






PackingIntunewin - Portable IntuneWin App Packaging Tool

Ein benutzerfreundliches, **vollständig portables** PowerShell-Tool zur automatischen Erstellung von Microsoft Intune **.intunewin** Paketen aus EXE-Dateien.

Ziel

Das Verpacken von **.exe**-Dateien in **.intunewin**-Apps so einfach und automatisiert wie möglich zu gestalten - **ohne manuelle Konfiguration oder feste Pfade**.

Features

-  **Vollständig portabel** - funktioniert von jedem Speicherort
-  **Automatischer Download** der Microsoft IntuneWinAppUtil.exe
-  **Intelligente Deinstallation** - Registry-Analyse & EXE-Parameter-Tests
-  **Automatische Batch-Erstellung** - install.cmd & uninstall.cmd
-  **App-Übersicht** - zeigt verfügbare Apps mit Status
-  **Fehlervalidierung** - prüft EXE-Anzahl und Ordnerstruktur
-  **Vollständige Metadaten** - für Microsoft Intune optimiert
-  **Farbige Ausgaben** - übersichtliche Fortschrittsanzeige

Ordnerstruktur

```
PackingIntunewin/
├── Create-IntuneWinApp.ps1      # 🚀 Hauptskript (portable)
├── README.md                   # 📖 Diese Dokumentation
├── apps/                       # 📁 Input-Ordner für Ihre Apps
│   ├── Chrome/                # Beispiel: Chrome-App
│   │   └── chrome-installer.exe # Ihre EXE-Datei
│   ├── VLC/                   # Beispiel: VLC-App
│   │   └── vlc-installer.exe   # Ihre EXE-Datei
│   └── YourApp/               # 👉 Ihr App-Ordner
│       └── yourapp.exe         # 👉 Ihre EXE-Datei (genau eine!)
├── IntuneWinApps/             # 📁 Output-Ordner (wird automatisch
    erstellt)                  # erstellt)
│   └── tools/                 # 🔑 Microsoft Tools (automatisch
        heruntergeladen)       # heruntergeladen)
│       └── IntuneWinAppUtil.exe # Microsoft Intune Win32 Content Prep
Tool
├── Chrome/                   # 📦 Verpackte Chrome-App
│   ├── install.cmd           # Automatisch generiert
│   └── uninstall.cmd         # Automatisch generiert
```

└─ chrome-installer.intunewin	# 📦 Fertiges Paket für Intune
└─ metadata.json	# Vollständige Metadaten
└─ YourApp/	# 📦 Ihre verpackte App
└─┬─ install.cmd	# Automatisch generiert
└─┬─ uninstall.cmd	# Automatisch generiert
└─┬─ yourapp.intunewin	# 📦 Fertiges Paket für Intune
└─┬─ metadata.json	# Vollständige Metadaten

Schnellstart

1. Repository klonen oder herunterladen

```
git clone https://github.com/Schepperto0/PackingIntunewin.git
cd PackingIntunewin
```

2. App vorbereiten

```
# Erstellen Sie einen Ordner für Ihre App im 'apps' Verzeichnis
mkdir "apps\MeineApp"

# Kopieren Sie Ihre EXE-Datei hinein (genau eine EXE pro Ordner!)
copy "C:\Downloads\meine-app-installer.exe" "apps\MeineApp\"
```

3. Skript ausführen

```
# Skript aus dem Hauptordner starten
.\Create-IntuneWinApp.ps1
```







4. App auswählen und konfigurieren

Das Skript führt Sie durch den Prozess:


- 📋 Zeigt verfügbare Apps an
- ? Fragt nach App-Name und Neustart-Anforderung
- 🔍 Ermittelt automatisch Deinstallationsinformationen
- 📦 Erstellt das fertige `.intunewin` Paket

Automatische Features




Tool Download Process

-  Prüfe IntuneWinAppUtil.exe...
-  IntuneWinAppUtil.exe nicht gefunden, lade von GitHub herunter...
-  Suche nach der neuesten Version...
-  Gefunden: IntuneWinAppUtil.zip (Version: v1.8.6)
-  Tool-Version: Microsoft Intune Win32 Content Prep Tool version 1.8.6.0
-  Download erfolgreich! Dateigröße: 0.89 MB

App-Status-Übersicht

-  Verfügbare Apps im 'apps' Ordner:
 - ☒ Chrome
 - ☒ VLC
 - ☒ Broken-App (Keine EXE)
 -  Multi-EXE-App (Mehrere EXE)

Intelligente Deinstallation

-  Ermittle Deinstallationsinformationen...
-  Suche nach Deinstallationsinformationen für 'Chrome'...
- ☒ Registry-Eintrag gefunden!
- ☒ Gefunden: Google Chrome
-  Verwende QuietUninstallString: MsiExec.exe /X{GUID} /quiet

