Content

1. Introduction

1.1 Purpose

The document describes how to install the latest version of applications located in Winget repository using an Intune Win32 app as a distribution method.

Winget currently has over 5900 applications available for installation.

It is desirable to publish an application in Intune once, and not have to constantly create a new publish app for each time the application published comes in a newer version.
This method performs this.

It is desirable that Winget applications that require UAC prompt for updates are automatically updated to the latest version available through Winget. This method performs this.

(This will then apply to all applications that are available in Winget repository and that are installed via system user on the user's PC).

- It is appropriate that programs installed are the latest version available due to security and functionality.

- It is appropriate for applications to be updated to the latest version available due to security and functionality.

- Functionality has also been added so that the existing version of specified programs is retained in the current version. As these do not need to or should not be upgraded.

Intune W32 app deployments created will always install the latest version of the app from Winget repository.

PIN application you do not want Winget to update with remediation script see “Prod – PIN Winget Programs”.

“Winget Update programs”.
Remediation script will upgrade all applications installed in System Context with newest version of your programs from winget, but not the programs you have Pinned.

2. WinGet (Windows Package Manager)

Winget is used to Install\Uninstall\Update\Pin programs installed through the Winget source.

Windows Package Manager | Microsoft Learn

https://learn.microsoft.com/en-us/windows/package-manager/winget/

Windows Package Manager - Wikipedia, the free encyclopedia

2.1 Winget ADMX Policies/CSP Policies.

It is possible to import ADMX Policies into Intune.

The file to be used is named "C:\Windows\PolicyDefinitions\DesktopAppInstaller.admx"
Don't see reason for this at this time.

DesktopAppInstaller Policy CSP - Windows Client Management | Microsoft Learn

2.2 How to find out if the program is available in WinGet source

Open Powershell and type `winget search -q `"`" >>c:\data\wingetApps.txt`

You will then be able to search c:\data\wingetApps.txt for applications.
There are now over 5900 applications in Winget source.

Tip: You need to retrieve the following *ID* and "Version".

Sometimes the full ID name does not appear.

Here's an example from wingetapps.txt as an example:

Name	Id	Version	Match	Source
PL/SQL Developer	AllroundAutomations.PLSQLDevelopâ€¦	15.0.		Winget

Then execute from powershell.
winget search -q "PL/SQL Developer"

You will then be listed with the full ID name.

```
Name                       Id                            Version      Source
-------------------------------------------------------------------------------
PL/SQL Developer AllroundAutomations.PLSQLDeveloper  15.0.3       winget
```

2.2.1 How to list your installed programs and see if they are available in WinGet source.

Run “Winget List”

You will se what programs you have installed on your computers and list those available in Winget Source.

Name:	ID:	Version:	Source:
7-Zip 23.01 (x64 edition)	7zip.7zip	23.01.00.0	winget
Notepad++ (64-bit x64)	Notepad++.Notepad++	8.6.5	winget
VLC media player	VideoLAN.VLC	3.0.20	winget

2.3 How to find out information about the Winget application, and whether to install the application in User Contex or System Contex in Intune.

Download the program and try and install it on local PC.

Run the following from PowerShell:

- Replace AppID with ID of the application from the Wingapps.txt in the command below.
- Replace the version number with the version number in Wingetapps.txt in the command below.

```
winget download --query 'AppID' --version version number --download-directory
c:\data
```

Try and install the program on your local PC.

If a UAC (Administrator Prompt) appears during installation, the application should be deployed in Intune via System Context.

Also, check if this is the program you later want to publish to Intune.

NB: Also look at the *.yaml file that was also downloaded. This contains information about where the applications are downloaded from, etc. "PackageUrl". Important for verifying that the application has a source you trust.

Or run winget show -q `AppID.`

Find out the path to the program EXE file. This can later be used as the Detection Rule in the Intune app publication, as the applications can be updated to newer versions.

