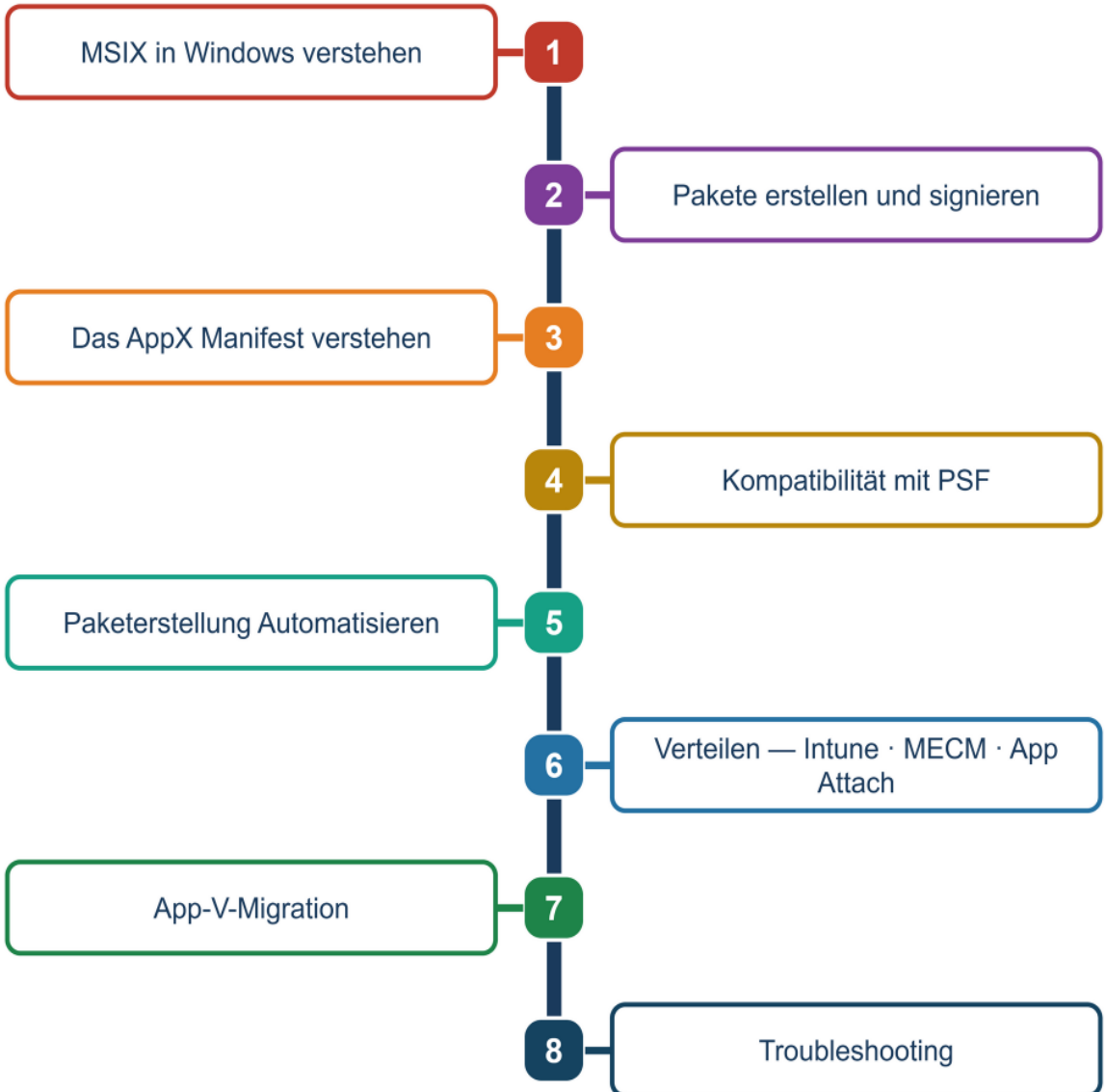


Das MSIX-Buch

Integration · Erstellung · Verteilung · Troubleshooting

Das Praxisbuch für Administratoren



Andreas Nick

Für Windows 11 und Windows Server



Das MSIX-Buch

Für meine Frau Dabge, danke für deine unendliche Geduld, wenn ich mal wieder 'nur noch schnell dieses eine Skript' testen musste. Ohne dich wäre dieses Buch nicht entstanden.

