.

Downloads-Cleanup liegt als eigenes GitHub-Repo (`github.com/crolj/downloads-cleanup`) und wird wie die anderen Tools automatisch nach **dl.dug-it.de** veröffentlicht — spätestens 10 Minuten nach jedem `git push`, unter `/downloads-cleanup/`.

Der Ein-Zeiler

PowerShell öffnen, dann:

```
Windows PowerShell

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

Das lädt den generischen DUG-iT-Tool-Launcher (`DUGiT_Tools_Launcher.ps1`) — zeigt bei mehreren Tools (aktuell WinPrep/NetMonitor/Downloads-Cleanup) eine nummerierte Auswahl. Direkt zu Downloads-Cleanup, ohne Auswahlschritt:

```
Windows PowerShell

PS> iex(irm https://dl.dug-it.de/downloads-cleanup/DUGiT_DownloadsCleanup_Bootstrap.
```

`DUGiT_DownloadsCleanup_Bootstrap.ps1` lädt bei jedem Aufruf `Cleanup-Downloads.ps1` frisch herunter und startet es. Ist die Aufgabe bereits installiert, wird die Kopie im *tatsächlichen* Installationsordner aktualisiert (auch bei angepasstem Pfad), sonst in `C:\Program Files\DUG-iT\Downloads-Cleanup` ; ist dieser Ordner für den aktuellen Benutzer nicht beschreibbar (kein Administrator), lädt der Bootstrap die frische Datei in den TEMP-Ordner und startet sie von dort. Das Menü läuft ohne Adminrechte; nur Installieren/Deinstallieren öffnen selbst ein erhöhtes Fenster (UAC) — es muss also **keine** PowerShell als Administrator geöffnet werden.

> ! `https://` davor ist Pflicht: ohne Schema geht `Invoke-RestMethod` erst auf Port 80, Caddy leitet per 308 auf https weiter — Windows PowerShell 5.1 folgt 308-Redirects nicht zuverlässig. Gleicher Grund, gleiche Lösung wie im WinPrep-Handbuch, Kapitel 09.

Mit Parametern (z. B. für RMM)

Alle Parameter des Skripts werden vom Bootstrap unverändert durchgereicht:

