InstallScout

User guide · version 1.9.2

From Windows installer to Intune: analyze the file, edit the silent command, package .intunewin and upload the Win32 app. Silent switches are one step — the destination is the package in Intune.

1. What InstallScout is

InstallScout is a Windows program for EXE, MSI, COM and MSIX. It does not run the installer. It analyzes the file, lets you edit the silent command, builds a local .intunewin (PSADT v4) and uploads the Win32 app to Intune with install, uninstall, detection and app logo.

Example	Engine	Typical result
Git-2.51.0-64-bit.exe	Inno Setup	Local and online agree on /VERYSILENT … · System
jre-8u491-windows-x64.exe	InstallShield	Local /s /v"/qn" · Java 8 /S · System
AcroRdrDC….exe	Self Extractor	/sAll … · online Acrobat Reader · System
SlackSetup.exe	Squirrel	Per-user · User

Start: double-click InstallScoutPortable.exe or run Start.bat. The UI defaults to English. About… shows the version, developer, and GitHub links. On startup, one anonymous call to Aptabase in the EU (eu.aptabase.com) sends the version and language. File names and paths are not sent. INSTALLSCOUT_USAGE=0 turns the call off.

The download is not code-signed. Windows or company security may warn or block it. Choose Keep / Run anyway, ask IT to allow the file, or try it in a VM or Windows Sandbox without those policies.

2. Start and window

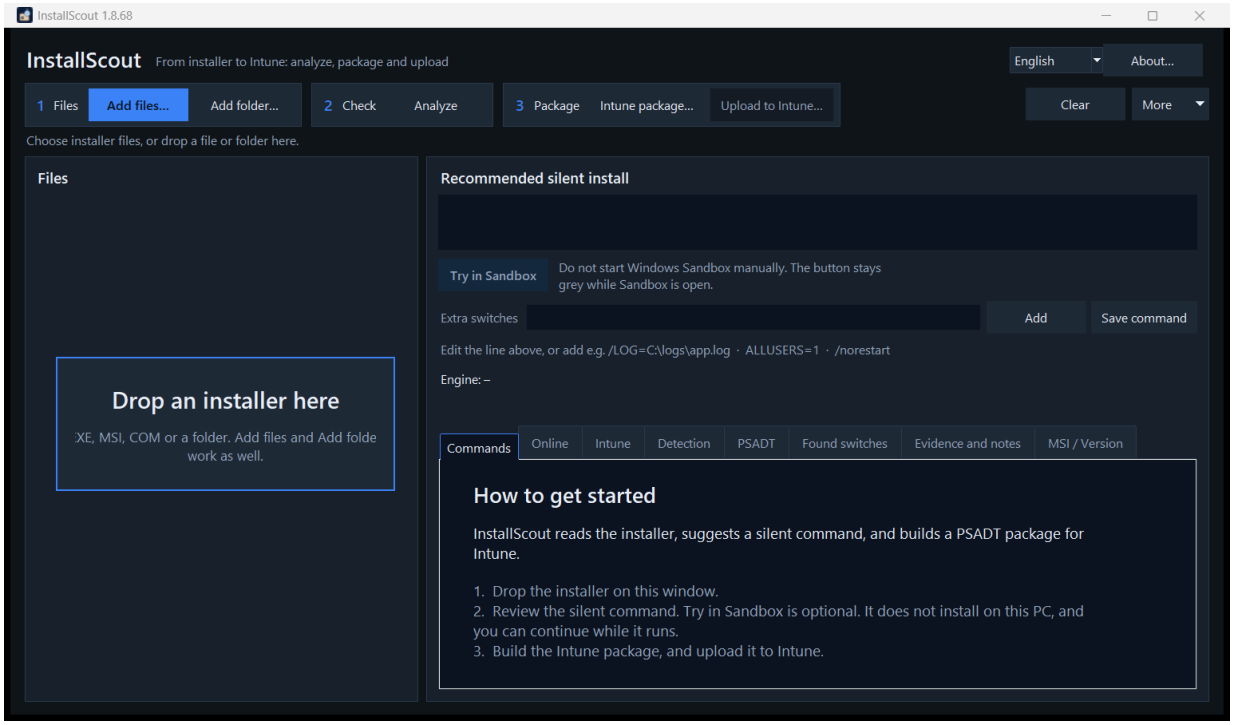


Figure 1. InstallScout 1.9.2 — Drop an installer here. Steps 1, 2 and 3 are grouped, and Try in Sandbox sits under the silent command.

Drop an installer on the window, or use Add files / Add folder. The empty start shows Drop an installer here in the file list and a short introduction on the Commands tab. Left: file, engine, confidence and status. Confidence is green from 75 %, amber from 40 %, and red below that. Status is Analyzed, Packaged or Rebuild. Right: the silent command wraps over three lines. Above the tabs: engine, confidence, System or User, and the online result. Only the next button is highlighted. Status and progress sit on their own row under the buttons.

Button	Action
Analyze	Read the file locally
Analyze	Local read, then WinGet + Chocolatey
Try in Sandbox	Optional. Run the silent command and detection in Windows Sandbox. You can continue while it runs
Silent Install HQ	Name search in the browser — the file is not uploaded
Add / Save command	Extra switches, or save the edited line
Intune package…	Local .intunewin — required before upload. The last folder is remembered
Package all…	One .intunewin per analyzed file. Upload stays one app
Upload to Intune…	Logo, sign-in, Win32 app and optional supersedence

3. Workflow

The process ends with upload. The silent command is a middle step.