Andreas Nick 2026

Die in diesem Buch enthaltenen Skripte und Programme werden ohne jegliche Garantie oder Verpflichtung weitergegeben. Der Autor und der Verlag übernehmen daher keine Verantwortung und lehnen jede daraus resultierende oder sonstige Haftung ab. Bei der Erstellung der Texte und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autor können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Fast alle in diesem Buch erwähnten Hard- und Softwarebezeichnungen sind gleichzeitig eingetragene Warenzeichen oder sollten als solche betrachtet werden. Das Dokument folgt im Wesentlichen den Schreibweisen der Hersteller.

Dieses Werk und all seine Bestandteile sind durch das Urheberrecht geschützt. Jegliche Nutzung, die das Urheberrechtsgesetz überschreitet, ist ohne die Zustimmung des Autors nicht gestattet und kann rechtliche Folgen haben. Dies betrifft insbesondere das elektronische Kopieren und die Speicherung in digitalen Medien. Die in den Beispielen genannten Namen und Begriffe sind, sofern nicht anders vermerkt, fiktiv. Jegliche Ähnlichkeiten mit realen Unternehmen, Organisationen, Produkten, Domains, Personen, Orten, Ereignissen, E-Mail-Adressen oder Logos sind rein zufällig.

Anschrift: Andreas Nick, Dribusch 2, 30539 Hannover

Web: <http://www.nick-it.de>

Verlag: Eigenverlag Andreas Nick, Hannover

Druck: Print on demand, siehe letzte Seite.

Umschlaggestaltung: Andreas Nick

Icons: Font Awesome Free, <https://fontawesome.com>, Lizenz CC BY 4.0

Schriften: Lora (Cyreal, SIL Open Font License); Source Sans 3 (Adobe, SIL Open Font License); JetBrains Mono (JetBrains, Apache 2.0)

Copyright © 2026 Andreas Nick

Version 1.0 2026

Über den Autor

Andreas Nick ist IT-Berater und Spezialist für Anwendungsvirtualisierung, Softwarepaketierung und Windows-Client-Management. Seit 1998 berät er Unternehmen in diesen Bereichen; 2007 gründete er die Nick Informationstechnik GmbH mit Sitz in Hannover.

Seit 2003 beschäftigt er sich schwerpunktmäßig mit der Anwendungsvirtualisierung und der optimalen Bereitstellung von Software für Terminalserver und Desktops. Sein fachliches Profil umfasst Microsoft App-V, MSI-Paketerstellung, Citrix Virtual Apps and Desktops sowie die Automatisierung der IT-Administration mit PowerShell. Er ist Autor des ersten und bislang einzigen deutschsprachigen Fachbuchs zu Microsoft App-V 5. In den vergangenen Jahren hat er zahlreiche Schulungen zu App-V und MSIX im deutschsprachigen Raum sowie international durchgeführt.

Mit dem vorliegenden Werk legt er nun das erste umfassende deutschsprachige Fachbuch zu MSIX vor – dem modernen Windows-Paketformat, das App-V und klassische MSI-Installationen zunehmend ablöst.

Weiterführende Informationen finden sich im Blog unter <https://www.nick-it.de/blog>, der Themen rund um PowerShell, Terminalserver, App-V und MSIX behandelt. Mit dem Zeroportal Server hat Andreas Nick im März 2026 einen kompatiblen Ersatz für den Microsoft App-V Publishing Server veröffentlicht.

Web: <http://www.nick-it.de>

Mail: info@nick-it.de [mailto:info@nick-it.de]

X (Twitter): @nickinformation

Hinweise auf Schreibfehler oder technische Korrekturen sind willkommen und können über die oben genannten Kanäle eingereicht werden.

