

Packaging and deploying applications with Patch My PC Cloud

A practical guide to Custom Apps for Microsoft Intune. How to take a raw installer and ship it as a managed app, with no manual Intune packaging step.

PREPARED BY Cornerstone Cyber **AUDIENCE** IT team

PLATFORM Patch My PC Cloud → Microsoft Intune **REVIEWED** June 2026

HOW TO READ THIS GUIDE

Every section links to the Patch My PC or Microsoft documentation it is based on. Treat those links as the source of truth. Patch My PC ships changes often, so confirm against the live docs before a production run. Screenshots are linked to the official pages rather than copied, so you always see the current console.

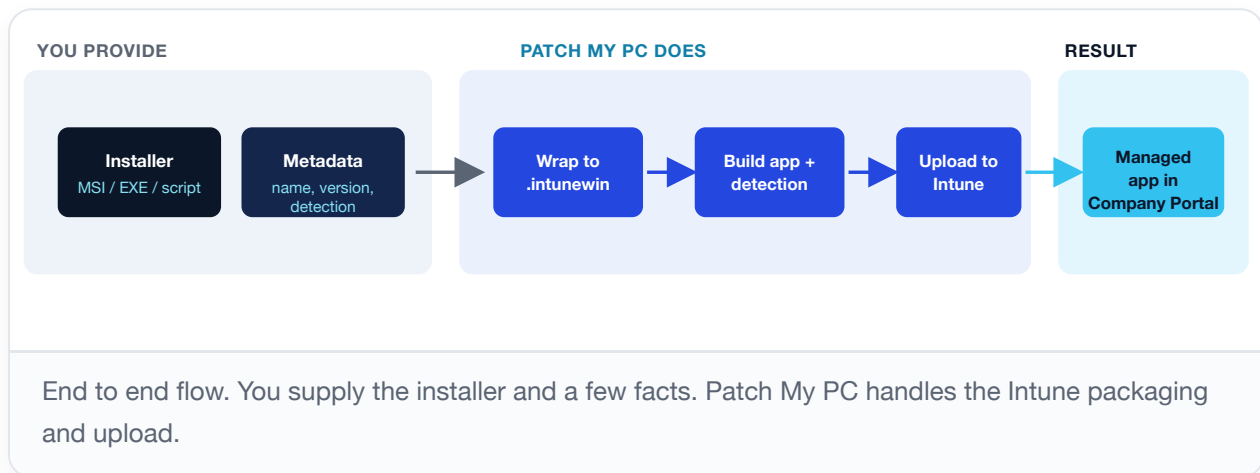
CONTENTS

1. What Custom Apps does
2. Catalog apps vs Custom Apps
3. Before you start
4. Choose your installer type
5. Step by step: create a Custom App
6. Worked example: MSI
7. Worked example: EXE
8. Worked example: Microsoft 365 Apps
9. Deploy the app
10. Migrating existing Intune apps
11. Maintain: update, modify, remove
12. How detection works
13. Troubleshooting and gotchas
14. Reference links

What Custom Apps does

Custom Apps lets you package your own or in-house software and publish it to Microsoft Intune straight from the Patch My PC Cloud portal.

You upload the raw installer. Patch My PC wraps the content into the Win32 `.intunewin` format and uploads it to Intune for you. You never run the Microsoft Win32 Content Prep Tool (`IntuneWinAppUtil.exe`) by hand, and you never upload a package to Intune manually.



What this replaces

MANUAL INTUNE METHOD	PATCH MY PC CUSTOM APPS
Run <code>IntuneWinAppUtil.exe</code> to build a <code>.intunewin</code>	Upload the raw installer, packaging is automatic
Write install and uninstall commands by hand	Auto filled for MSI, guided for EXE and script
Build and test detection rules yourself	Default detection script provided and recommended
Upload and configure the app in the Intune console	Published to Intune from the Patch My PC portal
Repeat the whole process for every new version	Add Version, the rest carries over

Source of truth: [Custom Apps Overview](#)

Catalog apps vs Custom Apps

Patch My PC has two ways to deliver software. Pick the right one before you build anything.

