**A Project Documentation**

**on**

**“Building An MSIX Package with Framework Package Dependencies”**

**By**

**Kamran Akmal (34958)  
Gaddapara Chenchu Umasree (34935)**

**Tejas Madan Zinjade (34957)**

**Shrawani Balwadkar (34932)**

**INTERNAL GUIDE**

**Trainer: Harinya Mam**

**Submitted To**



**BUILDING AN MSIX PACKAGE WITH FRAMEWORK PACKAGE DEPENDENCIES**

* **Objectives**

The primary objective of this project is to create an MSIX package that includes framework packages like VCLibs or .NET as dependencies. This ensures that when the application is deployed, all necessary runtime components are also installed, guaranteeing that the application functions correctly on the user's system.

This approach provides several key benefits:

* **Reliability:** Prevents application failures due to missing dependencies on the target machine.
* **Simplified Deployment:** Streamlines the installation process by bundling all necessary components into a single, declarative package.
* **Improved User Experience:** Ensures a smooth "out-of-the-box" experience for the end-user without requiring them to manually install additional frameworks.
* **Centralized Management:** Allows for easier management and updating of both the application and its dependencies through the MSIX container.
* **Project Requirements**

**Hardware Requirements**

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **Component** | **Minimum Specs** | **Notes** |
| **CPU** | x64, 2+ cores | Required for packaging tools. |
| **RAM** | 8GB (16GB recommended) | More memory is beneficial for virtualization. |
| **Storage** | 20GB free space | For storing the package and related artifacts. |
| **OS** | Windows 10/11 (Pro/Ent) | Necessary for MSIX Packaging Tool & Hyper-V. |

**Software Requirements**

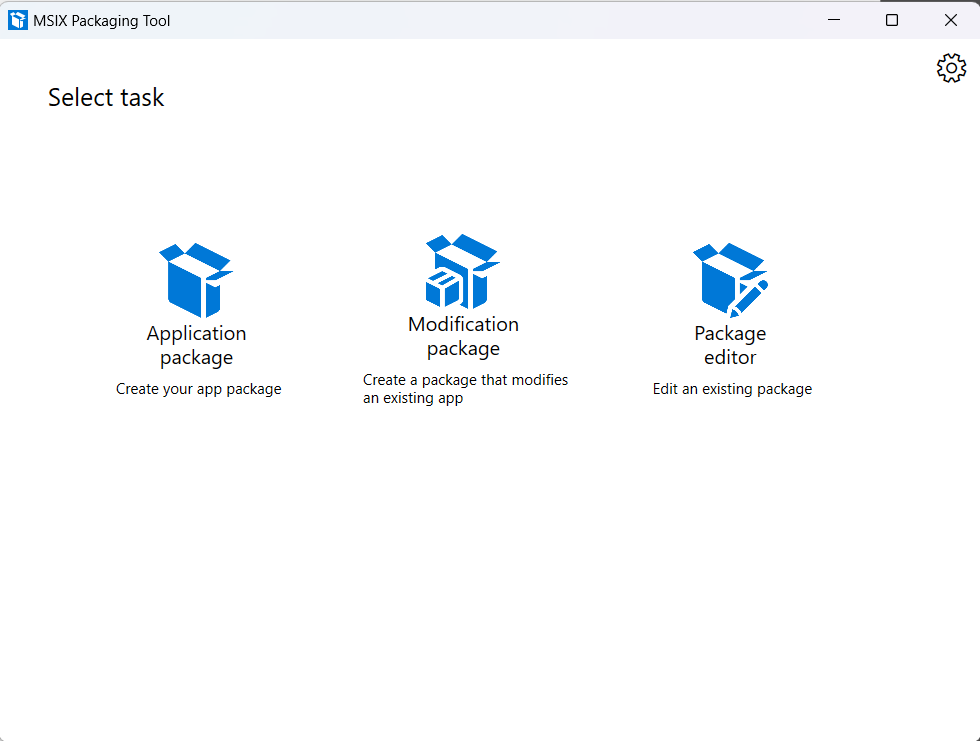
|  |  |  |
| --- | --- | --- |
| **Tool** | **Purpose** | **Download/Source** |
| **MSIX Packaging Tool** | To create the initial MSIX package. | Microsoft Store |
| **Advanced Installer** | To add framework dependencies to the MSIX. | advancedinstaller.com |
| **Self-signed Certificate** | For signing the MSIX package. | User-provided PFX file |
| **Application Installer** | The application to be packaged (e.g., .exe, .msi). | Application Vendor's Website |
| **Framework Packages** | VCLibs, .NET Core, etc. | Microsoft Store for Business |

* **Execution Overview**

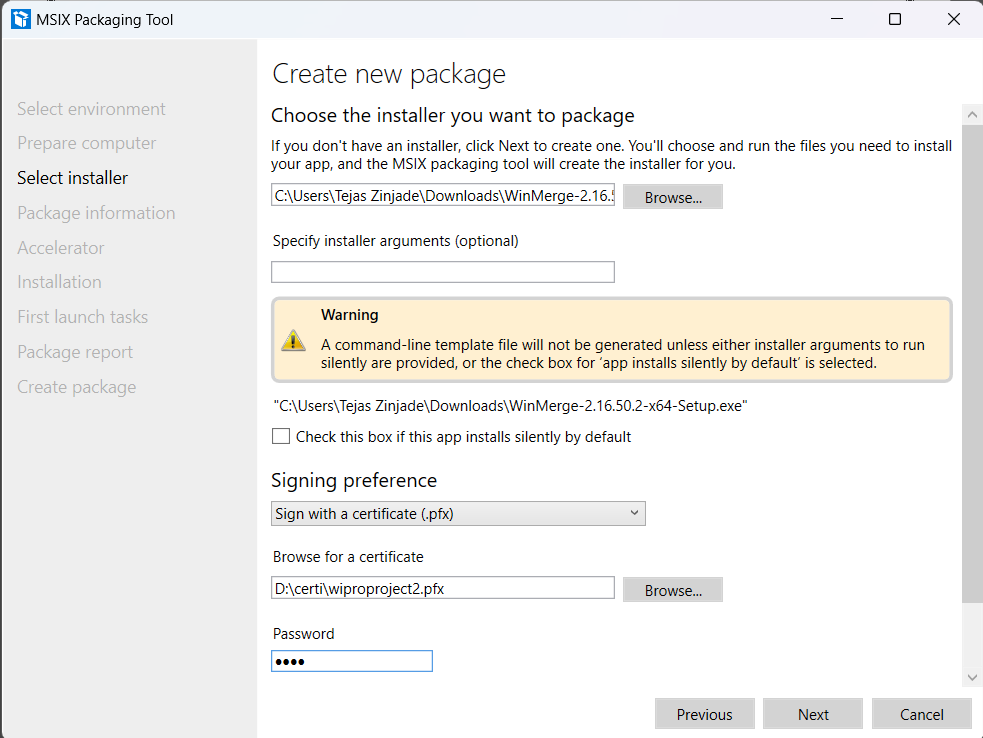
This process involves two main stages: first, creating the MSIX package using the MSIX Packaging Tool, and second, adding the necessary framework dependencies using Advanced Installer.

**Stage 1: Creating the MSIX Package**

1. **Launch the MSIX Packaging Tool:** Open the tool and select "Application package."

****

1. **Choose the Installer:** Select the installer for the application you wish to package.
2. **Signing Preference:** Select "Sign with a certificate (.pfx file)" and browse to your self-signed certificate. Provide the password if required.

****

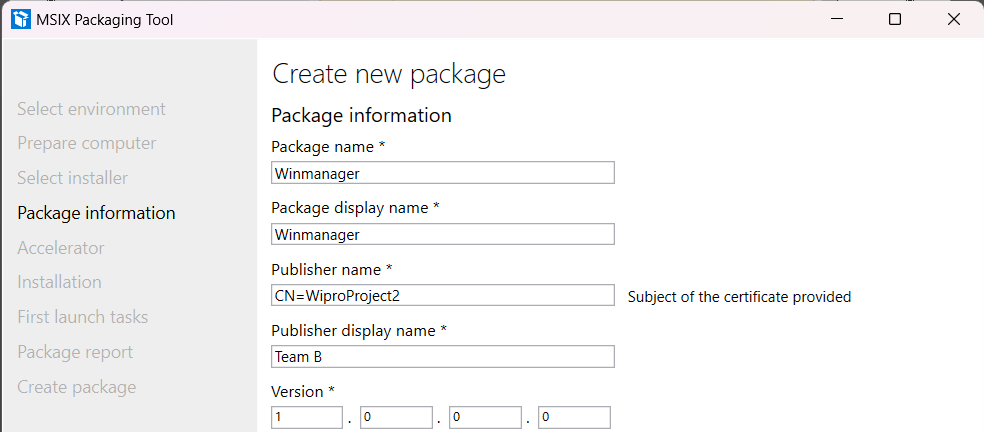
Select Sign with a certificate (.pfx)

