

Create Package in MSI packaging tool

The **MSIX Packaging Tool** is used to convert traditional desktop applications into the **MSIX** format, not MSI.

To create a package using the MSIX Packaging Tool, follow these steps. First of all we can need to **install a MSI Packaging Tool** and also **have your installer file ready** (e.g., .exe, .msi)

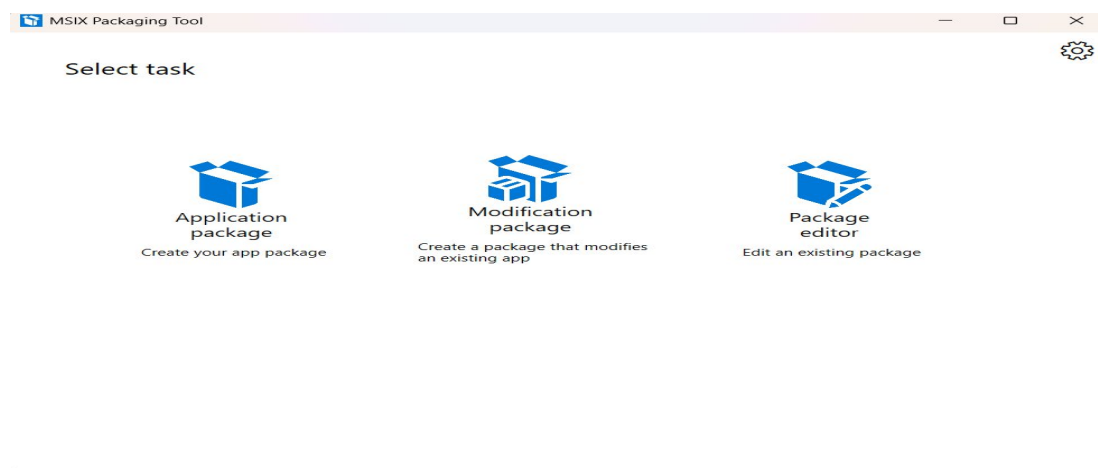
Steps to Create an MSIX Package

Step 1: Launch MSIX Packaging Tool

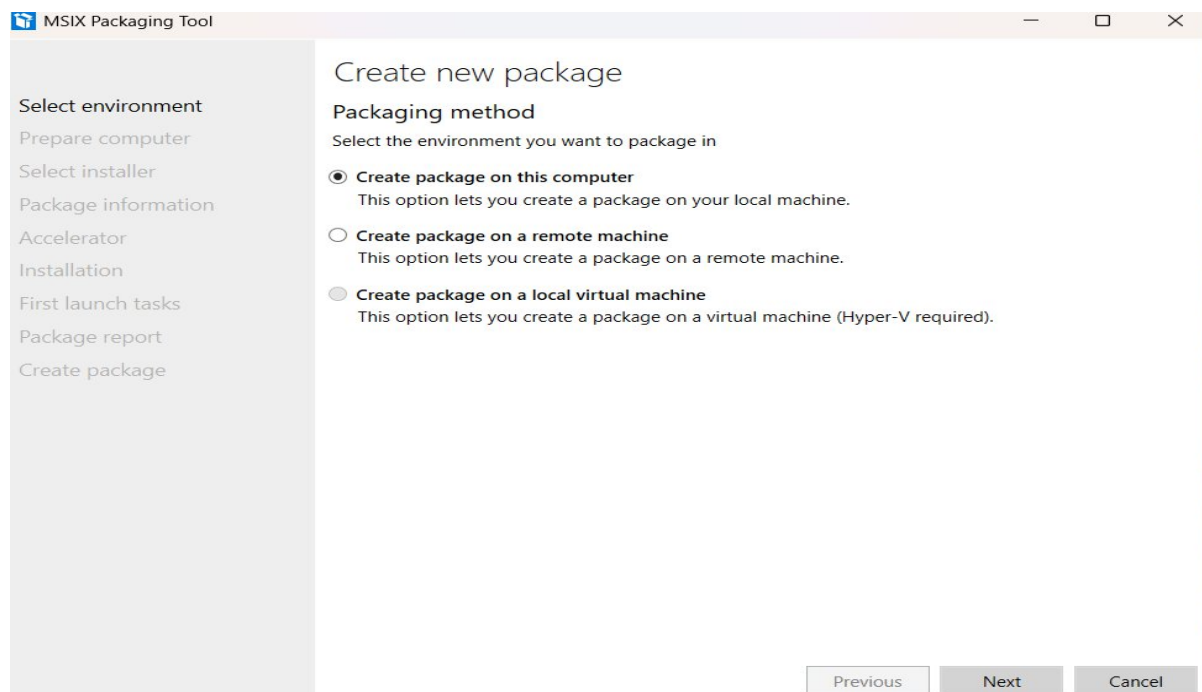
Open the tool from the Start Menu: **MSIX Packaging Tool**