ID	System Context	User Context	EXE file path	AppName
`Notepad++.Notepad++`	Yes	No	C:\Program Files\Notepad++\notepad++.exe	Notepad++ - Latest version

ID= see ID column in WingetAppt.txt

AppName= Application name (must be able to be saved as log file name)
Should the application be deployed in System Contxt or User Context = Yes/No

Path to EXE file e.g.= "C:\Program Files\Notepad++\notepad++.exe"

2.4 Scripts

These are installation and uninstall scripts that must be customized for each application published in Intune.

2.4.1 Install.ps1

```
## ----- Installed App with the latest version. ----------
##

## ----- What To Do ----------
## Search for application by running from PowerShell winget search %Name%
## Or to get all Winget applications run from PowerShell to an output file run
winget search -q `"`" >>c:\data\wingetApps.txt
## Replace %Name% with name on application you want to install.
## Replace $AppID with ID of application.
## Replace $AppName with NAME of application.


## ----- How To Download App just for testing and possible test downgrade the app
version (testing purpose). --------
## Run winget show -q AppID --versions
## Example run winget show -q Notepad++.Notepad++ --versions
## Run winget download --query 'AppID' --version versionnumber --download-directory
c:\data
## Example run winget download --query 'Notepad++.Notepad++' --version 8.5.7 --
download-directory c:\data
## Take a look at files downloaded one file will contain lot of info about the app
itself.
## Also when downloading, se source for download, if you trust it.


## Variables
$AppID = 'Notepad++.Notepad++'
$AppName = 'Notepad++.Notepad++'
## Variables End


# Who is running the script
$RunningAs=c:\windows\system32\whoami.exe
Write-Output "The script is Running as User $RunningAs"

####################################################
#### Section if installation is Running as SYSTEM.
####################################################
if ($RunningAs -like "*system*")
{
  # Start transcript to system path
  Start-Transcript -Path "$env:ALLUSERSPROFILE\log\$AppName Install.log"

  Write-Output "The script is Running as User $RunningAs"

  ## Help System to find winget.exe
  Write-Output "$RunningAs is provided with path to winget.exe"
  $JBNWinGetResolve = Resolve-Path "C:\Program
Files\WindowsApps\Microsoft.DesktopAppInstaller_*_x64__8wekyb3d8bbwe\winget.exe"
  $JBNWinGetPathExe = $JBNWinGetResolve[-1].Path

  $JBNWinGetPath = Split-Path -Path $JBNWinGetPathExe -Parent
  set-location $JBNWinGetPath

Write-Output "Path til Winget $JBNWinGetPath"

  ## Install parameters if app requires UAC prompt, and running as System.
  ## Install newest version of $AppID in system Context. For installation in system
Context use:  .\winget install -e --id $AppID --silent --accept-source-agreements -
-accept-package-agreements
 .\Winget -v
 .\Winget list -e --id $AppID --accept-source-agreements
 .\winget install -e --id $AppID --silent --accept-source-agreements --accept-
package-agreements
}
```

```
else

##################################################
#### Section if installation is Running as USER.
##################################################
{
  # Start transcript to user path
  Start-Transcript -Path "$env:APPDATA\$AppName Install.log"

  Write-Output "The script is Running as User $RunningAs"

  ## Install newest version of $AppID in user Context. For installation in User
Context use: .\winget install -e --id $AppID --silent --scope=user --accept-source-
agreements --accept-package-agreements
  Winget list -e --id $AppID --accept-source-agreements
  winget install -e --id $AppID --silent --scope=user --accept-source-agreements --
accept-package-agreements
Write-Output "$LASTEXITCODE"
}

Stop-Transcript
```