Inhaltsverzeichnis

1. Einführung	12
2. MSIX und die Universelle Windows Plattform (UWP)	14
2.1. Das MSIX-Paketformat – die App im Container	17
2.2. MSIX und MSI im Vergleich	20
2.3. Vorteile von MSIX	21
2.4. MSIX im Startmenü	24
2.5. Ausbau des MSIX-Dateiformats	25
Containertechnologie in Windows	27
2.6. MSIX- und AppX-Bundles	28
2.7. Filtertreiber	31
2.8. Datei- und Registry-Virtualisierung	33
2.8.1. Registrierungsvirtualisierung	33
2.8.2. Dateivirtualisierung	35
2.8.3. COM-Virtualisierung	36
2.9. Das primäre virtuelle Anwendungsverzeichnis (PVAD)	37
2.10. MSIX-Lizenzierung	38
2.11. Optimierte Updates – Single Instance Storage	39
2.12. Übermittlungsoptimierung für MSIX-Pakete	44
2.12.1. Übermittlungsoptimierung diagnostizieren	46
2.12.2. Konfiguration der Übermittlungsoptimierung mit einer GPO	47
2.13. Was ist eigentlich aus App-V geworden?	49
2.14. Featurevergleich App-V, MSIX und MSI	51
2.15. Entscheidungshilfe: MSIX oder nicht?	57
2.15.1. Wo MSIX an Grenzen stößt	57
3. Integration und Verwaltung von MSIX in Windows	59
3.1. Sideloading – nötig bis Windows 20H1	61
3.2. MSIX-Paketidentität	63
3.2.1. Berechnung der PublisherId	65
3.3. Anwendungspakete abfragen	68
3.4. Pakete bereitstellen	73
3.5. Staging und Registrierung	76
3.6. Reparieren von MSIX-Paketen	78
3.7. Wo die Benutzerdaten landen	79
3.8. Deinstallieren und Schutz vor Deinstallation	81
3.9. Anwendungen für alle Benutzer bereitstellen	84

3.10. MSIX von der Kommandozeile starten	90
3.10.1. Lokale Anwendungen im MSIX-Kontext starten	90
3.10.2. Eine MSIX-App über Shell:AppsFolder starten	91
3.11. Nützliche PowerShell-Einzeiler	94
3.12. Gruppenrichtlinien für moderne Anwendungen.	100
3.13. Richtlinien für Store-Anwendungen	103
3.14. MSIX-Anwendungen mit AppLocker absichern.	105
3.15. PowerShell-Cmdlets im Überblick	107
4. Digitale Signaturen für MSIX-Pakete.	109
4.1. Was ist ein Zertifikat – und was ist eine digitale Signatur?	109
4.2. Wie funktioniert ein Zertifikat?	111
4.3. Zertifikate auf einem Computer.	111
4.4. Zertifikate abrufen und anzeigen	115
4.4.1. Zugriff auf Zertifikate mit der MMC	115
4.5. Aufbau eines Zertifikats.	116
4.6. Zertifikate aus einer Public-Key-Infrastruktur (PKI)	121
4.7. Timestamp bei einer Codesignatur	121
4.8. MSIX-Pakete signieren	122
5. Das MSIX Packaging Tool	124
5.1. Die Windows App Runtime in jedem Paket.	125
5.2. Das Paketierungssystem.	127
5.2.1. Konfiguration des Paketierungssystems	127
5.2.2. Voraussetzungen für das Paketierungssystem	129
5.3. Installation des MSIX Packaging Tools	130
5.3.1. Installation über den Microsoft Store	130
5.3.2. Vorgefertigte VM aus der Hyper-V-Schnellerstellung.	131
5.3.3. Offline-Installation des MSIX Packaging Tools.	132
5.3.4. MSIX Packaging Tool-Treiber.	132
5.4. MSIX-Template	134
5.5. Paketerstellung mit dem MSIX Packaging Tool	137
5.5.1. Erststart und Diagnosedaten	137
5.5.2. Ablauf einer MSIX-Paketierung	137
5.5.3. Aufgabe wählen	139
5.5.4. Anwendungspaket erstellen	140
5.5.5. Accelerator (im Assistenten „Zugriffstaste“)	145
5.5.6. Installation der Anwendung	147
5.5.7. Aufgaben beim ersten Start.	150

