

Manifest (.psd1)

Was ist ein Manifest?

Ein Manifest (`.psd1`) beschreibt ein PowerShell-Modul.

Während die eigentliche Programmlogik in einer `.psm1`-Datei liegt, enthält das Manifest Informationen über das Modul selbst.

Dazu gehören beispielsweise

- Version
- Autor
- Abhängigkeiten
- exportierte Befehle
- unterstützte PowerShell-Versionen
- Lizenzinformationen

Das Manifest enthält **keinen ausführbaren Programmcode**. Es dient ausschließlich als Beschreibung und Konfiguration eines Moduls.

Allgemein

Das ist die Identität deines Moduls. Ohne das bist du einfach nur irgendein Skript mit Existenzkrise.

```
@{
    RootModule      = "TestModul.psm1"
    ModuleVersion    = "1.7.0"
    GUID             = "e9e0799e-734e-451d-9bb1-34983dd82f05"
    Author           = "ProgramMaster"
    CompanyName      = "Microsoft"
    Copyright        = "Microsoft i guess"
    Description      = "Ich beschreibe wundervoll, was das für ein Modul es ist"
}
```

RootModule

Hauptdatei (`.psm1` oder `.dll`)

ModuleVersion

Version (z. B. `1.0.0`)

GUID

Eindeutige ID

Author

Ersteller/Entwickler des Moduls

CompanyName

Unternehmen

Copyright

Urheberrechte

Description

Beschreibung des Moduls

Abhängigkeiten

Hier sagst du: „Ich funktioniere nicht alleine, ich brauche andere.“

- `RequiredModules`
 - Externe Abhängigkeiten
 - Angabe über **Modulnamen** (PowerShell sucht es selbst)
 - Müssen im System vorhanden sein (installiert / auffindbar, z.B. über `$PSModulePath`)
 - Gehören **nicht** zu deinem Modul
- `RequiredAssemblies` – DLLs
- `ScriptsToProcess` – Skripte, die beim Laden ausgeführt werden
- `ModuleList` – Nur die Meta-Information welche andere Module zugehörig sind
- `NestedModules`
 - Angabe über **Dateien/Pfade** (du sagst konkret, was geladen wird)
 - Interne Bausteine deines Moduls
 - Werden zusammen mit deinem Modul geladen
 - Gehören **zu deinem Modul dazu**

☐☐ Export (was nach außen sichtbar ist)

Das ist der Teil, wo du entscheidest, was du der Welt zeigst. Oder versteckst.

- `FunctionsToExport`
- `CmdletsToExport`
- `VariablesToExport`
- `AliasesToExport`

☐☐ Klassiker:

`'*'` = alles exportieren

oder explizit = bessere Kontrolle

☐☐ Kompatibilität & Anforderungen

Hier wird's picky.

- `PowerShellVersion`
 - `PowerShellHostName`
 - `PowerShellHostVersion`
 - `DotNetFrameworkVersion`
 - `CLRVersion`
 - `ProcessorArchitecture`
-

☐☐ Private Daten (der geheime Keller)

Alles, was du selbst definierst.

- `PrivateData` – Hashtable für eigene Infos

Beispiel:

```
PrivateData = @{
    PSData = @{
        Tags = @('UI', 'Forms')
        ProjectUri = 'https://...'
    }
}
```

☐☐ PSData (wichtig für Gallery etc.)

Teil von `PrivateData`, aber so wichtig, dass es eigentlich sein eigenes Ding ist.

- `Tags`
- `LicenseUri`
- `ProjectUri`
- `IconUri`
- `ReleaseNotes`

☐☐ Sonstiges (der „Warum gibt’s das?“ Bereich)

Selten gebraucht, aber existiert halt:

- `FileList`
- `TypesToProcess`
- `FormatsToProcess`
- `CompatiblePSEditions` (`Desktop`, `Core`)
- `HelpInfoURI`

☐☐ Mini-Beispiel

Damit du nicht nur trockene Theorie hast:

```
@{
    RootModule      = 'MeinModul.psm1'
    ModuleVersion   = '1.0.0'
    GUID            = '12345678-abcd-1234-abcd-1234567890ab'
    Author          = 'Jonny'
    Description      = 'Mein erstes Modul'

    FunctionsToExport = @('Get-Thing', 'Set-Thing')

    PowerShellVersion = '5.1'

    PrivateData = @{
        PSData = @{
            Tags = @('Example', 'Test')
        }
    }
}
```

Der eigentliche Punkt (den viele übersehen)

Du brauchst vielleicht **30% davon wirklich**.

Der Rest ist:

- entweder für Publishing
- oder für Leute, die Kontrolle lieben (oder Angst haben)

Wenn du lokal entwickelst:

→ `RootModule`, `ModuleVersion`, `FunctionsToExport`

→ fertig, läuft.

Revision #7

Created 2026-04-12 21:28:06 UTC by John-Andreas Borinas

Updated 2026-08-13 11:28:08 UTC by John-Andreas Borinas