2.4.2 Uninstall.ps1

```
## ----- Installed App with the latest versjon. ----------
##

## ----- What To Do ----------
## Search for application by running from PowerShell winget search %Name%
## Or to get all Winget applications run from PowerShell to an output file run
winget search -q `"`" >>c:\data\wingetApps.txt
## Replace %Name% with name on application you want to install.
## Replace $AppID with ID of application.
## Replace $AppName with NAME of application.


## ----- How To Download App just for testing and possible test downgrade the app
version (testing purpose). --------
## Run winget show -q AppID --versions
## Example run winget show -q Notepad++.Notepad++ --versions
## Run winget download --query 'AppID' --version versionnumber --download-directory
c:\data
## Example run winget download --query 'Notepad++.Notepad++' --version 8.5.7 --
download-directory c:\data
## Take a look at files downloaded one file will contain lot of info about the app
itself.
## Also when downloading, se source for download, if you trust it.


## Variables
$AppID = 'Notepad++.Notepad++'
$AppName = 'Notepad++.Notepad++'
## Variables End


# Who is running the script
$RunningAs=c:\windows\system32\whoami.exe
Write-Output "The script is Running as User $RunningAs"

####################################################
#### Section if installation is Running as SYSTEM.
####################################################
if ($RunningAs -like "*system*")
{
  # Start transcript to system path
  Start-Transcript -Path "$env:ALLUSERSPROFILE\log\$AppName Install.log"

  Write-Output "The script is Running as User $RunningAs"

  ## Help System to find winget.exe
  Write-Output "$RunningAs is provided with path to winget.exe"
  $JBNWinGetResolve = Resolve-Path "C:\Program
Files\WindowsApps\Microsoft.DesktopAppInstaller_*_x64__8wekyb3d8bbwe\winget.exe"
  $JBNWinGetPathExe = $JBNWinGetResolve[-1].Path

  $JBNWinGetPath = Split-Path -Path $JBNWinGetPathExe -Parent
  set-location $JBNWinGetPath

Write-Output "Path til Winget $JBNWinGetPath"

  ## Install parameters if app requires UAC prompt, and running as System.
  ## Install newest version of $AppID in system Context. For installation in system
Context use:  .\winget install -e --id $AppID --silent --accept-source-agreements -
-accept-package-agreements
 .\Winget -v
 .\Winget list -e --id $AppID --accept-source-agreements
 .\winget install -e --id $AppID --silent --accept-source-agreements --accept-
package-agreements
}

else

####################################################
#### Section if installation is Running as USER.
####################################################
{
  # Start transcript to user path
```

```
  Start-Transcript -Path "$env:APPDATA\$AppName Install.log"

  Write-Output "The script is Running as User $RunningAs"

  ## Install newest version of $AppID in user Context. For installation in User
Context use: .\winget install -e --id $AppID --silent --scope=user --accept-source-
agreements --accept-package-agreements
  Winget list -e --id $AppID --accept-source-agreements
  winget install -e --id $AppID --silent --scope=user --accept-source-agreements --
accept-package-agreements
Write-Output "$LASTEXITCODE"
}

Stop-Transcript
```

2.5 How to Create an Intune Win32 Application

Replace the following in Install.ps1 and Uninstall.ps1.

```
$AppID = ' Notepad++.Notepad++'
$AppName = 'Notepad++.Notepad++ - Latest version'
```

`$AppID` variable with ID name from Wingetapps.txt

`$AppName` variable with the desired name for application. The name is used as Log file for application installation and uninstallation, Then use a name that can be saved as file. Recommended the same name as you give the app in Intune. like "`Notepad++.Notepad++ - Latest version'`"

NB It is important that the following are tested before the application package is created.

- Is it the right program.
- Installs the application in User or System Contex.
    - Deploy App in Install behavior like User or System.
- Does the program have its own Autoupdate function that should\can be used.
    - Like Office, Edge,Chrome,Teams ++. Then use their own update method. And PIN these applications so Winget does not update these application.

Create an IntuneWin file with the same name as the application containing the 2 scripts.

