

Readme IntuneApp

See also: <https://github.com/ITAutomator/IntuneApp>

Overview	2
Features	2
Quick Start.....	2
Download IntuneApp.....	2
Test a pre-packaged installer	3
Test installing a few apps at once.....	3
Prepare your Org for Intune	3
Publish Apps.....	3
Push (Assign) Apps to Users.....	3
Manually install an App for a User.....	4
Main Menu.....	4
List / Create Apps (L)	4
Install / Uninstall apps (I)	5
Copy apps (to a USB key) (C).....	5
Publish / Unpublish apps (P)	5
Common Actions.....	7
Publishing to All Users	7
Checking App publishing status in Intune.....	7
Other Intune Apps (Non IntuneApp Apps)	8
Microsoft Store App: Company Portal.....	8
Steps.....	8
Microsoft 365 Apps (MS Office).....	8
Steps.....	9
Technical Details	9
Log Files.....	9
Intune Settings	9
Internal dependencies	11
Intune Tracing	11
IntuneApp files.....	11

Overview

AppsMenu.ps1

Installs and publishes apps using either Intune or manual methods.

About Apps

- Intune enabled (Win32) apps are a package of files that install a product on an e
- Typically, apps are pulled from the Winget or Choco online repositories.
- Apps can also be custom scripts.
- Most Apps only require an icon (graphic) file and a settings file to indicate the

AppsMenu.ps1 Main menu

- L - List / Create apps**
Shows the current apps. Creates app based on winget template
- I - Install / Uninstall apps**
Installs (or Uninstalls or Detects) apps on the current machine
- C - Copy apps (to a USB key)**
Copies apps to a portable folder for later install
- P - Publish / Unpublish apps**
Publishes apps to an existing org. Create new org.

Select an Action

[X] Exit [L] List [I] Install [C] Copy [P] Publish [?] Help (default is "X"):

Use this to easily publish an application to Intune PCs.

Features

Apps are packaged in portable folders designed for

- One-time installs (e.g. via \Downloads folder)
- Bulk installs (by copying many App folders and launching a single installer)
- Intune installs (using the native Intune system and endpoint agent service)

Apps are added to the Company Portal so people can self-install without admin rights.

Apps can be made mandatory or optional but always controlled by Group membership.

Apps can be created using

- [Winget](#) (Recommended)
- [Chocolatey](#)
- MSI installer
- Custom [PowerShell](#) .ps1 scripts
- Many [Sample Apps](#) are pre-built

Quick Start

Download IntuneApp

Go to <https://github.com/ITAutomator/IntuneApp>

Click *Code (the green button)* > *Download Zip*

Extract the zip into C:\IntuneAppMain (or a SharePoint folder or network drive) where you will keep a repository of all your organization's packages.

Test a pre-packaged installer

This will manually test a package on your machine.

Open `C:\IntuneAppMain \7zip`

Double click `intune_command.cmd`

Choose *(D)etect* – Look for the last line of info – it should say whether you already have 7zip.

Choose *(I)nstall* – This will install 7zip

Note: *(D)etect*, *(R)equirements*, *(I)nstall*, *(U)ninstall* are the four core Intune actions for Windows packages. Here, you are able to run them manually to see what Intune does behind the scenes.

Test installing a few apps at once

Open `C:\IntuneAppMain` and run `AppsMenu_Launcher.cmd`

Choose *(I)nstall* apps

On the list of apps that pops up, ctrl – click (select) one or more apps.

Choose *(I)nstall*

The installers should run for all the apps

Prepare your Org for Intune

This will prep your org for publishing by setting up a *registered app* in Entra

Choose *(P)* to begin publishing

Choose *(O)* Prep a new Org for publishing apps

Enter your organization's primary domain name.

Follow the prompts to install the publishing app in Entra.

*Modules: Some modules need to be on the publishing machine: Microsoft.Graph and IntuneWin32App
These will be checked during the process, but you may need to install these before proceeding. Use the
(M)odules menu to install these PowerShell modules.*

Publish Apps

This will publish / update apps to your org

Choose *(P)* to begin publishing

Choose your org from the list of prepped orgs (see above)

On the list of apps that pops up, ctrl – click (select) one or more apps.

After publishing the apps, look in Intune for the Apps themselves.

Look in Entra for assignment groups starting with IntuneApp

Push (Assign) Apps to Users

Entra group assignments control how apps to endpoint machines.