Step 2: Start a New Package

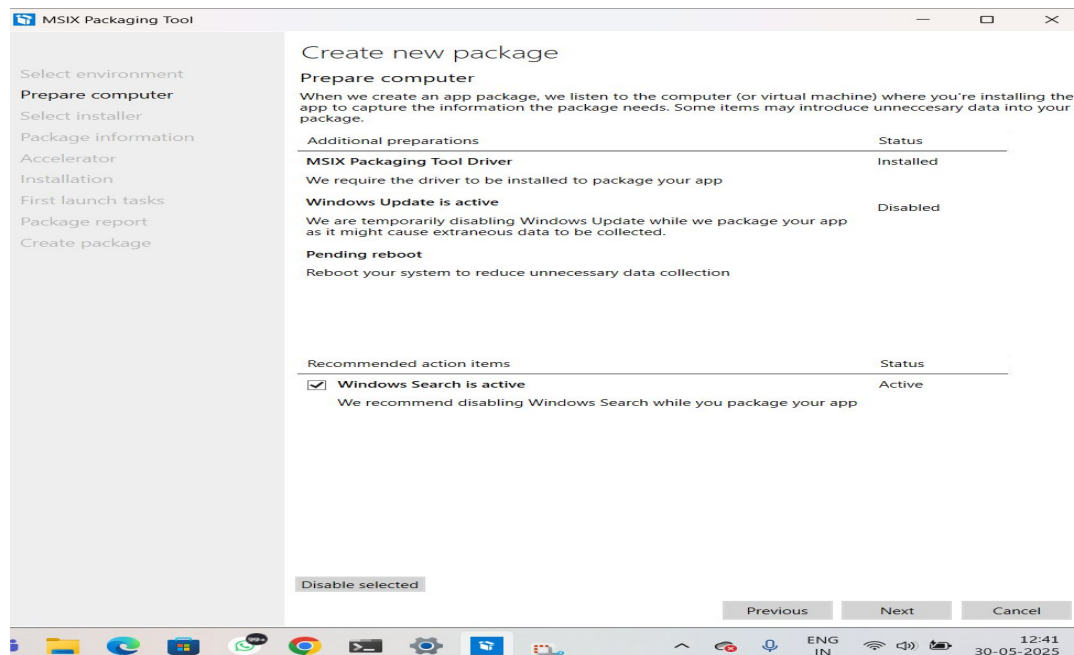
Click on “Application Package”



Step 3: Choose "Create package on this computer" and click Next.