App Catalog

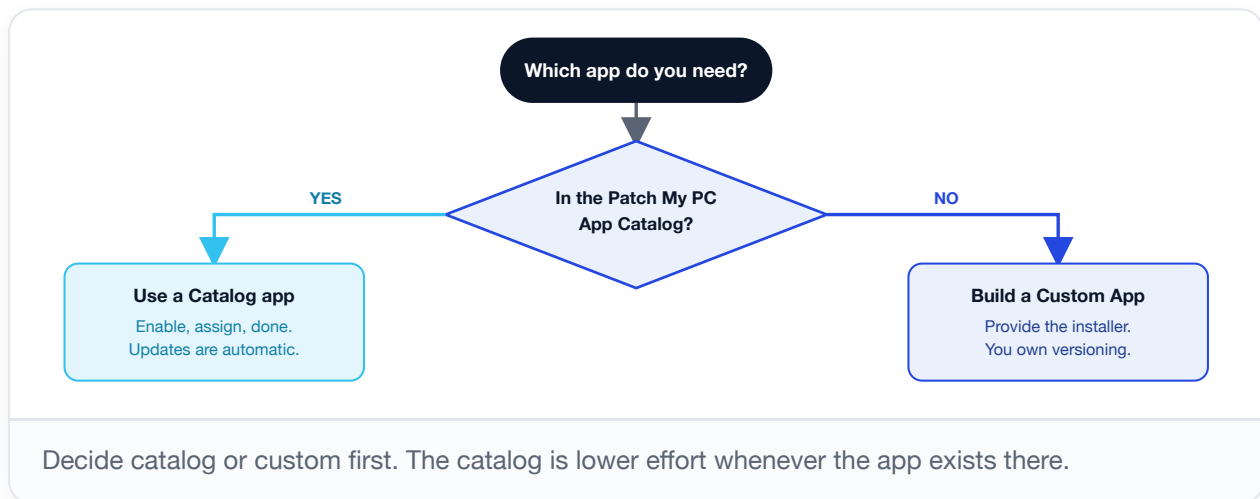
Over a thousand common third party apps that Patch My PC packages, tests, and keeps current for you. You enable the app and set assignments. Patch My PC handles new versions automatically.

Use when the app is a public product such as Chrome, Zoom, or 7-Zip.

Custom Apps

Your own or in-house software that is not in the catalog. You provide the installer and metadata. Patch My PC packages and publishes it. You own the version updates through Add Version.

Use when the app is line of business, internally built, or a public app with bespoke configuration.



NOTE

If you deploy catalog apps as well as Custom Apps, connect your Intune tenant first through Add an Intune Connection.

Before you start

Requirements

- An Enterprise Plus, Enterprise Premium, or MSP Patch My PC subscription.
- Your company is onboarded to Patch My PC Cloud and set up.
- An Intune connection configured if you publish to Intune from the Cloud, or also deploy catalog apps.
- The on premises Publisher is only needed if you publish Custom Apps through it. Version 2.1.20.0 or later is required in that case. Cloud to Intune does not need the Publisher.

IMPORTANT

Custom Apps does not support macOS. Windows only.