1. Add the file and click Analyze.
2. Edit the green command, or add extra switches, then click Save command / Add.
3. Read the Online tab when it fills in. A confident local engine is kept on conflict. On none, use Silent Install HQ.

4. Open Intune and check Install context (System/User).
5. Build an Intune package (local .intunewin). Test with Invoke-AppDeployToolkit.exe.
6. Upload to Intune — the dialog shows the whole form. One Edge profile is shown; open the field to pick another. Sign in, optionally choose supersedence, and upload.

Try in Sandbox is optional and sits under the silent command when Windows Sandbox is installed. The button runs the silent command and the detection script in a sandbox. This PC is not changed. You can build the package and upload while the test runs. Close the sandbox afterwards. Its disk is discarded. The button is omitted when Sandbox is not installed.

4. Local switch check

Local analysis looks inside the binary and builds the command from that engine's known silent flags. Inner setup is unpacked locally without running the installer when the wrapper itself has no reliable silent line.

- Engine + confidence — 75 %+ is typically usable.
- Recommended silent install — the command PSADT and Intune use.
- Found switches and evidence — why the engine was chosen.

Situation	Unpack
7-Zip/WinRAR SFX, IExpress, WiX Burn, InstallShield, Advanced Installer, Self Extractor	ZIP, embedded MSI, CAB
Unknown engine or low confidence	Same cheap unpack
Inno, NSIS, Squirrel, DDPM, Wacom, Chrome, Python	Not unpacked
Self Extractor with /sAll, --silent or /silent	Outer command kept
Self Extractor without outer silent flags	Inner setup (often MSI) used

Example: Git (Inno Setup)

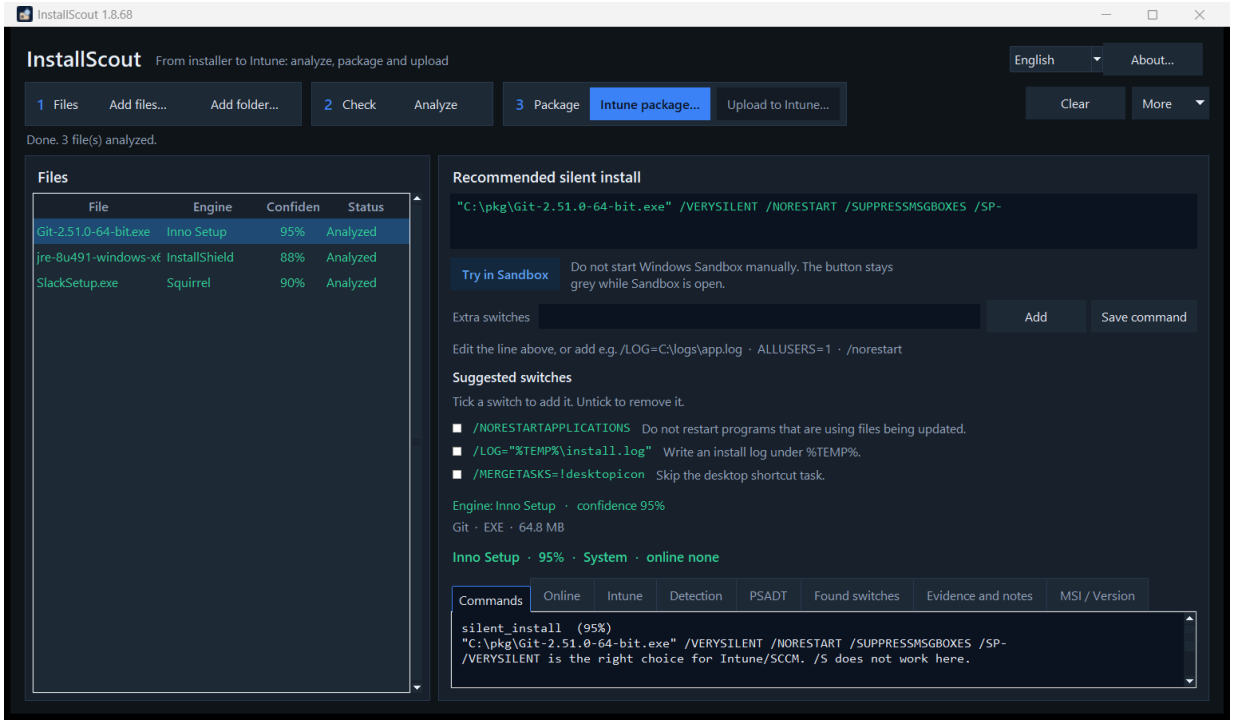


Figure 2. Git analyzed as Inno Setup. Try in Sandbox is ready.

```
Engine: Inno Setup · confidence 95 %
"C:\pkg\Git-2.51.0-64-bit.exe" /VERYSILENT /NORESTART /SUPPRESSMSGBOXES /SP-
```

Inno does not use /S.

Example: Java 8 (InstallShield)

```
jre-8u491-windows-x64.exe /s /v"/qn"
```

/s is InstallShield; /v"/qn" is passed to the embedded MSI.