Step:

- Find application available in Winget source, Test installer application.
- Edit Install.ps1 and Uninstall.ps1 and replace `$AppID and $AppName`
- Create a *. Intunewin file.
- At Intune create a Windows app (Win32).
    - Name like "*Application Name* - Latest Version"
    - App version. Latest Version
    - Install command = powershell.exe -ExecutionPolicy Bypass -WindowStyle hidden -file .\Install.ps1
    - Uninstall command = powershell.exe -ExecutionPolicy Bypass -WindowStyle hidden -file .\UnInstall.ps1
    - Install behavior = System or User (Dependent og App install demands UAC prompt).
    - Detection Rule /File = Full path to program exe file (maybe). "File or Folder exist"
- Does the program have its own auto update function that works satisfactorily. Or is there a requirement that the program should not be upgraded to the latest available version in Winget.
    - Then edit the Intune remediation script "Prod - PIN Winget Programs" and enter the ID of the application name. (Later in this document)

2.5.1 Log files for installation\uninstall of Intune application published.

If the application is installed in System Contex:

```
"$env:ALLUSERSPROFILE\log\$AppName Install.log"
```

If the application is installed in User Contex:

```
"$env:APPDATA\$AppName Install.log"
```

If System Contex uninstalls the application:

```
"$env:ALLUSERSPROFILE\log\$AppName UnInstall.log"
```

If the application is uninstalled in User Contex:

```
"$env:APPDATA\$AppName UnInstall.log"
```

3. How to always keep the same version for an application.

Sometimes you want to keep the same version for an application that is being installed.
i.e. it should not be updated to a newer version.

- This may be because the program has its own auto update functionality that you want to keep. (Office\Adobe\Edge +++). It is then important to PIN them.

- Some application must be at the same version due to versions on backend servers.

See chapter on Remediations Script – "Prod - PIN Winget Programs"

Here's an example for ID=TeamViewer.TeamViewer.Host

```
winget pin add --id TeamViewer.TeamViewer.Host
```

To list out which programs are pinned run:

```
Winget Pin List
```

NB: You will not be able to list which programs are pinned by running the command as normal user.
View log file generated from "Prod – Pin Winget Programs". It runs the script in System Contex.

```
Name Id Version Source Pin type
-------------------------------------------------------------------------
TeamViewer Host TeamViewer.TeamViewer.Host 15.48.5.0 winget Pinning
```

Otherwise, see Winget pin

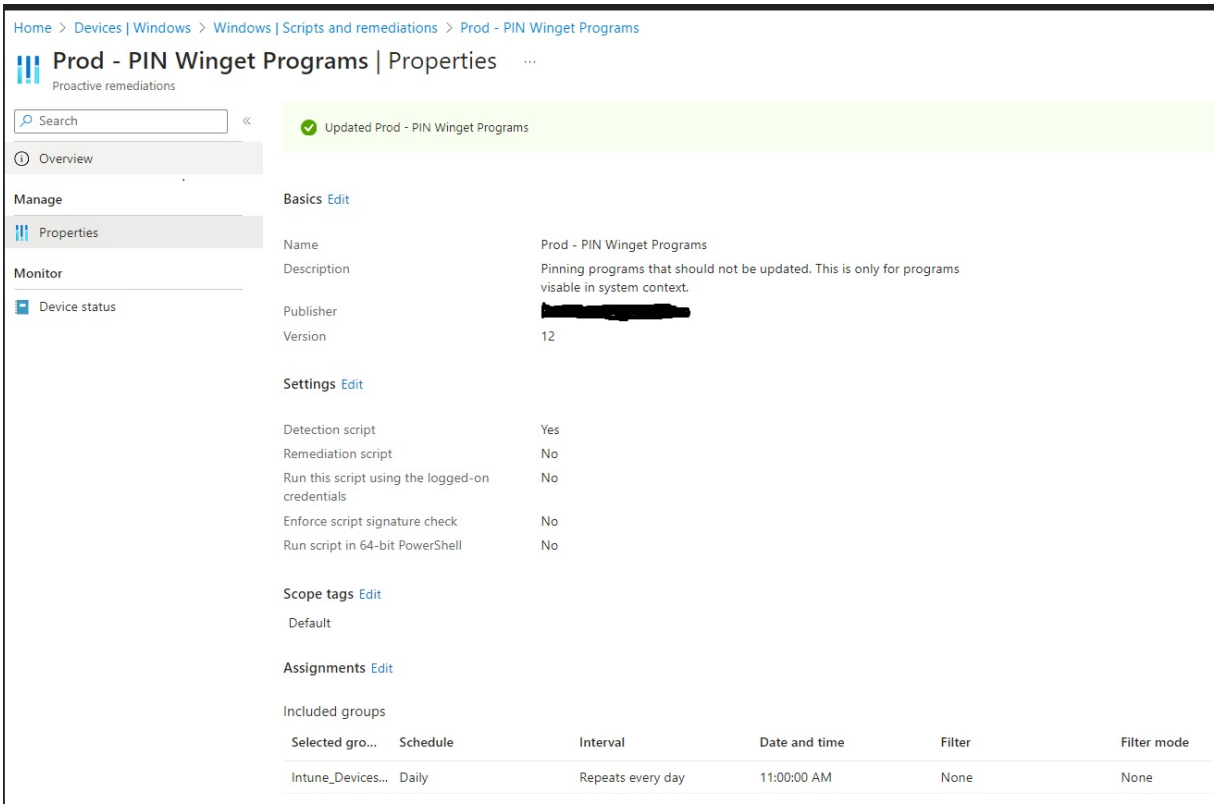
3.1 Remediations Script – "Prod - PIN Winget Programs"