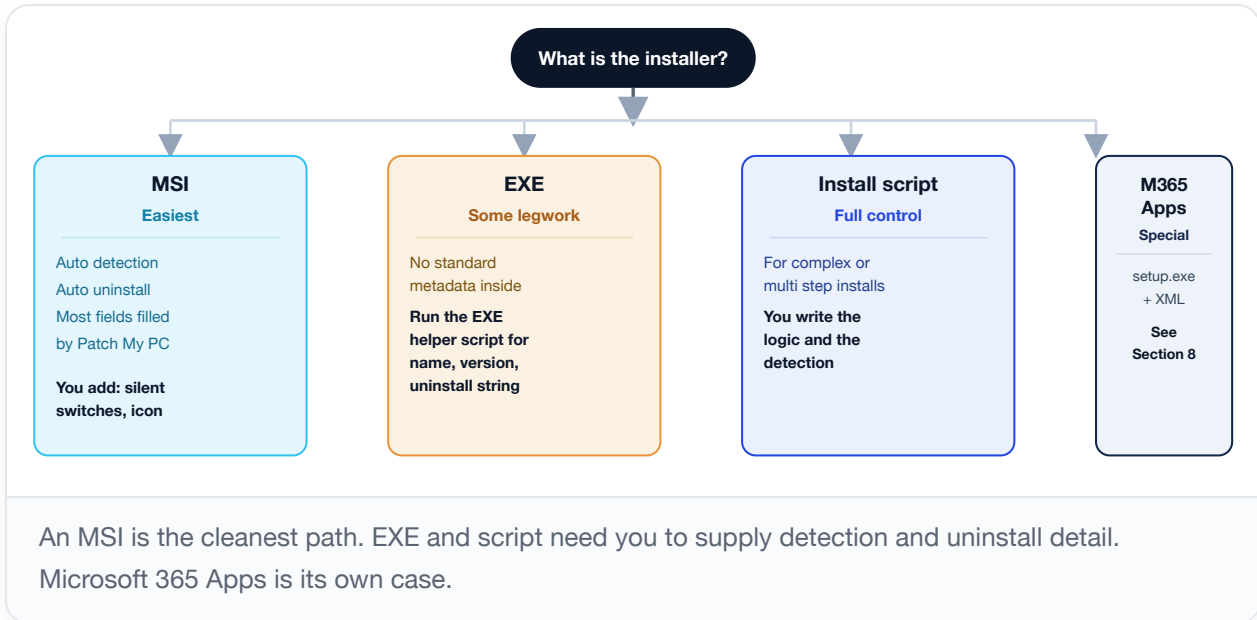
What to gather for each app

ITEM	WHY IT MATTERS
The installer file (MSI or EXE) or an install script	This is the primary install file Patch My PC packages
Silent install command	So the app installs with no user prompts
Apps & Features (ARP) display name and version	Drives reliable detection after install
A 512 x 512 px icon (PNG, JPG, or BMP)	Shows in the Company Portal for end users
Any extra files or folders the installer needs	Uploaded alongside the primary installer

Source of truth: [Custom Apps Requirements](#) · [Onboard to Custom Apps](#)

Choose your installer type

How much work the wizard does for you depends on what you feed it. This single choice drives the rest of the build.



TIP

If you have a choice between an MSI and an EXE for the same product, take the MSI. Patch My PC reads the product code, the uninstall string, and most Configuration fields straight from it.

Source of truth: **Find properties for EXE-Based Installers**

Step by step: create a Custom App

Sign in to portal.patchmypc.com, open the App Catalog, and click **Add App**. The Custom Apps Deployment Wizard opens. It has five tabs.



The five wizard tabs, in order. Click Next to move forward, or Save once everything is correct.

1 File tab: upload the installer

Click **Add Primary Install File** and select your MSI or EXE, or choose an installer script. The file hash is calculated as it uploads. Add any extra files or folders the installer needs, then click **Next**.

IMPORTANT

If you add PSADT scripts, the target devices need .NET 4.7.2 installed.

2 General Information tab: describe the app

Set the icon (512 x 512 px), App name, Vendor, Description, and Internal Notes. Write the App name exactly as it should read in the Company Portal, and keep it unique. If two apps share a name, only one shows in the Company Portal. Click **Next**.

3 Configuration tab: how it installs

Set Install Context (System or User), Architecture, Version, Apps & Features name, Language, Conflicting Processes, Silent Install parameters, and the Uninstall command. For an MSI most of this is filled for you. Add Information and Privacy URLs. Click **Next**.

4 Detection Rules tab: prove it installed

Leave **Patch My PC Default (Recommended)** selected unless you have a reason not to. It checks the uninstall registry for your app name and a version equal to or greater than the one you set. Click **Next**.

5 Summary tab: review and create

Check every value. Use **Prev** to fix anything. When correct, click **Create**. The app appears in your App Catalog ready to deploy.

Source of truth: [Create a Custom App](#) · [File tab](#) · [General Information tab](#) · [Configuration tab](#) · [Detection Rules tab](#) · [Summary tab](#)

MSI

EASIEST

This is the everyday case. Say you have `AcmeAgent.msi`.

1 Upload

Add `AcmeAgent.msi` as the Primary Install File. Next.

2 Describe

Icon, App name `Acme Agent`, Vendor `Acme`, Description. Next.

3 Confirm Configuration

Patch My PC reads the product code, version, and uninstall command from the MSI. Set Install Context to System and confirm Architecture. Add silent switches only if the MSI needs them, for example `/qn`. Next.

4 Detection

Leave Patch My PC Default. Next.

5 Create

Review and click Create.

WHY MSI IS SIMPLE

An MSI carries its own product code and uninstall logic, so Patch My PC fills the important Configuration fields and detection works without extra effort.

Source of truth: **MSI fields are auto filled (Custom Apps Reference)**

EXE

SOME LEGWORK

An EXE does not store its details in a standard place, so detection needs help. Patch My PC publishes a helper script that reads the right values from the registry.

1 Test install first

Install the EXE on a clean test VM or in Windows Sandbox. Open Installed Apps and note the exact display name and version.