In the PSADT package, an InstallShield line that contains quotes is started through cmd /d /s /c, so /v"/qn" is preserved. The package elevates itself when it is not already running as administrator.

Example: vendor Self Extractor

FileDescription Self Extractor is a vendor wrapper (Adobe, Lenovo and others), not a product. If /sAll or /rps is in the file, the line is /sAll /rs /rps /msi /quiet. Otherwise the inner setup is unpacked. Online and Intune use the product name or file stem (AcroRdrDC), not the wrapper label.

Engine	Silent
MSI / WiX	msiexec /i file.msi /qn
NSIS	/S (capital S)
Advanced Installer	/exenoui /qn
WiX Burn	/quiet
7-Zip SFX / Wacom	/s
Self Extractor	/sAll … or --silent / /silent
DDPM	/Silent

5. Manual switches

The green line is editable. Under it are Extra switches, Add and Save command.

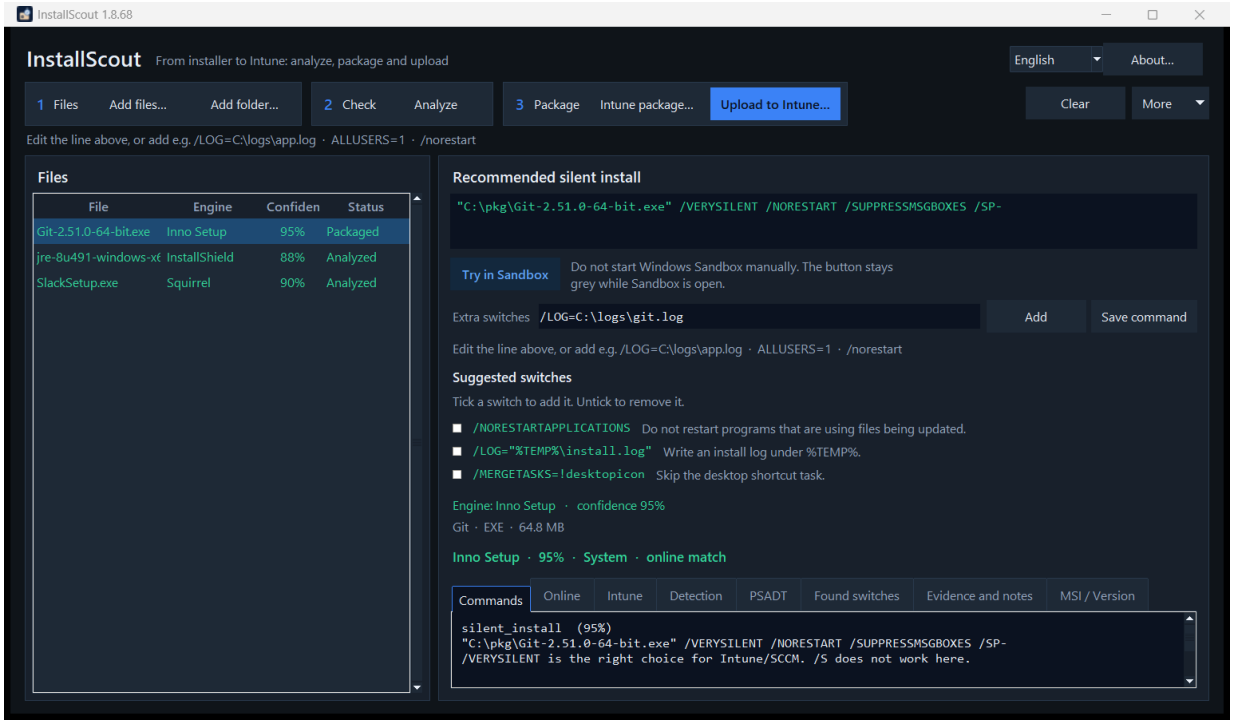


Figure 3. Extra switches, Add and Save command.

- Edit the whole command and click Save command (or press Enter).
- Put extra flags in Extra switches, e.g. /LOG=C:\logs\app.log, ALLUSERS=1 or /norestart, then click Add.

- Suggested switches appear with a short description. Tick one to include it. /install and /norestart are not added automatically. Inno keeps /NORESTART.
- Changes are also saved when you package, copy or check online.
- If the command changes, build the Intune package again before upload.