These Entra groups are created automatically by the publishing process

IntuneApp Windows Users	This group is where mandatory apps get published. Put all your Windows users in this group. It can be dynamic but, for now, add a few test users.
-------------------------	---

	See <i>Publishing to all users</i> below to convert the group to a dynamic group that auto-includes all users.
IntuneApp Windows Users Excluded	This group excludes people from any publishing
IntuneApp [Appname]	Each app will have a group where you can add people that are supposed to get the app.

Note: The groups created are of type Security Group (non-email enabled) and will not send welcome messages.

Manually install an App for a User

This is how you can manually install an app (as a user).

Published apps are available to users in the *Company Portal* app and can be installed from there (no admin rights are required).

Check the *C:\IntuneApps* folder on the endpoint for logging etc.

Admins looking to manually install can copy the individual app folders to the endpoint and run *intune_command.cmd* (see above).

Main Menu

Run [Apps]\Packaged***AppsMenu_Launcher.cmd*** to load the main menu.

Do not run as admin.

```
-----
AppsMenu.ps1
Installs and publishes apps using either Intune or manual methods.

About Apps
- Intune enabled (Win32) apps are a package of files that install a product on an e
- Typically, apps are pulled from the Winget or Choco online repositories.
- Apps can also be custom scripts.
- Most Apps only require an icon (graphic) file and a settings file to indicate the
-----

AppsMenu.ps1 Main menu
-----
L - List / Create apps
  Shows the current apps. Creates app based on winget template
I - Install / Uninstall apps
  Installs (or Uninstalls or Detects) apps on the current machine
C - Copy apps (to a USB key)
  Copies apps to a portable folder for later install
P - Publish / Unpublish apps
  Publishes apps to an existing org. Create new org.
-----

Select an Action
[X] Exit [L] List [I] Install [C] Copy [P] Publish [?] Help (default is "X"):
```

List / Create Apps (L)

CreateApp.ps1

Shows the current packaged apps.

Creates an app package (if it exists in winget)

AppsCreate.ps1 Main menu

```
L - List apps (to screen)
E - Export apps list to csv file
O - Open apps folder in explorer
B - Browse winget for app ids
C - Create a new app (using wizard)
```

App creation is beyond the scope of this document.

However, WinGet apps are fairly easy to create if you follow the prompts.

Each app's settings are controlled by a .CSV file within the app folder.

Install / Uninstall apps (I)

AppsInstall.ps1

install app on the current machine.

Do not run as local administrator, it will be take care of during the installation.

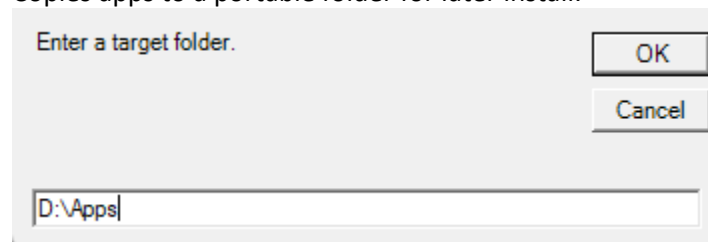
The AppGroup entries are based on a .CSV which bundles apps together for convenience.

```
Choices for 7Zip
I Install.ps1      Installs the app
U Uninstall.ps1   Uninstalls the app
D Detection.ps1   Detects if the app is already installed
R Requirements.ps1 Detects if the machine fulfills the requirements for installing the app
X Exit
Choice?
[I] Install [U] Uninstall [D] Detection [R] Requirements [X] Exit [?] Help (default is "I"):
```

Copy apps (to a USB key) (C)

AppsCopy.ps1

Copies apps to a portable folder for later install.



Enter a target folder.

OK

Cancel

D:\Apps

Publish / Unpublish apps (P)

AppsPublish.ps1

To publish your apps to an org. Do not run as local administrator.

```
AppsPublish.ps1 Publish menu
-----
C - Check local apps prior to publication
  Checks local apps for changes / updates hash (recommended for Publish option)
P - Publish apps
  Publishes apps to an og
O - Prep a new Org for publishing apps
  Creates an entry in AppsPublish_Orgs.csv
M - Install / upgrade modules
  Installs the PowerShell modules (required for Publish option)
-----
```

Check local apps prior to publication (C)

This makes sure the hash values for the current apps (in the local folder) are up to date. The hashes are important because they are checked against the online hash to see if any package changes have been made.

- The check process is run automatically prior to publishing
- Unmatched hashes will signal that an update is required.

Publish apps (P)

Use this to select an org and publish apps.