It is desirable that some of the programs that are installed and located in Winget source are not upgraded to newer versions.

This is performed via the Intune remediation script «Prod - PIN Winget Programs".

This is performed at 11:00AM every day per PC.

Properties of “Prod – PIN Winget programs” that you must make using script below.



NB If you want to make changes: Copy out the current remedation script and edit it.

Edit the script with other IDs (applications) if needed.

NB: You should also change $Version

```
## ----- Pinning Winget applications to prevent versions updates  ----------
##
$PSDefaultParameterValues = @{ '*:Encoding' = 'utf8' }
$Version = "Version 1.0"

# Who is running the script
$RunningAs=c:\windows\system32\whoami.exe

if ($RunningAs -like "*system*")
{
##################################################
#### Section if installation is Running as SYSTEM.#
```

```
##################################################

    Start-Transcript -Path
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log"

    ## Help System to find winget.exe
    Write-Output "$RunningAs is provided with path to winget.exe"
    $JBNWinGetResolve = Resolve-Path "C:\Program
Files\WindowsApps\Microsoft.DesktopAppInstaller_*_x64__8wekyb3d8bbwe\winget.exe"
    $JBNWinGetPathExe = $JBNWinGetResolve[-1].Path
    $JBNWinGetPath = Split-Path -Path $JBNWinGetPathExe -Parent
    set-location $JBNWinGetPath

# $PSDefaultParameterValues

Write-Output "Pinning winget programs"

Write-Output "Path to Winget.exe is $JBNWinGetPath"



Stop-Transcript


Write-Output "Winget.exe Version is" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
.\winget.exe -v | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append -
encoding ascii



##################################################
####    PINNING programs                        #
##################################################


Write-Output "Pinning Microsoft.Teams" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
.\winget.exe pin add --id Microsoft.Teams --accept-source-agreements | Out-File -
FilePath "$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
-encoding ascii


Write-Output "Pinning Microsoft.Office" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
.\winget.exe pin add --id Microsoft.Office --accept-source-agreements | Out-File -
FilePath "$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
-encoding ascii

Write-Output "Pinning Microsoft.OneDrive" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
.\winget.exe pin add --id Microsoft.OneDrive --accept-source-agreements | Out-File
-FilePath "$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -
Append -encoding ascii

Write-Output "Adobe.Acrobat.Reader.64-bit" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
.\winget.exe pin add --id Adobe.Acrobat.Reader.64-bit --accept-source-agreements |
Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append -
encoding ascii

Write-Output "Microsoft.Edge" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
.\winget.exe pin add --id Microsoft.Edge --accept-source-agreements | Out-File -
FilePath "$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
-encoding ascii

Write-Output "Google.Chrome" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
.\winget.exe pin add --id Google.Chrome --accept-source-agreements | Out-File -
FilePath "$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
-encoding ascii
```

```
##################################################
####     Listing PINNED programs                 #
##################################################


Write-Output "Listing Pinned programs" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append
.\winget.exe pin list --accept-source-agreements | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\PIN_Winget_Programs_$Version.log" -Append -
encoding ascii



exit 0
}

else

{
##################################################
#### Section if installation is Running as USER. #
##################################################

    Start-Transcript -Path "$env:APPDATA\Winget\PIN_Winget_Programs_$Version.log"

    ## List programs installed and show versions in Winget.
    Winget List --accept-source-agreements

}
Stop-Transcript
exit 0
```