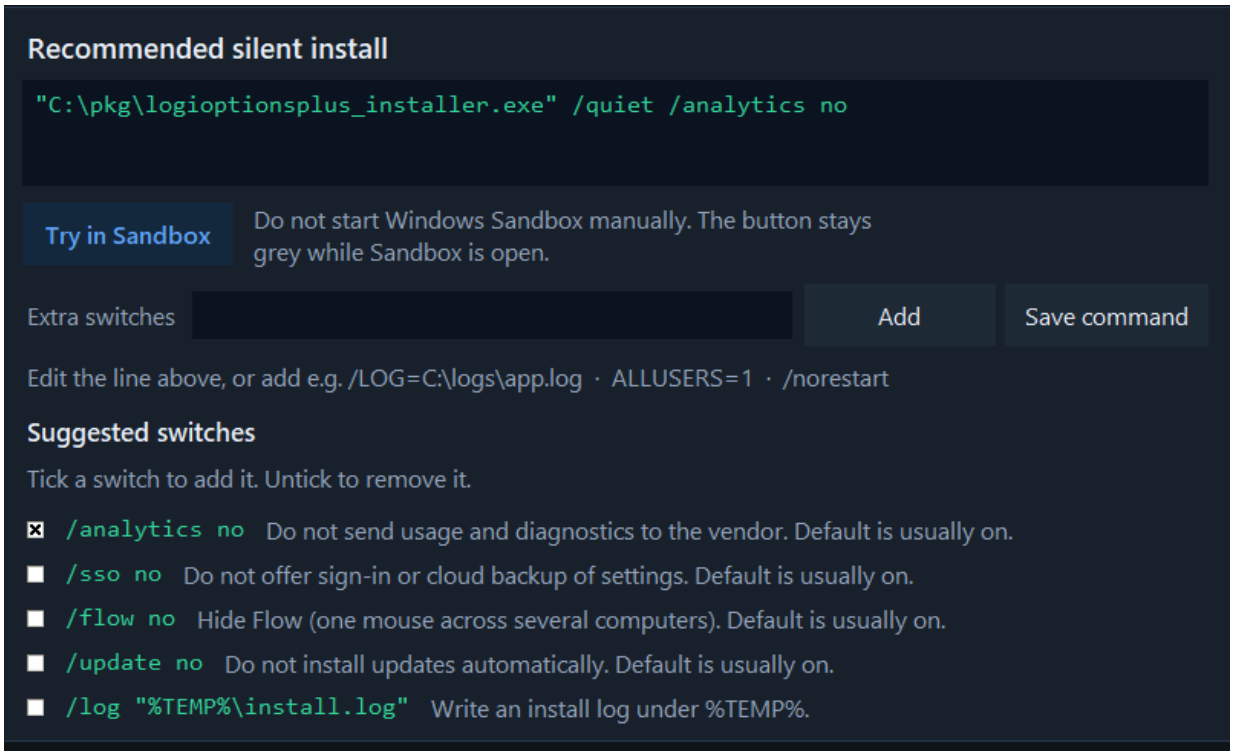


Figure 4. Suggested switches. Tick a box to add the switch. Here /analytics no is selected.

6. Online switch check

Analyze looks the product up in WinGet and Chocolatey afterwards. A confident local engine is kept when the catalog disagrees. A weak local guess is replaced when the catalog matches. Silent Install HQ (silentinstallhq.com) opens a name search — the file is not uploaded. Only the product name and vendor are sent. Wrapper labels such as Adobe Self Extractor are not searched. Compact file names are expanded: AcroRdrDC → Acrobat Reader, GoogleChromeStandalone… → Google Chrome, npp.8.6 → Notepad++. Java Platform SE 8 U491 is listed as Java 8.

Result	Meaning
match	Same installer family
partial	Some flags overlap
conflict	Different family — test in a VM
none	No hits
error	Network or source failure

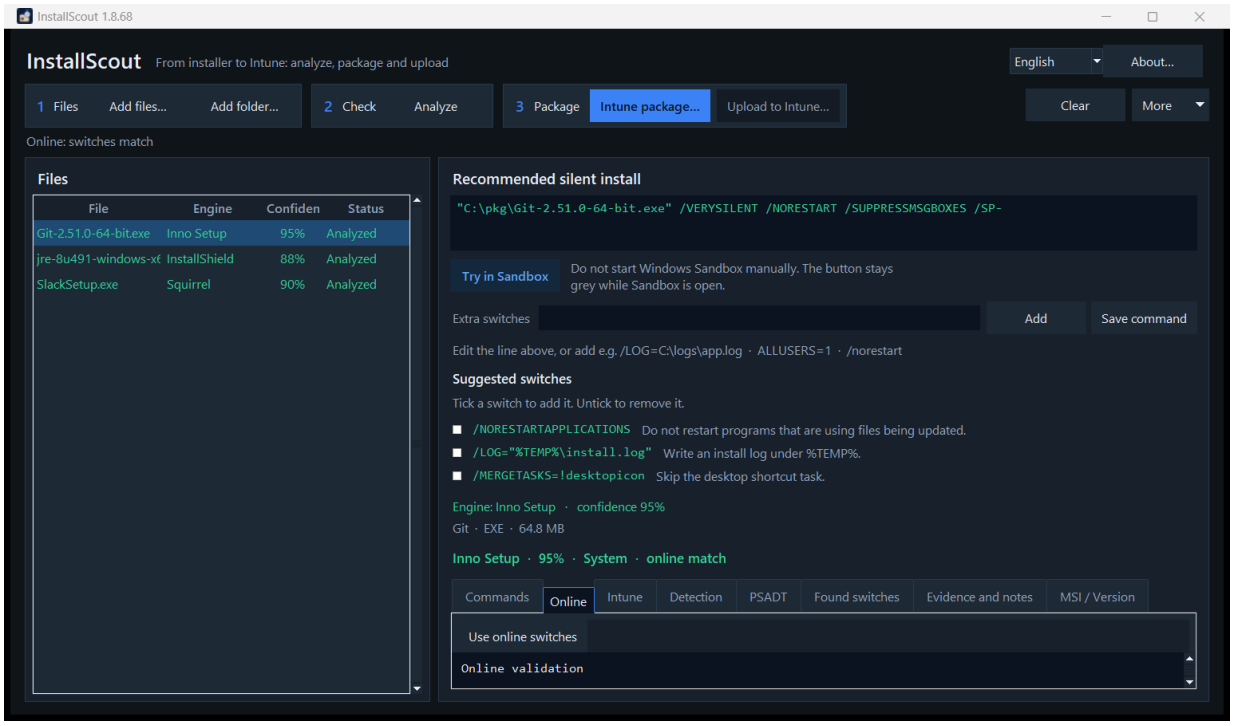


Figure 5. Online match for Git.