5.5.8. Windows-Dienste im Paket	151
5.5.9. Paket erstellen	153
5.6. Remote-Paketierung	155
5.7. Änderungspakete erstellen	160
5.8. Pakete mit dem Paket-Editor bearbeiten	161
5.8.1. Funktionen (Capabilities) im Paket-Editor	164
5.8.2. Virtuelle Registrierung bearbeiten	167
5.8.3. VFS-Dateien bearbeiten	169
5.8.4. Paketanalyse	171
5.8.5. PSF-Fixups im Paket-Editor	171
5.9. Einstellungen des MSIX Packaging Tools	173
5.9.1. Standardeinstellungen	173
5.9.2. Daten und Einstellungen	176
5.9.3. Datei- und Registrierungsausschlüsse	177
5.9.4. Feedback	179
5.9.5. Info	180
5.10. Visual C++ Runtime	181
5.11. Troubleshooting im MSIX Packaging Tool	183
5.12. Praxisbeispiel: Ein vollständiger Paketierungsablauf	185
6. Das MSIXForcelets-PowerShell-Modul	190
6.1. Auszug aus den Cmdlets des Moduls	191
6.2. Modul einbinden	194
6.3. Paket öffnen, ändern und schließen	194
6.4. Paket signieren	195
6.5. Arbeitsablauf mit den MSIXForcelets	197
6.6. Beispiel: MSIX-Paket neu signieren	198
6.7. Beispiel: Fix für Notepad++	199
7. Das MSIX-AppxManifest	201
7.1. Das Application-Element	203
7.1.1. Application-Einträge mit den MSIXForcelets bearbeiten	205
7.1.2. VisualElements: Logos und Assets	206
7.1.3. App-Kacheln im Startmenü gruppieren (VisualGroup)	209
7.1.4. Arbeitsverzeichnis und Parameter (uap11:CurrentDirectoryPath)	211
7.2. File-Type-Associations im MSIX-Paket	213
7.2.1. File-Type-Associations mit den MSIXForcelets bearbeiten	215
7.3. Desktop7-Shortcuts	217
7.3.1. Regeln für Desktop7:Shortcuts aus der Praxis	220

7.3.2. Desktop7:Shortcuts mit den MSIXForcelets bearbeiten	221
7.4. Suchpfade im MSIX-Paket	223
7.4.1. Suchpfade mit den MSIXForcelets setzen	225
7.5. Kontextmenü und Shell-Extensions in MSIX	226
7.5.1. Kontextmenü-Varianten mit den MSIXForcelets	232
7.5.2. MSIX-Kontextmenü Diagnoseoptionen	233
7.6. Autostart für MSIX-Anwendungen	235
7.6.1. Autostart ohne den ersten App-Start setzen	237
7.7. Schriftarten im Paket	238
7.8. Firewallregeln über das AppxManifest	239
7.8.1. Firewallregel mit den MSIXForcelets	241
7.9. URL-Protokollhandler	243
7.10. InstalledLocationVirtualization	244
7.10.1. ILV mit den MSIXForcelets setzen	246
7.11. App Execution Alias	247
7.11.1. Wie der ExecutionAlias funktioniert	249
7.11.2. App Execution Alias mit den MSIXForcelets	252
7.12. Flexible Virtualization	253
7.12.1. Flexible Virtualization mit den MSIXForcelets	255
8. Microsoft-Werkzeuge zur MSIX-Paketbearbeitung	257
8.1. MakeAppx.exe – Pakete bauen und entpacken	257
8.2. Weitere Werkzeuge	259
9. Shared Package Container	261
9.1. Voraussetzungen und Plattformunterstützung	263
9.2. Eigenschaften und Einschränkungen	264
9.3. Vergleich mit App-V Connection Groups	265
9.4. Technische Funktionsweise	265
9.5. Verwaltung mit PowerShell	269
9.6. Container-XML mit den MSIXForcelets erzeugen	271
10. Das Package Support Framework (PSF)	272
10.1. Typische Probleme, die das PSF löst	273
10.1.1. Downloadquellen für das PSF	274
10.2. Dateien im PSF	275
10.3. Technischer Ablauf eines MSIX-Starts mit dem PSF	278
10.4. Der PSF-Launcher	279
10.4.1. Basiswerte der config.json	280
10.4.2. Pseudo-Variablen des PSFLaunchers	283