3.1.1 Log file: "c:\Progradata\log\Winget\PIN_Winget_Programs_$Version.log"

4. Winget update application version.

It is important to update applications to latest version due to security and functionality.

Make a remediation script.

This remediation script runs in System Contex, i.e. it only updates programs installed in System Context on PC. It then updates apps who require a UAC prompt for updates (Installed with Administrator rights) and are in the Winget repository.

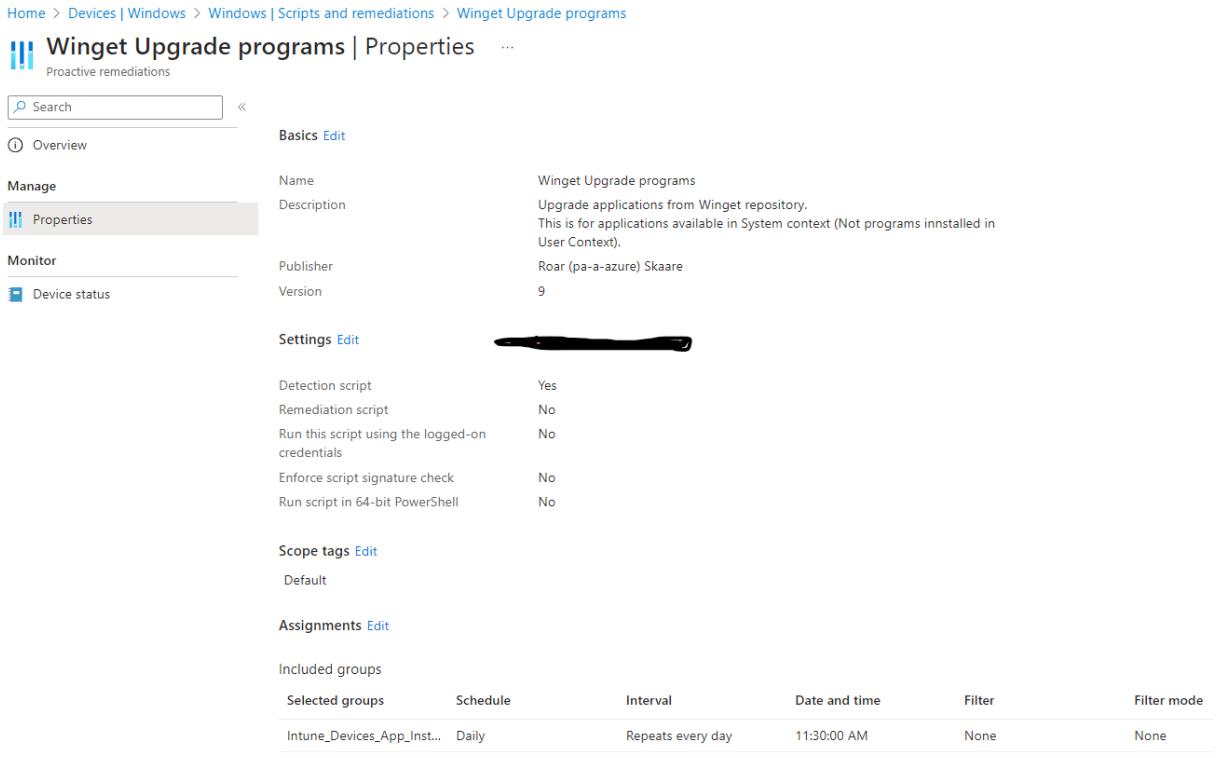
Important: It will not update applications installed in User context.