Java: online finds /S, locally it is InstallShield /s /v"/qn". Keep local switches when they match the engine in the file. Use online switches shows both lines before the local command changes.

7. User vs system context

Intune Win32 must run as system or user. InstallScout infers context and shows it on the Intune tab and in Upload to Intune. You can override it at upload.

- Squirrel, ALLUSERS=0 / InstallAllUsers=0, or AppData in the arguments → User.
- ALLUSERS=1, MSI, InstallShield, Advanced Installer, WiX, Wacom, MSIX → System.
- Inno, NSIS and 7-Zip SFX with no per-user marker → System.
- A path under %LOCALAPPDATA%\Temp does not count — only the arguments after the file path.

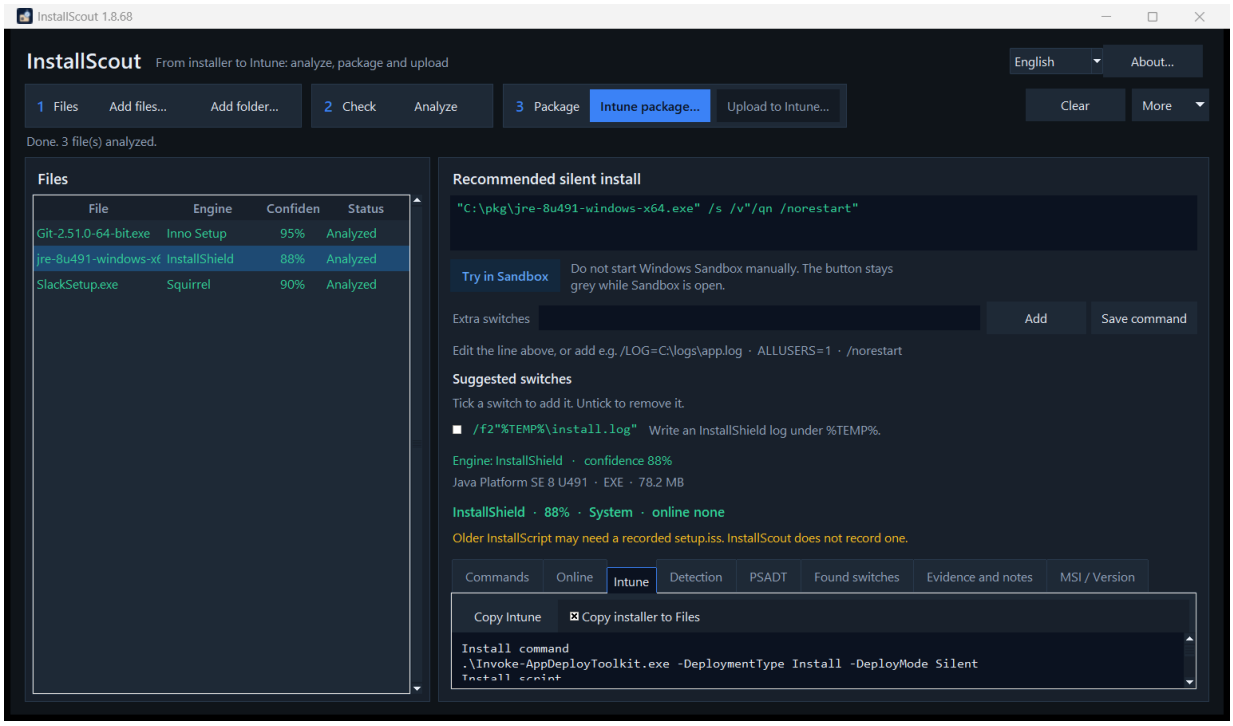


Figure 6. InstallShield/Java with Install context System.