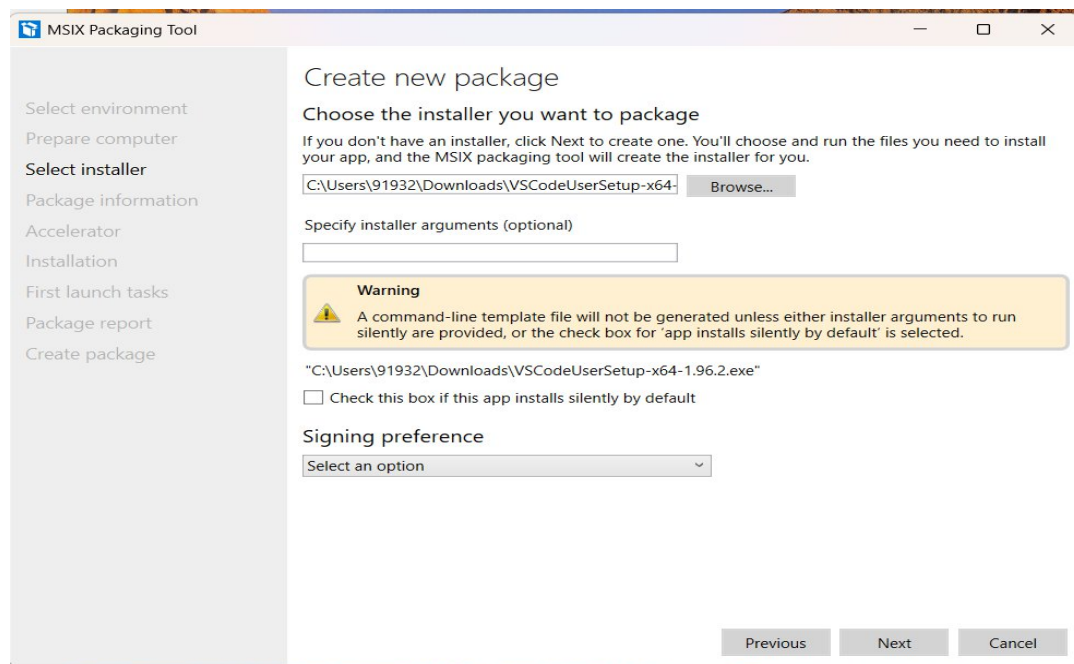


Step 4: Install the MSIX Packaging Tool Driver



Step 5: Enter the path to the installer (e.g., C:\Installers\MyApp.exe).

And it's Optional to specify any installer arguments



Step 6: Provide the following:

- Package Name
 - Package Display Name
 - Publisher Name
 - Publisher Display Name
 - Version
 - Installation Location
- Click “Next”.

The screenshot shows the 'MSIX Packaging Tool' window with the 'Create new package' dialog. The 'Package information' tab is active. The left sidebar lists steps: Select environment, Prepare computer, Select installer, Package information (selected), Accelerator, Installation, First launch tasks, Package report, and Create package. The main area contains the following fields and options:

- Package name ***: VSCode
- Package display name ***: Visual Studio code
- Publisher name ***: CN=Visual Studio Code
- Publisher display name ***: VSCode
- Version ***: 1 . 0 . 0 . 0
- Package Description**: My Description
- Installation location**: C:\Users\91932\Downloads (with a 'Browse...' button)
- ☒ Add support for MSIX Core to this package. See <https://aka.ms/MSIXCore> to learn more.
- Windows Version for MSIX Core support**: Windows 7, SP 1 (6.1.7601.0)

Below these fields is a warning: "Once you continue to the next page, installation will begin and you will not be able to return." At the bottom right are three buttons: Previous, Next, and Cancel.

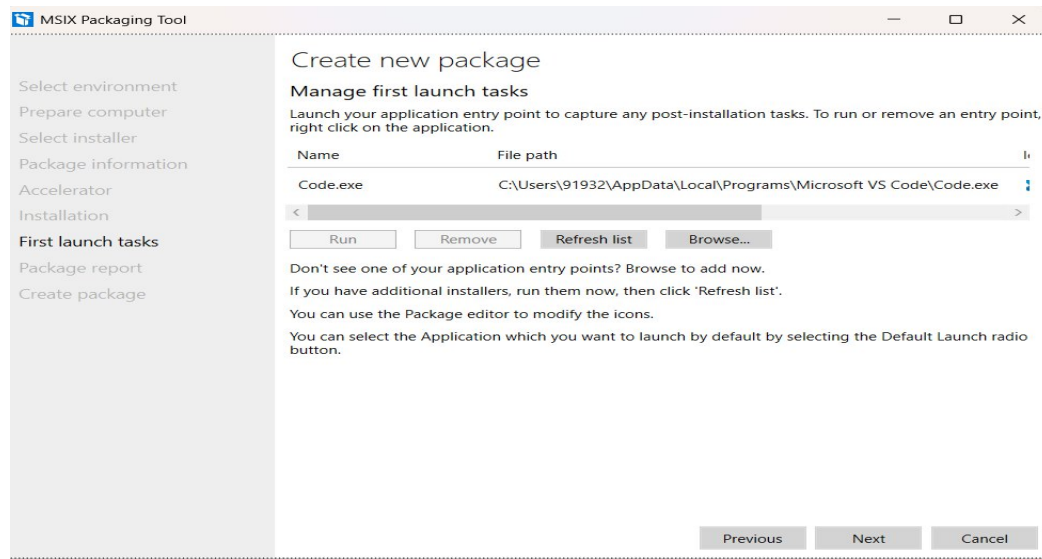
Step 7: This is the core step where the tool transforms a traditional app installer into a modern package

The screenshot shows the 'MSIX Packaging Tool' window with the 'Create new package' dialog. The 'Installation' tab is active. The left sidebar is the same as in Step 6, with 'Installation' now selected. The main area contains the following text and a button:

- Installation**: Your installer has finished running. If you have additional installers or executables, run them in the same environment as your initial installer. Do this before moving to the next step.
- Restart machine**: If your installer requires a system restart, click 'Restart machine'. The system will restart and MSIX Packaging Tool will automatically resume after restart.