2 Run the helper

Use the CustomAppsHelper script to pull the display name, version, and uninstall string from the registry.

3 Build the app

Upload the EXE. On the Configuration tab enter the Apps & Features name and Version exactly as found, the silent install switches, and the uninstall command.

4 Detection

Patch My PC Default works once the name and version match the real installed values.

WATCH OUT

If the Apps & Features name or version is even slightly wrong, the app installs but never detects, so Intune reports failure and may retry. Verify against a real install.

Source of truth: [Find properties for EXE-Based Installers](#)

Microsoft 365 Apps SPECIAL CASE

Microsoft 365 Apps is not an MSI. It uses Click to Run, driven by `setup.exe` from the Office Deployment Tool and an XML config. That is why it needs more steps than a normal app.

Prepare the files

1. Build your config at config.office.com and save it as `Install.xml`.
2. Download the Office Deployment Tool and extract `setup.exe`.
3. Create an `Uninstall.xml` from the install config. Do it by hand, or use Patch My PC's helper.

Build the Custom App

- Primary Install File: `setup.exe`
- Extra Files: `Install.xml` and `Uninstall.xml`
- Silent install: `/configure Install.xml`
- Uninstall command (custom): `& '.\setup.exe' /configure '.\uninstall.xml'`
- Detection: Patch My PC Default. Test install once to capture the exact ARP name and version.

Online vs offline content

	ONLINE (RECOMMENDED)	OFFLINE
Package size	Small, content streams from the Microsoft CDN at install	Large, often 4 GB or more, content is bundled
Best for	Most environments	Devices that cannot reach the Office CDN
Extra step	None	A pre install script expands the bundled content first

SERVICING

Do not push new versions of Microsoft 365 Apps by redeploying a Custom App. Manage updates through Microsoft's own servicing model and update channels. Use the Custom App only for the initial install.

SIZE LIMIT

Intune caps a single app at 30 GB. An offline package with bundled Office content can approach that, which is another reason to prefer online content unless the network demands otherwise.

FASTER PATH

Patch My PC publishes `Invoke-M365AppsHelper.ps1`, which reads your Install.xml and generates the package files, commands, and detection for you. See the Community Scripts repo.

Source of truth: [How to package Microsoft 365 Apps as a Custom App](#) · [Office Deployment Tool \(Microsoft Learn\)](#) · [Change update channels](#)

Deploy the app

Creating the app adds it to your catalog. Deploying it sends it to Intune and assigns it to people.

1 Open the app

Find your Custom App in the App Catalog and click **Deploy**.

2 Configurations

Add pre or post install scripts if needed. For zipped content, a pre install script expands the archive before the installer runs.

3 Assignments

Target your Entra ID groups. Start with a small test group.

4 Deploy

Confirm. Patch My PC publishes the app to Intune.

5 Validate

Install from the Company Portal on a test device. Check the install log under `C:\Windows\Temp` if anything fails.

TIP

New versions reach devices on the next daily sync. To push immediately, use Sync Now.

Migrating existing Intune apps

If apps are already in Intune, do not blindly repackage everything. Treat it as a triage and cutover exercise: sort what is there, move each class the right way, and align detection so the switch is silent.

1. Triage what is already there

Pull the existing app list from Intune and sort each app into one of four buckets. Each is handled differently.

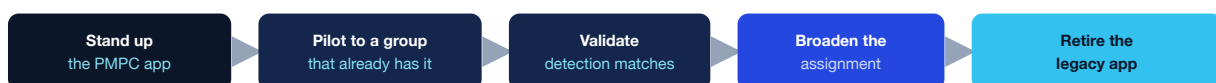
EXISTING APP TYPE	BEST TREATMENT
Public third party app in the Patch My PC catalog (Chrome, Reader, Zoom, 7-Zip)	Do not repackage. Enable the catalog app and let Patch My PC own packaging and patching.
In-house or line of business app	Rebuild as a Patch My PC Custom App, then retire the manual one.
Microsoft 365 Apps	Leave it. Manage through Microsoft's servicing model and update channels, not a redeploy.
Store or winget apps	Usually leave to the Store. Move only if you specifically want Patch My PC patching.

THE BIG WIN

Most existing apps are common third party tools. Those should become catalog apps, not custom repackages, so Patch My PC keeps them current automatically.

2. Migrate without disruption

The key to a clean cutover is detection. When the new app's detection (Apps & Features display name, and version equal to or greater than) matches what is already installed, devices report as installed and Patch My PC simply takes over patching, with no reinstall and no user impact.