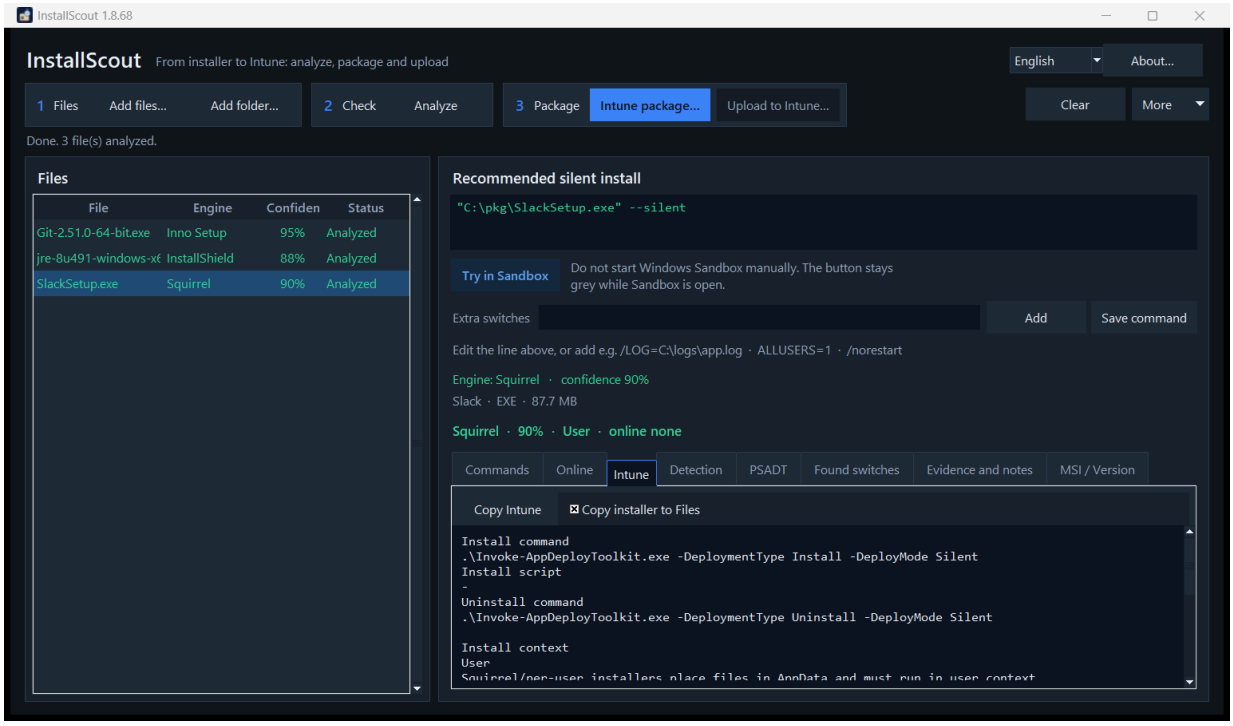


Figure 7. Squirrel/Slack with Install context User.

Context	Typical target	Detection
System	C:\Program Files, HKLM Uninstall	Machine ARP / MSI ProductCode
User	%LOCALAPPDATA%, HKCU Uninstall	The user's Uninstall key

8. Upload to Intune

Two steps: first a local .intunewin, then Upload to Intune. The program uploads the package you already saved.

`Analyze → optional extra switches → Intune package… → Upload to Intune… (logo + Graph)`

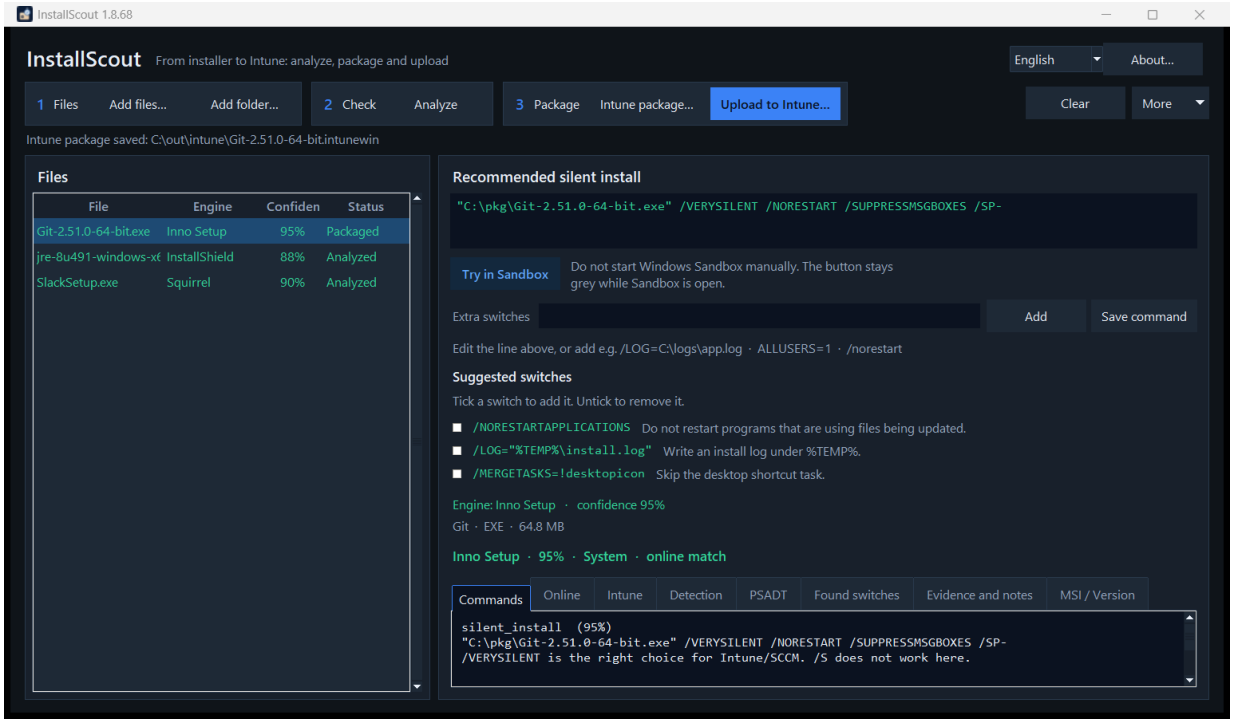


Figure 8. Upload to Intune is active when a local .intunewin exists.