```
Windows PowerShell

PS> iwr https://dl.dug-it.de/downloads-cleanup/DUGiT_DownloadsCleanup_Bootstrap.ps1
PS> .\boot.ps1 -Install -Folders '{Downloads}' -Days 0
PS> .\boot.ps1 -WhatIf
```

Über den Launcher geht dasselbe mit `.\boot.ps1 -Tool cleanup -Install -Days 0` (nach `iwr https://dl.dug-it.de/wp.ps1 -OutFile boot.ps1`).

Handbuch unterwegs

- dl.dug-it.de/downloads-cleanup/DUGiT_DownloadsCleanup_Handbuch.html
- dl.dug-it.de/downloads-cleanup/DUGiT_DownloadsCleanup_Handbuch.pdf

08 Anhang: Ordner, Troubleshooting & Release

Nachschlagen statt lesen — Ordnerstruktur, häufige Stolpersteine, und wie ein neues Release entsteht.

Ordner im Repo

DATEI/ORDNER	ZWECK
`Cleanup-Downloads.ps1`	Das Skript (Kapitel 01–06).
`DUGiT_DownloadsCleanup_Bootstrap.ps1`	Lädt & startet immer die aktuelle Version (Kapitel 07).
`VERSION`	Einzeilige Versionsnummer, Quelle für `build.ps1` und den Live-Deploy.
`build.ps1`	Erstellt ein versioniertes Release-Zip, siehe unten.
`tests/`	Pester-5-Tests gegen ein Sandbox-Verzeichnis (Löschlogik, Schutzliste, Platzhalter, Einstellungen).

Häufige Stolpersteine

- **Kein UAC-Fenster, Konsole schließt sofort** — meist Execution-Policy oder Domänen-GPO. `Get-ExecutionPolicy -List` prüfen (v. a. `MachinePolicy` / `UserPolicy`).
- **Fehler beim Download** — `Test-NetConnection dl.dug-it.de -Port 443` prüft, ob der Kunden-PC den Server erreicht (Firewall/Proxy).
- **Es wird nichts gelöscht** — Menüpunkt 3 (Testlauf) zeigt, was der Lauf täte, und das Log (Menüpunkt 4) zeigt Übersprungenes: `UEBERSPRUNGEN (geschuetzter Pfad)` bedeutet, der Ordner steht auf der Schutzliste aus Kapitel 04. Außerdem prüfen, ob die Aufgabe überhaupt installiert ist (Statuszeile im Menü).

- **Datei bleibt liegen** — im Log steht `FEHLER Datei: ...` mit der Meldung; meist hält ein Programm die Datei geöffnet. Sie wird bei der nächsten Anmeldung erneut versucht.
- **Downloads-Ordner liegt woanders als erwartet** — `{Downloads}` löst den tatsächlichen Pfad auf (Umleitung/OneDrive); in Vorauswahl und Zusammenfassung wird der aufgelöste Pfad angezeigt.
- **"Ich habe keine Adminrechte und kann das Skript nicht ausführen"** — seit V1.1 kein Problem mehr: Menü, Testlauf, "Jetzt bereinigen", Log und Einstellungen laufen ohne Adminrechte. Nur Installieren/Deinstallieren verlangen per UAC ein Admin-Konto.
- **Aufgabenplanung meldet "Das Benutzerkonto hat keine Berechtigung zum Ausführen dieser Aufgabe"** — die Anmelde-Aufgabe hat bewusst die *Gruppe* "Benutzer" als Principal (damit sie für jeden Benutzer läuft, der sich anmeldet); Aufgaben mit Gruppen-Principal lassen sich in der Aufgabenplanung nicht zuverlässig von Hand starten. Dafür gibt es seit V1.2 die Aufgabe `DUG-iT Downloads-Cleanup (manuell)` (Rechtsklick → Ausführen, läuft als der angemeldete Benutzer), alternativ Menüpunkt 4 "Jetzt bereinigen". Die Anmelde-Aufgabe selbst prüft man am besten, indem sich der Benutzer ab- und wieder anmeldet (eine Minute später steht ein Eintrag im Log).
- **Warum Program Files und nicht C:\DUGITInstall?** — `C:\DUGITInstall` erbt von `C:\` Ändern-Rechte für alle Benutzer. Das Skript läuft bei der Anmeldung im Kontext *jedes* Benutzers; läge es dort, könnte ein Standardbenutzer es ersetzen und damit Code im Kontext eines anderen Benutzers (z. B. eines Administrators) ausführen. Unter Program Files dürfen nur Administratoren schreiben. Alte Installationen (V1.0–V1.2) einmal über Menüpunkt 1 neu installieren — sie werden automatisch verschoben und der alte Ordner entfernt.
- **Es wird der falsche Downloads-Ordner angezeigt** — angezeigt wird der Ordner des *am PC angemeldeten* Benutzers (Zeile `Angemeldet` im Menükopf). Ist am PC niemand angemeldet (z. B. nur eine RDP-Sitzung), gilt das ausführende Konto. Die geplante Aufgabe selbst löst `{Downloads}` immer für den Benutzer auf, in dessen Sitzung sie läuft.

Release-Workflow

Für die interne Auslieferung als Zip: Version in der `VERSION`-Datei erhöhen, dann `.\build.ps1` ausführen — erzeugt `dist\Downloads-Cleanup-vX.Y.zip` mit dem Skript (Versionszeile automatisch gestempelt), README.md und optional (`-IncludeClaudeMd`) den internen Projekthinweisen. Für den laufenden Betrieb über dl.dug-it.de (Kapitel 07) reicht dagegen ein einfacher `git push` — der Deploy-Timer auf dem Server zieht die neue Version automatisch, ganz ohne Zip. Auf Kunden-PCs, auf denen die Aufgabe schon installiert ist, genügt danach ein erneuter Aufruf des Bootstraps, um die installierte Kopie zu aktualisieren.

Gelöscht wird nur, was der angemeldete Benutzer selbst löschen darf