A silent cutover. Matched detection means existing installs are recognised, so devices move to Patch My PC management with no reinstall.

1 **Stand up the PMPC version**

Enable the catalog app, or build a Custom App for line of business software.

2 **Pilot to a small group**

Target an Entra ID group that already has the legacy app installed.

3 **Validate detection**

Confirm the app reports as installed with no reinstall, and that future updates flow.

4 **Broaden the assignment**

Extend the PMPC app to the full target groups.

5 **Retire the legacy app**

Remove its assignment, or delete the legacy app object.

THE PITFALL TO AVOID

Be careful how you retire the old app. Deleting a Win32 app object in Intune stops management but does not uninstall it from devices. If you instead add an Uninstall assignment, or remove a Required assignment in a way that triggers uninstall intent, Intune pulls the software off machines. Switch management to the PMPC app first, then remove the legacy assignment rather than assigning it to uninstall.

TIP

Consider Intune app supersedence for a formal hand off from the old package to the new one. Never let the old and new app both actively install or uninstall against the same devices at once.

Source of truth: [Win32 app management in Intune](#) · [Add and assign Win32 apps](#) · [confirm migration steps against the Patch My PC Custom Apps docs](#)

Maintain: update, modify, remove

TASK	HOW	NOTES
New version	Open the app, click Add Version , walk the File, Configuration, Detection, and Summary tabs, then Create	The new version installs on the next sync, or use Sync Now
Change settings	Use Modify a Custom App	Adjust metadata, configuration, or detection
Uninstall	Use Uninstall a Custom App	Removes the app from targeted devices
Delete	Use Delete a Custom App	Removes the app from your catalog

IMPORTANT

You cannot change the Installer Type of an existing Custom App. If you started with an EXE you cannot switch it to a script later. Plan the type up front.

Source of truth: [Update a Custom App](#)

How detection works

Intune needs a way to know the app is present. The default detection method scans the uninstall registry and looks for a match.

- The Display Name matches the Apps & Features name you set on the Configuration tab.
- The Display Version is equal to or greater than the version you set.

When you deploy the latest build of a product and do not know the exact version number, this name plus version rule is what makes detection reliable. The values you enter must match what the installer actually writes. This is why a test install matters for EXE and Microsoft 365 Apps.

IN SHORT

Right name, right or higher version, reliable detection. Wrong name or version, the app installs but reports as failed.

Troubleshooting and gotchas

SYMPTOM	LIKELY CAUSE	FIX
App installs but Intune shows failed	Detection name or version does not match the real install	Test install, capture the exact ARP name and version, correct the Configuration tab
Install never completes	Missing silent switch, or a conflicting process is open	Add the correct silent parameters and list conflicting processes
EXE detection unreliable	No standard metadata in the EXE	Use the CustomAppsHelper script to read the registry values
PSADT package fails on some devices	.NET 4.7.2 not present	Ensure .NET 4.7.2 is on targeted devices
Offline Microsoft 365 package too large to manage	Office content bundled into the package	Use online mode, or compress content and expand it with a pre install script
Need the new version out now	Waiting on the daily sync	Use Sync Now

Install logs for most Custom Apps land in `C:\Windows\Temp`. Microsoft 365 Apps writes its own log there too.

Source of truth: [Log Reference Guide](#)

Reference links

Patch My PC documentation

- Custom Apps home
- Custom Apps Overview
- Custom Apps Requirements
- Onboard to Custom Apps
- Create a Custom App (File, General Information, Configuration, Detection Rules, Summary tabs)
- Update a Custom App
- Modify a Custom App
- Uninstall a Custom App
- Delete a Custom App
- Find properties for EXE-Based Installers
- Add an Intune Connection

Worked examples and scripts

- Package Microsoft 365 Apps as a Custom App
- Package Visual Studio 2022 as a Custom App
- Patch My PC Community Scripts (GitHub)
- CustomAppsHelper script (EXE properties)

Microsoft

- Win32 app management in Intune
- Office Deployment Tool
- Office Customization Tool (config.office.com)
- Change Microsoft 365 Apps update channels



Secure today. Stronger tomorrow.
info@cornerstonecyber.com.au

Prepared as a client handover reference. Based on Patch My PC and Microsoft documentation current at June 2026. Confirm steps against the linked source pages before production use, as the Patch My PC portal changes regularly.