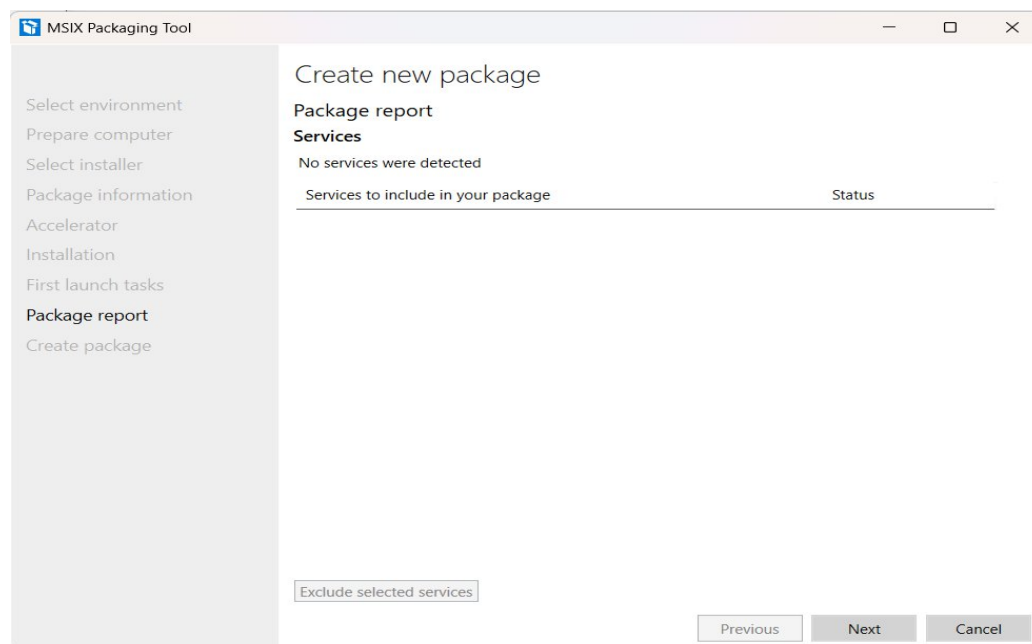
Below this text is a button labeled 'Restart machine'. At the bottom right are three buttons: Previous, Next, and Cancel.

Step 8: The **"Manage First Launch Tasks"** step in the **MSIX Packaging Tool** is an optional step that allows you to automate tasks that must run the first time the application is launched after it's packaged.



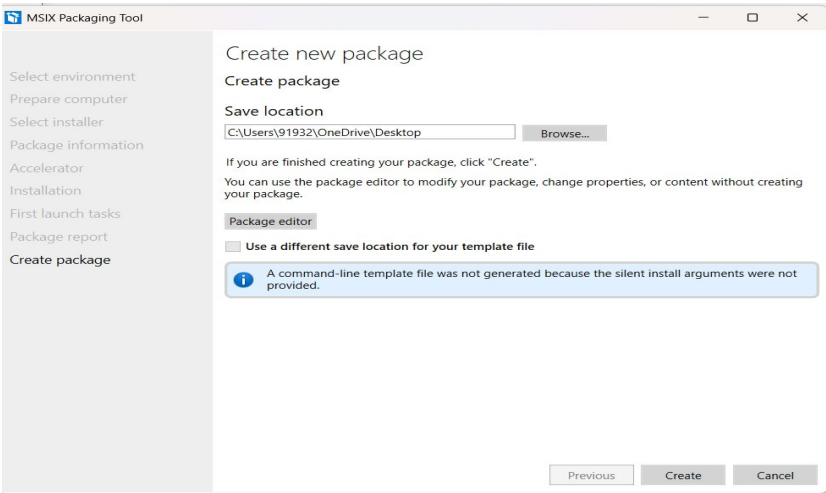
Step 9: It provides a **summary of the packaging process**, highlighting issues, warnings, and details about what was captured.

The Package Report helps **ensure package quality** and avoids publishing an incomplete or broken MSIX file



Step 10: Create The Package

Click **Create** to build the .msix Package



You will get an .msix file



Step 12:

