**DotNET 7 Deployment**

This repository contains exe for the installation file for .Net 7 and an Uninstall Powershell script. The deployment package includes the following file:

* dotnet-sdk-7.0.302-win-x64.exe
* Uninstall.Ps1 (Please see below for the PowerShell Script syntax)

**Deployment Steps**

1. dotnet-sdk-7.0.302-win-x64.exe file in an empty folder along with the Uninstall Ps1.
2. Package the folder as a .intunewin file using the IntuneWinAppUtil.exe tool provided by Microsoft.
3. Upload the .intunewin file to the Intune portal as a Win32 app.
4. Configure the following settings in the Intune Win32 app deployment:

**Install command:**

dotnet-sdk-7.0.302-win-x64.exe /install /quiet /norestart

**Uninstall command*:***

*Uninstall.ps1*

**Detection rule:**

**Rule Type:** *File*

**Path:** C:\Program Files\dotnet\sdk

**File or folder:** *7.0.302 \*****(Folder)\****

**Detection Method:** *File or folder exists*

Assign the app to the desired users or devices.

**Uninstall.Ps1:**

|  |
| --- |
| # Define the version to be removed  $versionToRemove = "7.0.302"  # Define the paths for .NET Core SDK and Runtime installations  $sdkPath = "C:\Program Files\dotnet\sdk"  $runtimePath = "C:\Program Files\dotnet\shared\Microsoft.NETCore.App"  # Construct the paths for the specific version  $sdkVersionPath = Join-Path -Path $sdkPath -ChildPath $versionToRemove  $runtimeVersionPath = Join-Path -Path $runtimePath -ChildPath $versionToRemove  # Check and remove the SDK version  if (Test-Path -Path $sdkVersionPath) {  Remove-Item -Path $sdkVersionPath -Recurse -Force  Write-Output "SDK version $versionToRemove has been removed."  } else {  Write-Output "SDK version $versionToRemove not found."  }  # Check and remove the Runtime version  if (Test-Path -Path $runtimeVersionPath) {  Remove-Item -Path $runtimeVersionPath -Recurse -Force  Write-Output "Runtime version $versionToRemove has been removed."  } else {  Write-Output "Runtime version $versionToRemove not found."  } |