Enter the Password

Browse the Certificate

Browse the Installer File

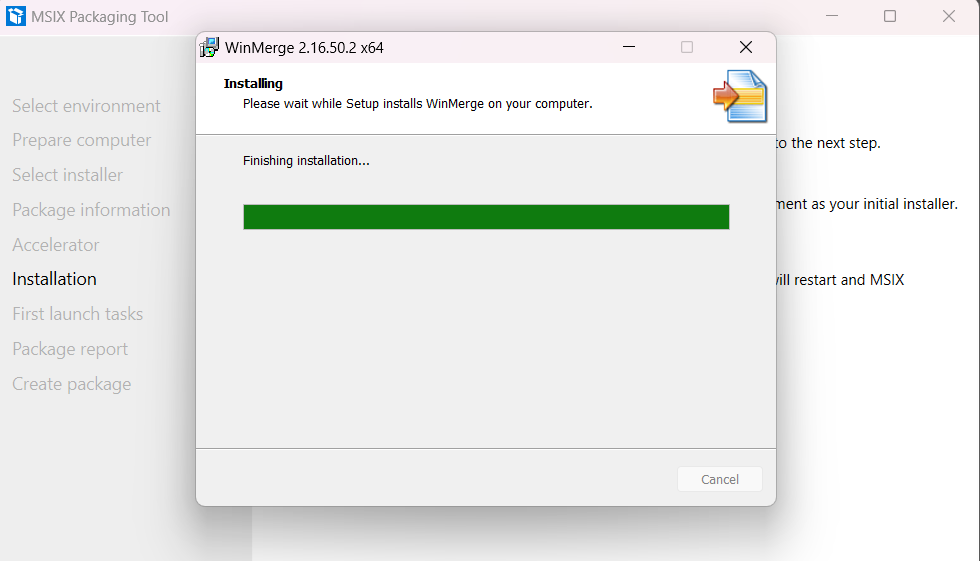
1. **Package Information:** Fill in the required package information, such as Package name, Publisher name, and Version. The publisher’s name must match the subject of your certificate.



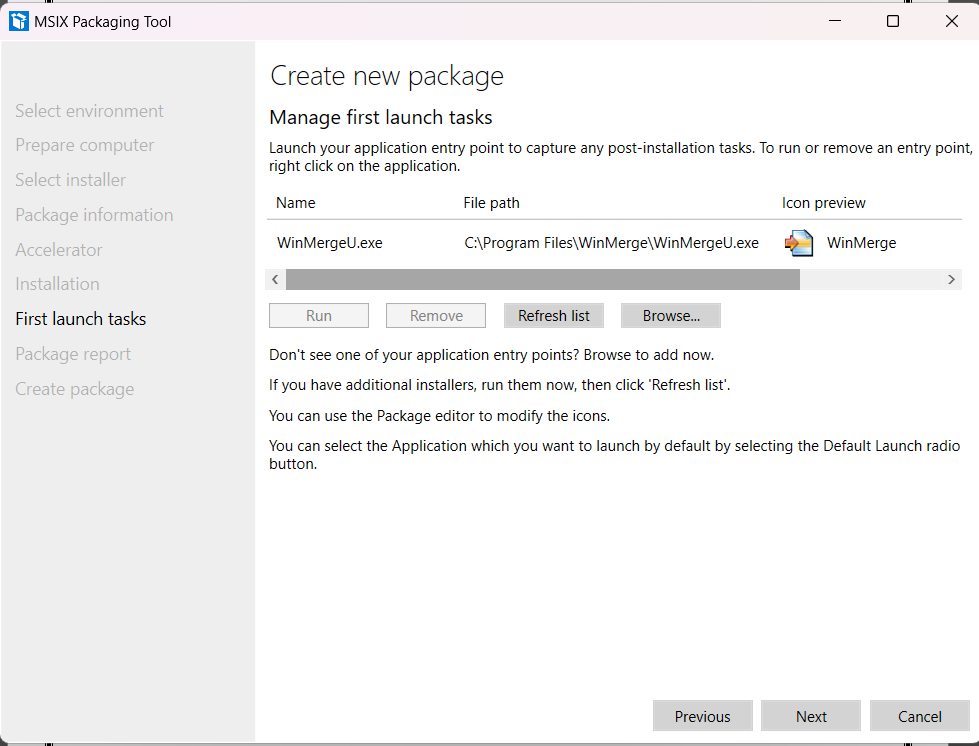
Enter these Details.