- Publishing means uploading packaged apps to Intune.
- Depending on Group memberships, this might also mean the *immediate push* of the apps to client devices.
- Prior to publishing
 - Make sure you use the Prep option (O) menu below
 - Powershell Modules. To publish from a particular machine, you must install the required modules. Use the Install module (M) menu first.
- Admin Credentials
 - When publishing, you will need enter admin credentials to Endpoint Manager.
 - If your org specifies a target group to publish, you will need to enter AZAD admin credentials also.
 - successive runs will not ask for creds.
- When ready, choose the apps to publish from the list. The list will show which apps already exist in the org and whether an update is needed.
- Groups
 - Each published app will auto-create an Entra group which is tied to the app.
 - Each published app may also be a member of the general Entra group for Windows users (depending on the *PublishToOrgGroup* setting within the app CSV).
 - The Entra group memberships controls who gets the app

Prep a new Org for publishing apps (O)

Creates an entry in AppsPublish_Orgs.csv

- Creates the 2 groups mentioned in the Org Setup steps (above)
- Creates the **IntuneApp Publisher** Entra App

Install / upgrade modules (M)

AppModules.ps1

```
Modules:
1] IntuneWin32App
2] Microsoft.Graph
3] MicrosoftTeams
4] ExchangeOnlineManagement
-----
Which module? (blank to exit):
What do you want to do.
[X] Exit this module [R] Relaunch as Admin [C] Check [U] Uninstall [I] Install
```

This will make sure your machine has up to date module versions installed.

- If in doubt, it is best to use the *Check* feature to make sure each module is up to date.
- While modules can be installed as the user (non admin), it's better to install them as admin so they all users of the machine will see the correct version.
- The basic steps are

- Check (as user) - it will let you know if there's a user-level or an admin-level module installed.
- If update is required: Press R - Relaunch as Admin then U – Uninstall then I – Install
- Do this for each module. But Intune publishing only needs *IntuneWin32App* and *Microsoft.Graph* modules.

Common Actions

Publishing to All Users

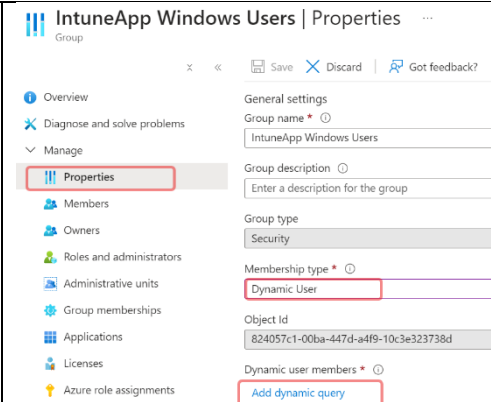
If an App is to be an Intune Required App for all users, two items need to be checked.

The app's `intune_settings.csv` file needs to have `PublishToOrgGroup=TRUE`. This will ensure the *IntuneApp Windows Users* group is assigned to the app as required.

The *IntuneApp Windows Users* group membership must include the users.

Normally, the *IntuneApp Windows Users* group should include everyone in the organization. Manually maintain this group as users join the organization will be cumbersome.

Convert the *IntuneApp Windows Users* group from an assigned group where users are manually added, to a dynamic group that automatically includes everyone.

<p>Open the group in Entra and click on Properties</p> <p>Change the type from Assigned to Dynamic User</p> <p>Click Add Dynamic Query click Edit Query and paste the query from below.</p>	
---	---