10.5. Anpassungen am Manifest für den PSF-Launcher	284
10.6. PSF zum Starten von Anwendungen	284
10.6.1. PSFLauncher Struktur mit den MSIXForcelets erzeugen	286
10.7. Skripte im PSF	287
10.7.1. Skripte mit den MSIXForcelets	290
10.8. Fixups konfigurieren: der Processes-Abschnitt	290
10.9. File Redirection Fixup	292
10.9.1. FRF Startelement <code>redirectedPaths</code>	293
10.9.2. FRF Element <code>redirectTargetBase</code>	295
10.9.3. Beispielkonfigurationen FRF ohne Exe und Dll Dateien	296
10.9.4. Aus der Praxis: gezielt umleiten, nicht alles	297
10.9.5. Verhalten von umgeleiteten Pfaden	298
10.9.6. Löschen und Ändern von Dateien	299
10.9.7. File Redirection Fixup mit den MSIXForcelets	299
10.10. Managed File Redirection Fixup (MFRFixup)	300
10.10.1. Wann ist das MFRFixup nötig?	301
10.10.2. Copy-on-Write für Programmdateien	303
10.10.3. MFRFixup mit den MSIXForcelets	304
10.10.4. Copy-on-Write in Aktion	304
10.11. DynamicLibraryFixup	305
10.11.1. DynamicLibraryFixup mit den MSIXForcelets	306
10.12. RegLegacyFixup	307
10.12.1. Universelle Konfiguration (empfohlen)	308
10.12.2. RegLegacy <code>HKLM2HKCU</code> im Detail	310
10.12.3. RegLegacy <code>ModifyKeyAccess</code> im Detail	311
10.13. PsfFtaCom Fix	311
10.13.1. FtaCom-Integration mit den MSIXForcelets	315
10.14. Umgebungsvariablen mit dem EnvVarFixup	317
10.14.1. EnvVarFixup mit den MSIXForcelets	318
10.15. Electron-Anwendungen im MSIX-Container	318
10.16. PSF-Tracing und Debugging	320
10.16.1. Der erste Weg: DebugView	320
10.16.2. Der zweite Weg: TraceFixup	323
10.16.3. Tracing mit dem PSFMonitor	326
10.16.4. PSFMonitor automatisch starten	327
11. MSIX App Installer	329
11.1. AppInstaller Gruppenrichtlinie	330

11.2. Die MSIX AppInstaller Konfigurationsdatei	332
11.3. App Installer Konfigurations-Tool	333
11.3.1. Einfache Bereitstellung von MSIX-Paketen über einen SMB-Share	334
11.4. Komplexes Beispiel	334
11.5. Abhängigkeiten (Dependencies) im AppInstaller	336
11.5.1. Abbrechen des Anwendungsstarts vor dem Update	340
11.5.2. Update durch den Benutzer unterbinden	341
11.5.3. Downgrade der Anwendung	341
11.5.4. Aktualisierung der Anwendung im Hintergrund	343
11.6. Veröffentlichung an einem Webspeicherort (HTTP)	344
11.6.1. App Installer-Pakete über einen Apache-Webserver	344
11.7. Optionale Pakete	347
11.8. Fehlerbehebung beim App Installer	348
11.9. Windows Package Manager (winget)	351
11.9.1. Windows Package Manager nutzen	353
11.9.2. Windows Package Manager und die PowerShell	356
12. Automatisierte Paketerstellung	359
12.1. Automatische Konvertierung zu MSIX-Anwendungen	364
12.2. Automatische Paketierung in der Praxis	365
12.3. Protokolle im Fehlerfall	368
13. Migration von App-V zu MSIX	370
13.1. App-V und MSIX im Vergleich	370
13.2. Migrationsstrategie	371
13.3. App-V-Pakete mit dem MSIX Packaging Tool konvertieren	371
13.4. Connection Groups migrieren	371
13.5. Dynamic Configuration migrieren	372
13.6. Häufige Herausforderungen bei der Migration	372
13.7. Testempfehlungen nach der Migration	373
14. Pakete mit MSIX Core bereitstellen	374
14.1. Wie MSIX Core ein Paket installiert: Devirtualisierung	375
14.2. MSIX Core auf Windows 10 und 11	377
15. MSIX und Microsoft Intune	379
15.1. Azure, Intune und Autopilot	380
15.2. Warum MSIX-Pakete mit Intune verwenden?	381
15.3. Intune und die MSIX-Codesignaturen	382
15.4. Verteilung von MSIX-Paketen	383
15.5. Erkennung und Updates	387