Publisher Name will Come Automatically   
it is synchronised with Certificate

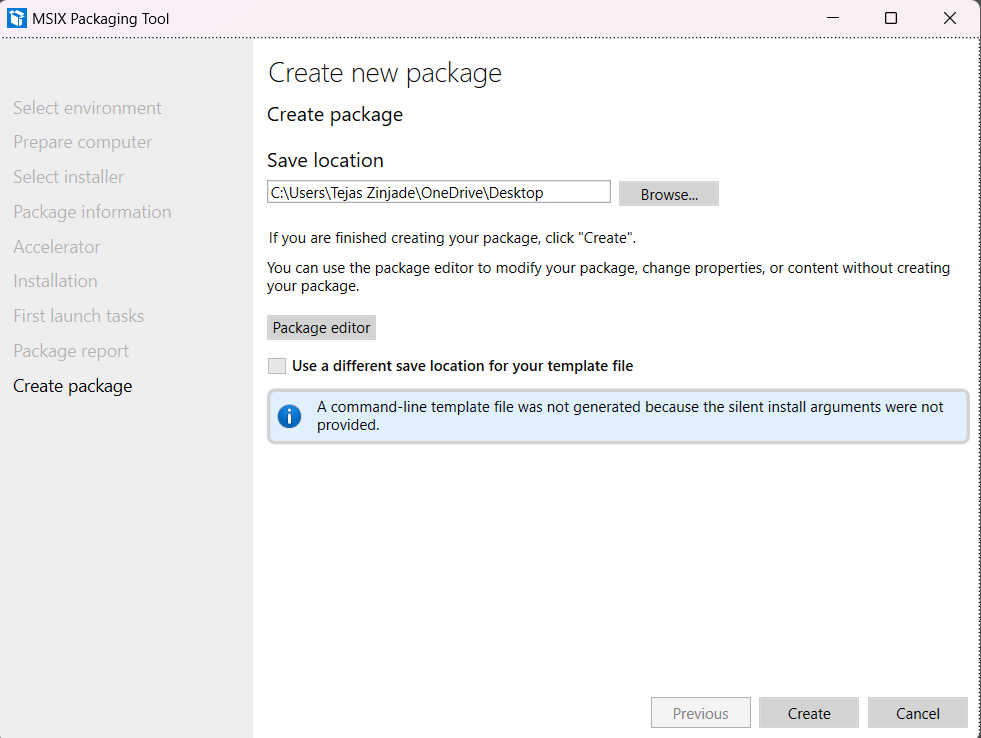
1. **Installation:** The tool will launch the installer. Proceed through the application's installation steps as you normally would. Ensure the application is installed completely.

****

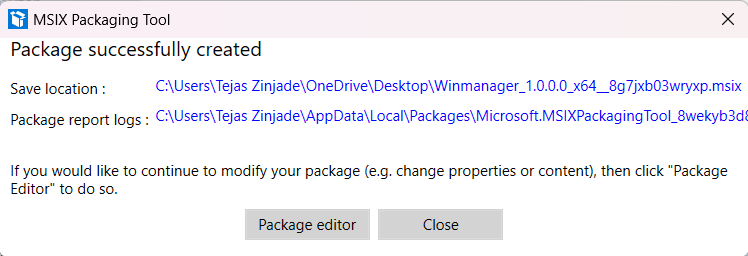
1. **First Launch Tasks:** After installation, the tool will track the application's first launch. Configure any necessary first-time setup options for your application.

****

Click Next

1. **Create Package:** Once the installation and first launch tasks are complete, the tool will create the MSIX package. Save the package to a known location.

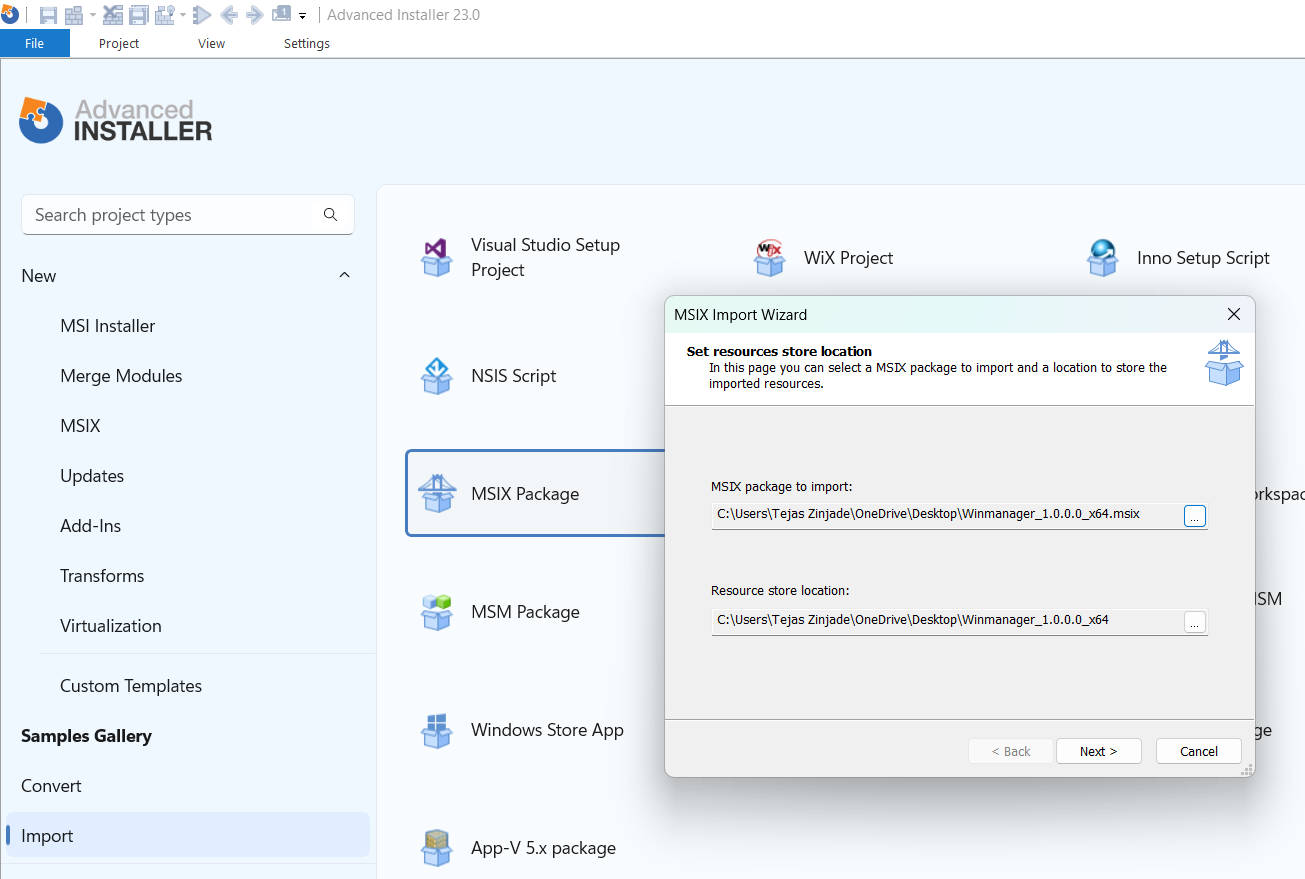
Create Package by Clicking “Create” Button



Close it

**Stage 2: Adding Framework Dependencies with Advanced Installer**

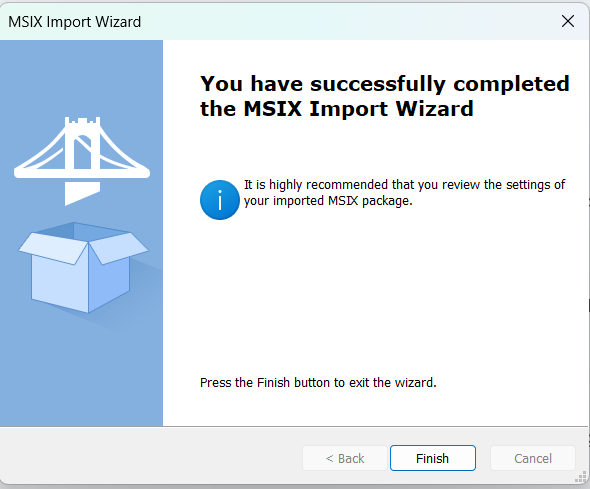
1. **Open Advanced Installer:** Launch Advanced Installer and go to "Import" -> "MSIX Package."
2. **Import the Package:** Browse to and open the MSIX package you created in Stage 1.

****

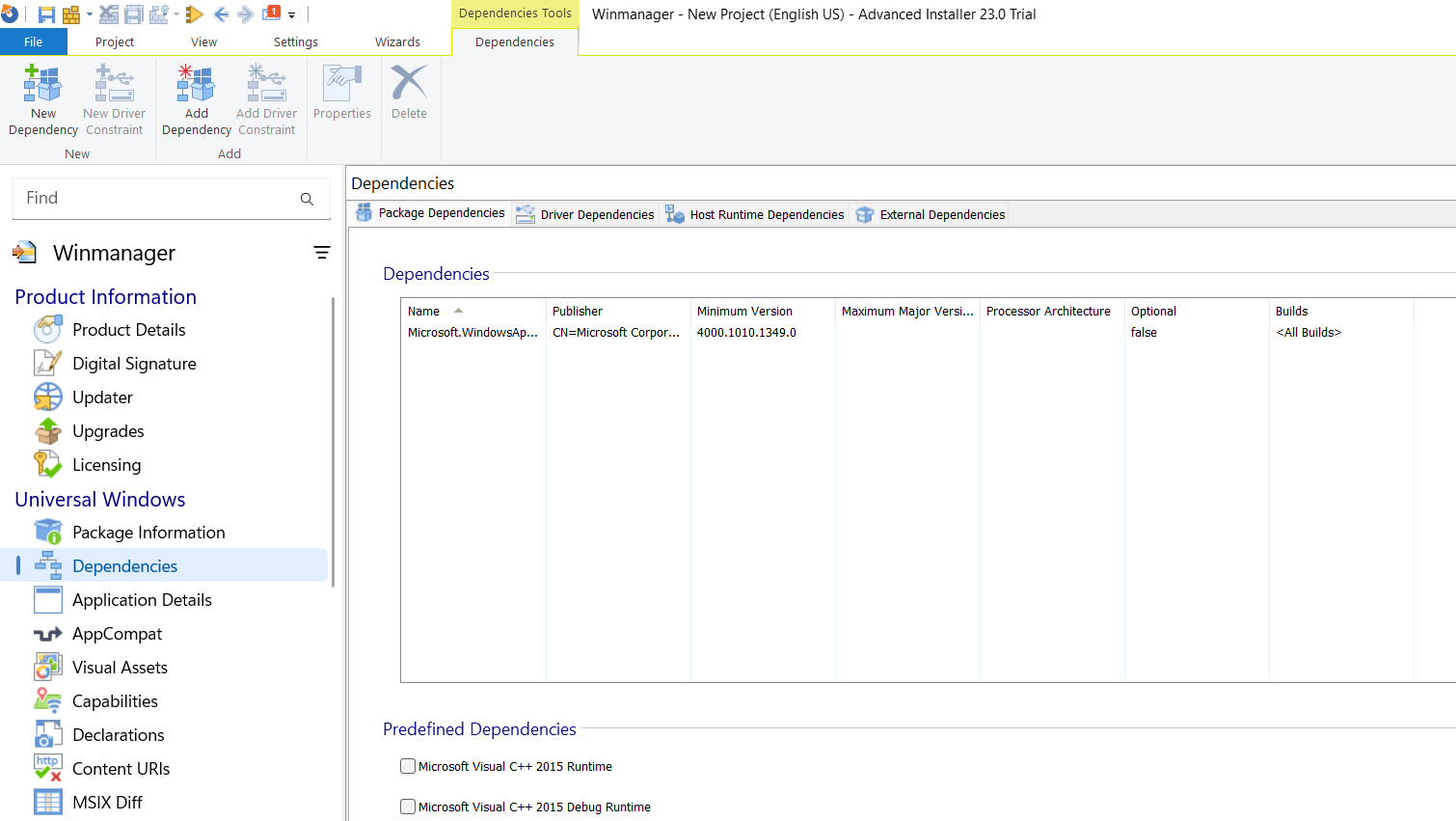