There is little need to update the Winget applications that are installed in User context.

This as those Apps who have been tested updated themselves, or user can do it themself.

Remediation script:

Winget Upgrade Programs



This will Repeats every day at 11:30AM (After PIN programs has been run).
It runs in System Context (Not as logged on user).

The script will also delete files older than 30 days from c:\programdata\log\Winget.

4.1 Remedation Script - Winget-Update-Programs.

```
## ----- Updating Winget applications to newer versions from Winget repository  ---
-------
##

# Using $Time to create logging to Log files to see Version history.
$TimeSystemTime = Get-Date -Format "yyyy/MM/dd HH"
## If using US format. Replace / with . in format. Otherwise you are getting
folders insted of a singel file.
$Time = $TimeSystemTime.replace('/','.')

Start-Transcript -Path
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log"


$PSDefaultParameterValues = @{ '*:Encoding' = 'utf8' }


# Who is running the script
$RunningAs=c:\windows\system32\whoami.exe

if ($RunningAs -like "*system*")
{
##################################################
#### Section if installation is Running as SYSTEM.#
##################################################

    ## Help System to find winget.exe
    Write-Output "$RunningAs is provided with path to winget.exe"
    $JBNWinGetResolve = Resolve-Path "C:\Program
Files\WindowsApps\Microsoft.DesktopAppInstaller_*_x64__8wekyb3d8bbwe\winget.exe"
    $JBNWinGetPathExe = $JBNWinGetResolve[-1].Path
    $JBNWinGetPath = Split-Path -Path $JBNWinGetPathExe -Parent
    set-location $JBNWinGetPath


    ## Stopping transcript because remediation script does not provide output like
regular PowerShell scripts does.
    Stop-Transcript


    Write-Output "Path til Winget $JBNWinGetPath" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii


    ## List programs installed and show versions in Winget.
    Write-Output "Running .\Winget List --accept-source-agreements" | Out-File -
FilePath "$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append
-encoding ascii
    .\Winget List --accept-source-agreements | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii

    Write-Output "------------------" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii
    Write-Output "Listing already Pinned programs set by intune Remediation script,
Prod - PIN winget Programs" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii
    .\Winget pin List --accept-source-agreements | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii

    Write-Output "------------------" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii
```

```
    Write-Output "Update programs to newer versions from Winget." | Out-File -
FilePath "$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append
-encoding ascii
    .\Winget upgrade --all --silent --accept-source-agreements --accept-package-
agreements | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii

}

else

{
#####################################################
#### Section if installation is Running as USER.  #
#####################################################
Stop-Transcript
Write-Output "Running script as user (Not SYSTEM), no winget update will be
executed" | Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii
}

#####################################################
#### Delete files older than 30 days.              #
#####################################################
Write-Output "Delete files from C:\ProgramData\log\Winget\ older than 30 days. " |
Out-File -FilePath
"$env:ALLUSERSPROFILE\log\Winget\Winget_Update_Programs_$Time.log" -Append -
encoding ascii
Get-ChildItem "C:\ProgramData\log\Winget\" -Recurse -File | Where CreationTime -lt
(Get-Date).AddDays(-30)  | Remove-Item -Force
```

4.1.1 How to see which programs have been automatically upgraded.

View log files
"C:\ProgramData\log\Winget\Winget_Update_Programs_%YEAR%.%MM%.%DD%.11..log"

Some programs will not be upgraded due to being pinned (keep current version, not upgrade) ref remediation script "**Prod - PIN Winget Programs**"

Applications installed in system context and are in winget repository, If there is a newer version of the program will be attempted to be updated.

4.1.1.1 Log files
"C:\ProgramData\log\Winget\Winget_Update_Programs_%YEAR%.%MM%.%DD%.11..log"