Generierte Dateien

install.cmd

```
@echo off
echo Installing YourApp...
yourapp.exe /silent
if %errorlevel% neq 0 (
    echo Installation failed with exit code %errorlevel%
    exit /b %errorlevel%
)
echo Installation completed successfully
exit /b 0
```

uninstall.cmd

```
@echo off
echo Uninstalling YourApp...
MsiExec.exe /X{12345678-1234-1234-1234-123456789012} /quiet
```

```
if %errorlevel% neq 0 (  
    echo Uninstallation failed with exit code %errorlevel%  
    exit /b %errorlevel%  
)  
echo Uninstallation completed successfully  
exit /b 0
```

metadata.json

```
{  
  "AppName": "YourApp",  
  "Installer": "yourapp.exe",  
  "InstallCommand": "install.cmd",  
  "UninstallCommand": "uninstall.cmd",  
  "UninstallMethod": "Registry",  
  "UninstallString": "MsiExec.exe /X{GUID} /quiet",  
  "ExitCode": 0,  
  "RebootRequired": false,  
  "DetectionType": "Registry (manuell konfigurieren)",  
  "CreatedOn": "2025-07-03 14:30",  
  "CreatedBy": "username",  
  "ScriptVersion": "2.0",  
  "WorkingDirectory": "C:\\Path\\To\\PackingIntunewin"  
}
```

Microsoft Intune Konfiguration

Nach der Erstellung erhalten Sie alle benötigten Informationen:

 Informationen für Microsoft Intune:

App-Name: YourApp
Install-Befehl: install.cmd
Uninstall-Befehl: uninstall.cmd
Rückgabecodes: 0 (Erfolg), 3010 (Neustart erforderlich)

Erweiterte Funktionen

Deinstallations-Erkennung

Das Skript verwendet eine **intelligente Mehrstufen-Erkennung**:

1. Registry-Analyse (bevorzugt)

- Durchsucht `HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall*`

- Durchsucht
HKLM:\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall*
- Durchsucht HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall*

2. EXE-Parameter-Tests (Fallback)

- Testet häufige Parameter: `/uninstall`, `/remove`, `/u`, `/x`
- Kombiniert mit Silent-Flags: `/silent`, `/quiet`, `/s`, `/q`

3. Standard-Fallback

- Verwendet `/uninstall` `/silent` als letzte Option

Fehlerbehandlung

- ✗ **Keine EXE**: Klare Fehlermeldung bei fehlender EXE-Datei
- ✗ **Mehrere EXEs**: Warnung bei mehr als einer EXE-Datei
- ✗ **Ordner nicht gefunden**: Hilfreiche Pfad-Anzeige
- 🌐 **Netzwerkfehler**: Graceful Fallback mit manuellen Anweisungen

💡 Tipps & Best Practices

App-Vorbereitung

- ☒ **Eine EXE pro Ordner**: Genau eine EXE-Datei pro App-Ordner
- ☒ **Aussagekräftige Namen**: Verwenden Sie klare Ordner-Namen (z.B. "Chrome", "VLC")
- ☒ **Silent-Parameter testen**: Prüfen Sie vorab, ob Ihre EXE Silent-Installation unterstützt

Portabilität

- 📁 **Gesamten Ordner kopieren**: Kopieren Sie den kompletten `PackingIntunewin` Ordner
- 📁 **Keine Pfad-Anpassungen**: Das Skript funktioniert von jedem Speicherort
- 📁 **USB-Stick tauglich**: Perfekt für mobile Nutzung


Intune-Integration

- 📄 **Detection Rules**: Konfigurieren Sie manuell die Erkennungsregeln in Intune
- 🔍 **Registry-Erkennung**: Nutzen Sie die Metadaten für Registry-basierte Erkennung
- 📄 **Exit-Codes**: Berücksichtigen Sie die dokumentierten Rückgabecodes




🆘 Häufige Probleme & Lösungen

"IntuneWinAppUtil.exe konnte nicht heruntergeladen werden"


- 🌐 Internetverbindung prüfen
- 🔒 Firewall/Proxy-Einstellungen überprüfen
- 📄 Das Skript versucht automatisch mehrere Download-Methoden:
 - GitHub Release ZIP-Dateien (bevorzugt)
 - Direkte Download-URLs
 - Repository Raw-Links

-  **Manueller Download:** Laden Sie von [Microsoft GitHub](#) herunter und speichern Sie als `IntunewinApps\tools\IntuneWinAppUtil.exe`

"Keine EXE-Dateien gefunden"

-  Prüfen Sie den App-Ordner unter `apps\YourApp\`
-  Stellen Sie sicher, dass genau eine `.exe` Datei vorhanden ist
-  Verwenden Sie die App-Übersicht zur Diagnose

"Deinstallation nicht gefunden"

-  Das Skript testet automatisch verschiedene Parameter
-  Sie können die `uninstall.cmd` manuell anpassen
-  Konsultieren Sie die Dokumentation Ihrer Software

Beitragen

Verbesserungsvorschläge und Pull Requests sind willkommen!

1. Fork des Repositories
2. Feature Branch erstellen
3. Änderungen committen
4. Pull Request erstellen

Lizenz

Dieses Projekt steht unter der MIT-Lizenz.

Links

- [Microsoft Intune Win32 Content Prep Tool](#)
- [Microsoft Intune Dokumentation](#)
- [PowerShell Dokumentation](#)

 **Das Tool ist vollständig portabel und kann überall verwendet werden!** 

KI-Unterstützung

Dieses Projekt wurde mit Unterstützung von KI (GitHub Copilot) entwickelt, um die Codequalität, Dokumentation und Benutzererfahrung zu verbessern.