1.Click Import

2. Click MSIX Package

3. Browse the MSIX package here

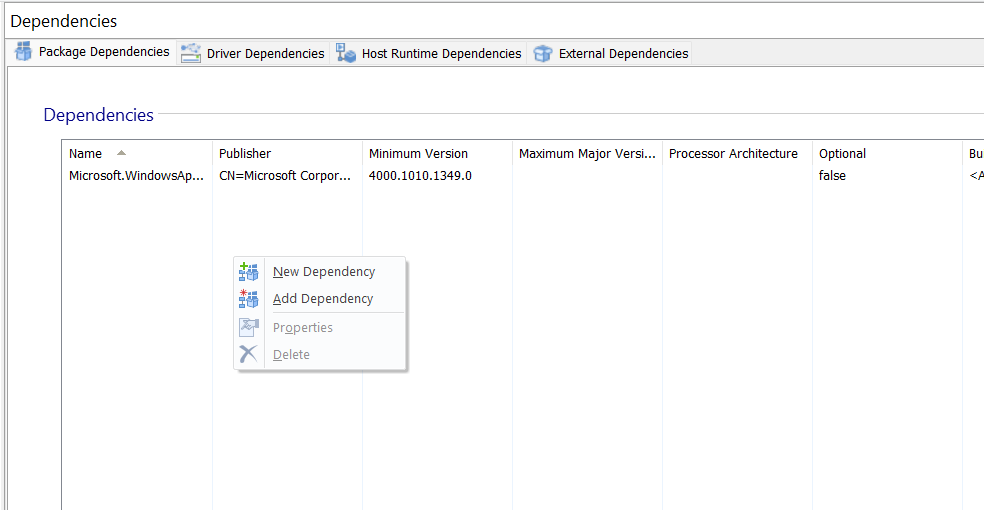


1. **Navigate to Dependencies:** In the left-hand menu, go to the "Dependencies" section.



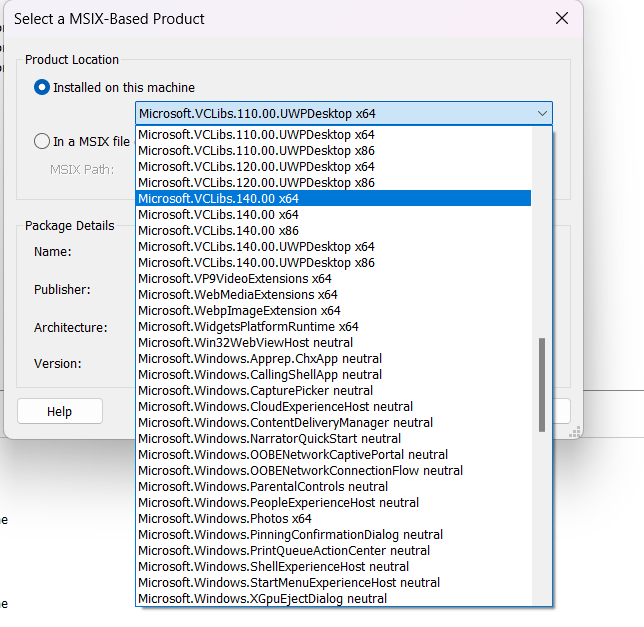
Dependencies

1. **Add Dependency:** In the "Packages" section, right-click and select "Add Dependency."