15.6. Abhängigkeiten und Framework-Pakete.	387
15.7. Bundles und Architektur	388
15.8. Grenzen gegenüber Win32-Apps.....	388
15.9. Fehlersuche bei der MSIX-Verteilung.....	388
16. Microsoft Endpoint Configuration Manager und MSIX-Pakete	390
16.1. Warum MECM für die MSIX-Verteilung nutzen?	390
16.2. Voraussetzungen und unterstützte Paketformate	391
16.3. MSIX-Pakete mit MECM verteilen	391
16.3.1. Abhängigkeiten.....	392
16.3.2. Installationskonfiguration.....	393
16.4. Diagnose und Fehlersuche	393
16.4.1. Relevante Log-Dateien auf dem Client.....	393
16.4.2. Client-seitige Fehler im Eventlog	394
17. MSIX in einer VDI- und RDS-Umgebung	396
17.1. Non-Persistent-Umgebungen.....	396
17.2. Windows Virtual Desktop (WVD)	396
17.3. App Attach.....	397
17.4. Azure Virtual Desktop und App Attach.....	399
17.5. App Attach einsetzen	402
17.5.1. Zertifikatsvertrauen auf Session Hosts	402
17.5.2. CIM-Image erstellen.....	403
17.5.3. CIM vs. VHDX – Entscheidungshilfe.....	403
17.5.4. VHDX-Image für App Attach per PowerShell erstellen.....	404
18. MSIX-Fehlersuche und Diagnose	410
18.1. Wo Windows die Fehler protokolliert	410
18.1.1. Erweiterte Diagnose-Kanäle aktivieren	412
18.1.2. Bereitstellungsprotokolle per PowerShell.....	413
18.2. Bereitstellungs-Ereignisse mitschneiden: NIT MSIX EventlogTracer.....	413
18.3. Laufzeit-Debugausgaben mit DebugView.....	416
18.4. Erweitertes Logging mit den MSIXForcelets	417
18.5. In den Container hineinsehen mit Invoke-CommandInDesktopPackage	418
18.6. Reparatur und Cache zurücksetzen	420
18.7. Weiterführende Microsoft-Dokumentation	420
19. Third-Party-Tools	421
19.1. PsfTooling.....	421
19.2. TMEditX.....	423
19.3. Advanced Installer.....	425

19.4. Advanced Installer Express	425
19.5. AppVentiX	427
19.6. MSIX Hero	428
19.6.1. Signierung	430
19.6.2. App Installer und winget	430
19.6.3. Shared Package Container	430
19.6.4. Ereignisprotokoll und Diagnose	430
19.6.5. Mehrsprachigkeit und Oberfläche	430
19.7. MSIX Commander	431
19.8. Store offline Download: store.rg-adguard.net	432
20. Anhang	435
20.1. Eine kurze Einführung in RegEx	435
20.2. Pfadvariablen in MSIX Paketen	436
20.3. Wichtige Begriffe	440
20.4. MSIX Packaging Tool Insider-Programm	452
20.5. Häufige MSIX-Fehlercodes	452
20.6. Kurzlinks und Quellen	453
Stichwortverzeichnis	465