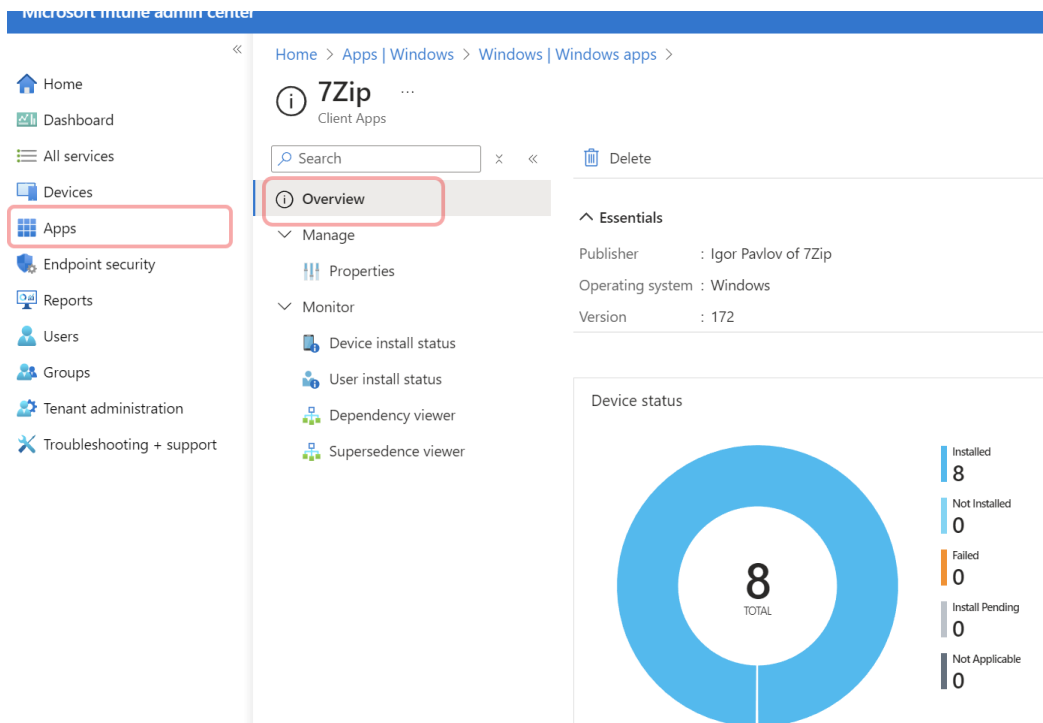
This query means: *All Licensed users, that are not disabled, and not any of these named users*

```
(user.assignedPlans -any (assignedPlan.servicePlanId -ne "[noplanplaceholder]" -and
assignedPlan.capabilityStatus -eq "Enabled"))
-and (user.accountEnabled -eq True)
```

Checking App publishing status in Intune

Open Intune > Apps > Windows

[App] > Overview



Other Intune Apps (Non IntuneApp Apps)

This section covers apps that would not be installed with the IntuneApp system. These apps are easily (and better) installed natively by Intune. They are mentioned here because they are commonly needed.

Microsoft Store App: Company Portal

Overview

Company Portal is a Windows app that

- Shows users all available apps and what has been installed
- Allows users to install apps at a time convenient to them
- Allows the installation to happen in system context or user context (as packaged) without the user needing admin rights.

Here's how to add the company portal app to everyone.

Steps

Go to the Windows App List [link](#)

Add > **Microsoft Store app (new)** > **Select**

Search for **Company Portal (UWP)**

Leave all defaults

Upload icon (logo) from here: CompanyPortalIcon.png [link](#) or from the store [link](#)

Assign to: **Required** > **Add All users**

Microsoft 365 Apps (MS Office)

Overview

Outlook, Excel, Word, PowerPoint, Teams, etc do not require packaging. There is a built-in Endpoint Manager App for Microsoft 365 Apps

Steps

Intune Admin > Apps > Windows ([link](#))

> Add > Microsoft 365 Apps > Windows 10 and later

Choose all defaults except:

Company Portal featured app: **Yes** > Next

Default file format: **Office Open XML Format** (.docx/.xlsx/.pptx)

Update Channel: **Monthly Enterprise**

Shared Computer Activation: **No**

(Yes means it won't count against your 5, but it will require more frequent check-ins via password)

Install background service for Microsoft Search in Bing: **No**

(Yes means it installs a background service that helps determine whether a Microsoft Search in Bing extension for Google Chrome is installed on the device)

Languages: **EN-US**

> Next

Install as Required: for all Users (Note: this is safe to do) (or same user groups as above)

Technical Details

Log Files

Look in C:\IntuneApp for debug files

Look in \Utils for tools

Intune Settings

Each App package has an intune_settings.csv file to control its publication.

Name	Value	Comment
AppName	7Zip	(Required) Base package name (e.g 7zip) (Remove ! Character which hides sample apps)
AppVersion	111	(Optional) Package version. Whole numbers beginning with 100. Nothing to do with product version. (e.g. 100 for 7zip-v100) (Recommended)
AppInstaller	winget	(Required) winget,choco,exe,msi,ps1,cmd. If winget, choco is used, leave AppUninstallName blank unless you want to uninstall an additional app
AppInstallName	7zip.7zip	(Required) winget appid (case sensitive) or choco appid or filename of .exe or .msi or .ps1 or .cmd (wildcard OK). See https://winstall.app or https://community.chocolatey.org/packages for package ids.
AppInstallArgs		(Optional) Prefix with ARGS: Installer arguments for msi,ps1,exe,cmd (for msi usually /quiet or /q /norestart) (for

		ps1 with multi param try -var1 xyz -var2 pdq named format, else entire value is first param)
<i>AppDescription</i>		(Required) Info for company portal
<i>AppUninstallName</i>		(Optional) winget name or winget id (Use 'winget list' to show them) to detect/uninstall (also done prior to install). This is in addition to the normal choco,winget uninstall. This field is required for non-winget packages, unless customcode.ps1 is used, as it is the only way to detect an app
<i>AppUninstallVersion</i>	22	(Optional) Winget product version (e.g. 4.5.0) below which to the app is considered not detected. Below this version will be uninstalled / reinstalled. Blank=All versions are OK and will not be reinstalled. Use current version to repackage an app to upgrade everyone to latest version.
<i>AppUninstallProcess</i>	*7zip*	(Optional) Processes to end prior to uninstall. (e.g. Acrobat*) Show running names via powershell: get-process -Name Acrob*. Wildcards can be used
<i>SystemOrUser</i>		(Required) System to install as system, User to install as user
<i>Publisher</i>	Igor Pavlov of 7Zip	(Required) Info for company portal
<i>AppInstallerDownload1URL</i>		Enter a URL to download into install folder before installer starts. Can be a public share from Google Drive (drive.google.com). Zip files will be extracted automatically.
<i>AppInstallerDownload1Hash</i>		Enter the file hash (optional) PSH: gci *.zip Get-FileHash -Algorithm SHA256 Select-Object Hash, @{n="File";e={Split-Path \$_.Path -Leaf}}
<i>AppInstallerDownload2URL</i>		
<i>AppInstallerDownload2Hash</i>		
<i>RestartBehavior</i>	allow	Allow Installer to Restart (Default is allow)
<i>Developer</i>		Info for company portal
<i>Owner</i>		Info for company portal
<i>Notes</i>		Info for company portal
<i>InformationURL</i>	https://www.7zip.org/?ref=wininstall	Info for company portal
<i>PrivacyURL</i>		Info for company portal
<i>CompanyPortalFeaturedApp</i>	FALSE	(Required) Company Portal Featured (Default is FALSE)
<i>AvailableInCompanyPortal</i>	TRUE	(Required) Company Portal Availability (Default is TRUE)
<i>PublishToOrgGroup</i>	FALSE	(Required) App will be pushed immediately to the PublishToGroup group from AppsPublish_OrgList.csv (as a required app). False means do not push app to that group. Independent of this, apps are always published to its own IntuneApp [appname] group (Default is FALSE)
<i>AppVar1</i>		Custom var useable by ps1 files

<i>AppVar2</i>	Custom var useable by ps1 files
<i>AppVar3</i>	Custom var useable by ps1 files
<i>AppVar4</i>	Custom var useable by ps1 files
<i>AppVar5</i>	Custom var useable by ps1 files

Internal dependencies

This script publishes a Win32 App (intunewin) [link](#)

Uses Microsoft Graph modules (and others) that are auto loaded at runtime

This script auto-loads this github module (IntuneWin32App [link](#)) by NickolajA

Which auto-loads this Microsoft module (Microsoft Win32 Content Prep Tool [link](#))

Intune Tracing

CMTrace.exe

Use the Configuration Manager Trace Log Tool to view log files.

C:\ProgramData\Microsoft\IntuneManagementExtension\Logs

AgentExecutor.log

IntuneManagementExtension.log

IntuneApp files

Package File	Description	Notes
intune_settings.csv	App Name and version	Later versions will automatically supersede (uninstall) prior versions
intune_install.ps1	Silent installation script	0,1707=Success 3010=Soft Reboot 1641=Hard Reboot 1618=Retry
intune_uninstall.ps1	Silent uninstall script	
intune_requirements.ps1	Reports if your app applies to this user/system. Apps that fail the requirement rule will display as <i>Device Status: Not Applicable</i> in Endpoint Manager (and will not be installed)	Script must output string "OK" for requirements to pass. Run this script using the logged on credentials: Yes, No Look for this integer eq 0 See link . No access to other package files. Must be a standalone script.
intune_detection.ps1	Detects if your app is installed. Only Apps that are 'not detected' will be installed. Will run as the machine or User depending on the package type.	When the script exit code is 0 and STDOUT contains any data, the app is detected. See link No access to other package files. Must be a standalone script.
intune_icon.png	Icon that's displayed in Endpoint Manager and Company Portal	