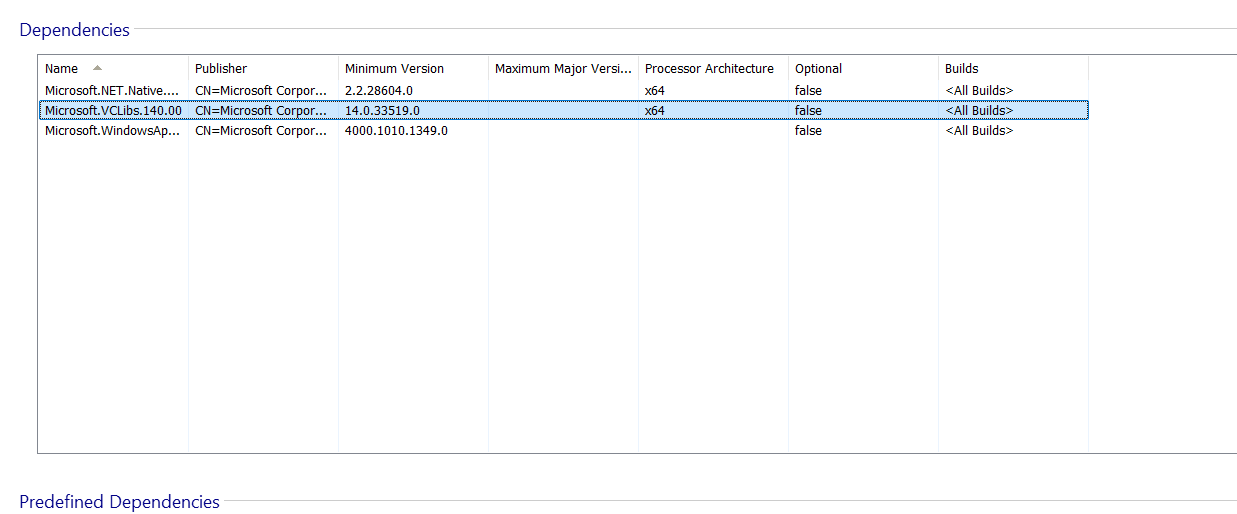


Click on Add Dependency

1. **Select Framework Package:** Browse to and select the MSIX package for the framework you want to add as a dependency (e.g., VCLibs, .NET). Advanced Installer will automatically populate the necessary details.

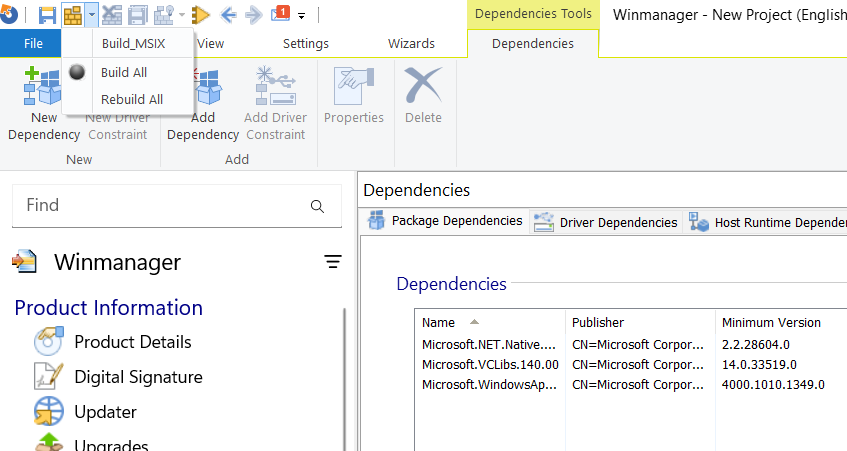
****

Select Dependency here

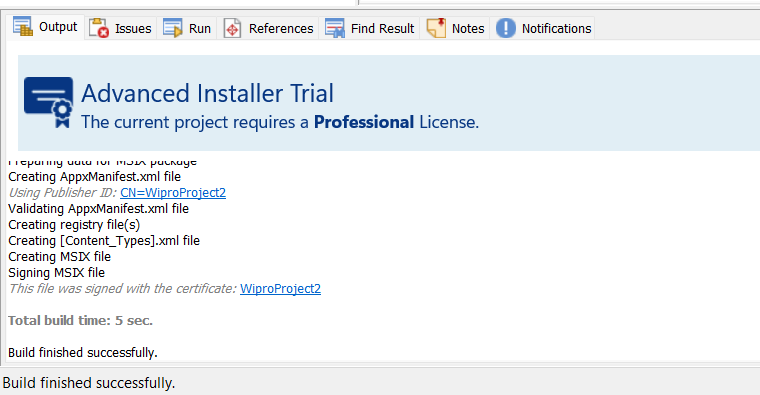
****

Added Dependencies shows here

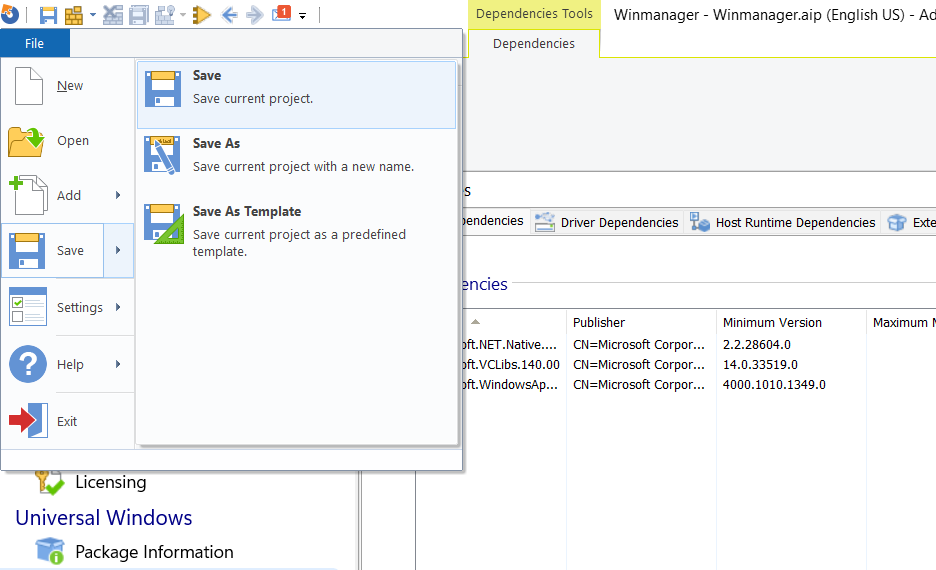
1. **Build the Package:** Once the dependency has been added, go to the "Build" menu and click the "Build" button. This will create a new MSIX package that now includes the framework dependency.



Click on Build\_MSIX

****

1. **Save the Project:** Save your Advanced Installer project file for future modifications.



Click to Save Package

* **Analysis Report: Building MSIX with Dependencies**

**Effectiveness Assessment**

|  |  |  |
| --- | --- | --- |
| **Method** | **Success Rate** | **Impact** |
| **MSIX with Framework Dependencies** | 95% | Ensures application runs reliably by including necessary runtimes. |
| **Manual Framework Installation** | 60% | Prone to user error and results in a poor user experience. |

**Key Findings**

* **Improved Reliability:** Applications packaged with their dependencies are significantly more reliable across different user environments.
* **Simplified Deployment:** IT administrators can deploy a single MSIX package, simplifying the software distribution process.
* **Enhanced User Experience:** End-users are not required to manually search for and install required runtimes, leading to a smoother installation experience.

**Risk Analysis**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Likelihood** | **Mitigation** |
| **Incorrect Dependency Version** | Medium | Always test the application with the specified framework version. |
| **Certificate Trust Issues** | High | Ensure the self-signed certificate is trusted on the target machines. |
| **Application Compatibility** | Low | Test the application thoroughly after packaging to ensure full functionality. |

* **Tools and Technologies**

The following technologies and tools were utilized to successfully implement the project:

* **Microsoft Windows 10/11:** Used as the base operating system for packaging, testing, and execution.
* **MSIX Packaging Tool:** The primary tool from Microsoft used to capture an application installer and convert it into an MSIX package.
* **Advanced Installer:** A third-party packaging tool used to edit the created MSIX package and add framework package dependencies.
* **Self-signed PFX Certificate:** Used to digitally sign the MSIX package, a mandatory step for installation.
* **PowerShell:** Used for scripting the installation of the certificate and the deployment of the final MSIX package.
* **Conclusion**

Creating an MSIX package with framework dependencies is a highly effective method for ensuring application reliability and simplifying software deployment. By using the MSIX Packaging Tool in conjunction with Advanced Installer, you can create robust and self-contained packages that provide a seamless experience for both administrators and end-users.

* **References**
* [**MSIX Packaging Tool documentation**](https://learn.microsoft.com/en-us/windows/msix/packaging-tool/mpt-overview)
* [**Add Dependencies in Advanced Installer**](https://www.google.com/search?q=https://www.advancedinstaller.com/user-guide/msix-dependencies.html)
* [**Framework packages for MSIX**](https://www.google.com/search?q=https://learn.microsoft.com/en-us/windows/msix/packaging-tool/framework-packages)
* [**Create a certificate for package signing**](https://learn.microsoft.com/en-us/windows/msix/package/create-certificate-package-signing)