Test the source before upload:

```
.\Invoke-AppDeployToolkit.exe -DeploymentType Install -DeployMode Silent
```

App logo

On Upload to Intune the icon is extracted from the EXE or the MSI Icon table (and logo.png / icon.ico next to the file, if present). Override with Choose file…, Paste (Ctrl+V) or Remove. A package with no icon has no logo.

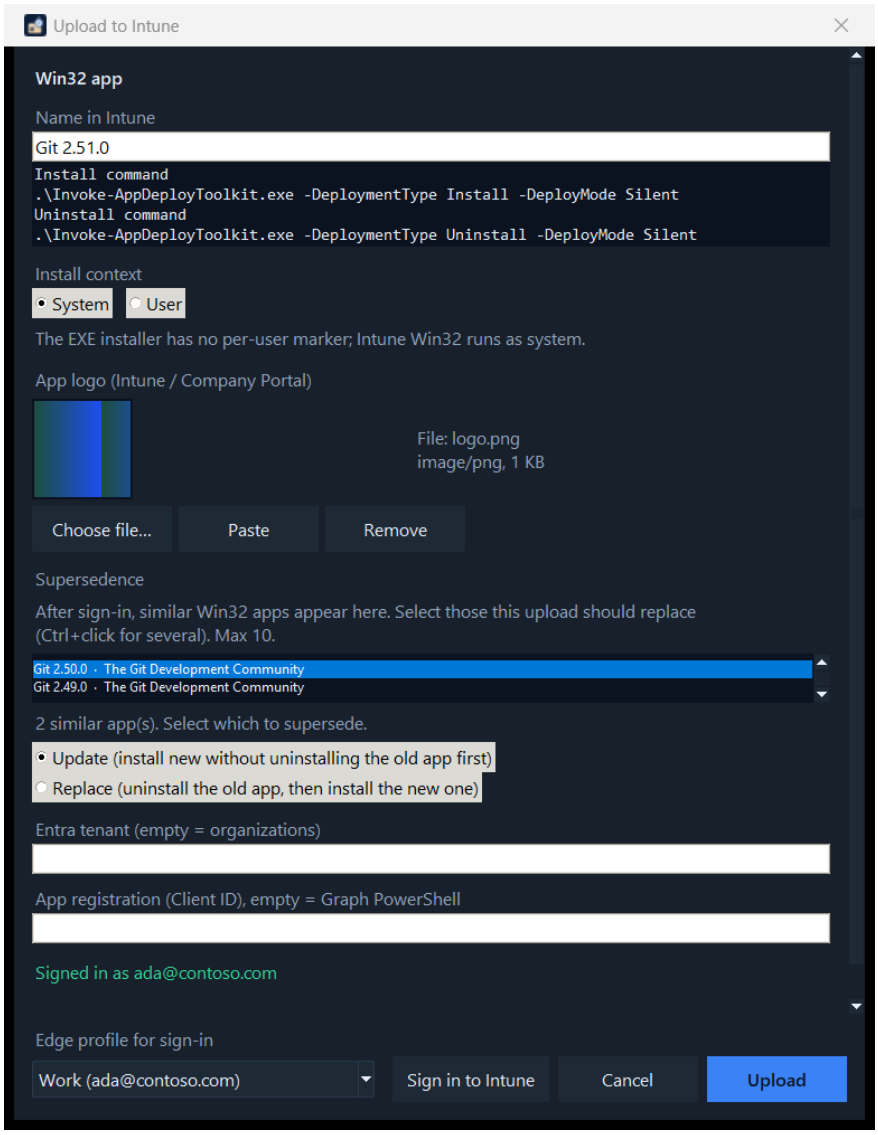


Figure 9. App logo in Upload to Intune: preview, Choose file, Paste, Remove.

7. Check install, uninstall and detection.
8. Confirm Install context (System/User).
9. Check the app logo. Override with a file or paste if the automatic icon is wrong.
10. Tenant and Client ID can stay empty (organizations + Microsoft Graph PowerShell). InstallScout does not create an Entra app.
11. The field shows one Edge profile. Open it to pick another, then click Sign in to Intune (PKCE, localhost). Sign-in is bound to the tenant and Client ID and is stored with Windows DPAPI.
12. After sign-in: optionally select supersedence (Update or Replace) among similar Win32 apps. Nothing is selected automatically.
13. Upload — creates the app, uploads .intunewin, sets detection, logo and optional supersedence.

Supersedence

The new Win32 app can replace older versions in the tenant (the same feature as in the Intune portal). Wait until the status is Signed in as …, not only that the Edge profile opened. The search uses the product name, not the entire Win32 catalogue. Nothing is selected automatically.

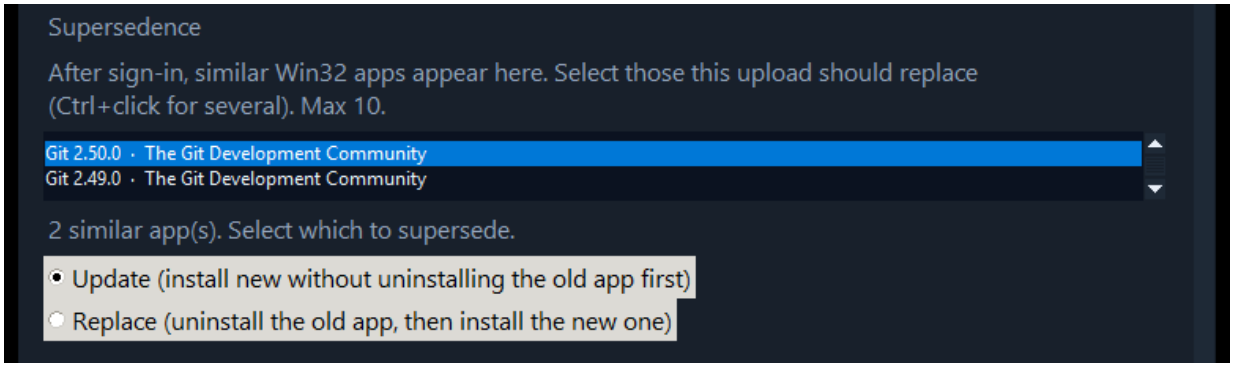


Figure 10. Supersedence in Upload to Intune: Git 2.50.0 selected, Update is the default.

- Ctrl+click to select (at most 10 — Intune's limit).
- Update (default): install the new app without uninstalling the old one first.
- Replace: uninstall the old app, then install the new one.
- Relationships are written only after the new app is created (Graph updateRelationships).
- If supersedence fails, the new app still remains. You can set the relationship in the portal afterwards.
- CLI: --supersede update|replace matches on product name. Repeat --supersede-id with a known app ID.

Detection

Intune treats the app as installed when Detection.ps1 writes a line to STDOUT and exits 0. Empty output means not installed. The same rules apply to every vendor.

- Product code (MSI, or a GUID in the uninstall command) is tried first, and the version must be the same or newer.
- An EXE without a product code is matched on display name, publisher and version. Logi Options+ Installer matches Logi Options+.
- A trailing Installer, Setup, Bootstrapper, or Driver word may be absent. Wacom Tablet Driver matches Wacom Tablet.
- Legal suffixes are ignored: Logitech, Inc. matches Logitech, and Microsoft Corporation matches Microsoft.
- Version is compared numerically. 2.7.961922 and a newer version count; 2.6 does not. 2.8.1 is only an example.
- MSIX is matched on the package name. Both the machine and the user uninstall keys are read.

Intune field	Value
App type	Win32 (win32LobApp)
Install / uninstall	.\Invoke-AppDeployToolkit.exe -DeploymentType … -DeployMode Silent
Detection	PowerShell script, 64-bit
runAsAccount	system or user
Minimum OS	Windows 10 22H2
Max runtime / restart	60 min / No specific action
Logo	largeIcon (PNG/JPEG from EXE, MSI, file or clipboard)
Supersedence	Optional: Update or Replace against up to 10 older Win32 apps

- An account that can create Win32 apps: Intune Administrator, or the Intune role Application Manager.

- Admin consent for DeviceManagementApps.ReadWrite.All. Default Client ID: Microsoft Graph PowerShell (14d82eec-204b-4c2f-b7e8-296a70dab67e).
- Conditional Access can block. IT can set a custom Client ID with redirect http://localhost.

9. Command line

```
.\InstallScoutPortable.exe --cli "C:\installers\Git-64-bit.exe"
.\InstallScoutPortable.exe --cli .\setup.exe --online
.\InstallScoutPortable.exe --cli .\setup.exe --sihq
.\InstallScoutPortable.exe --cli .\setup.exe --switches "/LOG=C:\logs\app.log"
.\InstallScoutPortable.exe --cli .\setup.exe --intune C:\out\intune
.\InstallScoutPortable.exe --cli .\setup.exe --intune C:\out\intune --publish-intune --icon
.\logo.png
.\InstallScoutPortable.exe --cli .\setup.exe --intune C:\out\intune --publish-intune --
supersede update
.\InstallScoutPortable.exe --cli .\setup.exe --intune C:\out\intune --publish-intune --
supersede replace --supersede-id <app-guid>
```

10. Supported engines

MSI, Inno Setup, NSIS, InstallShield, WiX Burn, Advanced Installer, InstallAware, BitRock, Squirrel, Wise, install4j, Qt Installer, 7-Zip SFX, Wacom, WinRAR SFX, IExpress, Self Extractor, DDPM, MSIX/AppX, ClickOnce. Wrappers and unknown EXEs are unpacked locally first so an inner MSI can replace Unknown 15 %.

11. Limits

- The download is not code-signed. Windows or company security may warn or block it. Choose Keep / Run anyway, ask IT to allow the file, or try it in a VM or Windows Sandbox without those policies.
- The program does not install anything during analysis. Unpack never runs the installer. Try in Sandbox is the test that does run the command, and only inside Windows Sandbox.
- Online catalogs may use a different name or a newer build. Wrapper VersionInfo (Self Extractor) is not a catalog product.
- Context and logo are qualified guesses. Override them in Upload to Intune.
- A package with no icon has no logo — use Choose file or Paste.
- Upload to Intune is inactive until a local .intunewin exists.
- The supersedence list appears only after sign-in has finished (Signed in as …).
- If supersedence fails, the new Win32 app still remains in Intune.
- Installer binaries are not sent during the online check.