Create Package in MSIX 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 a of all we can need to **install a MSI Packaging Tool** and also **have your installer file ready** (e.g., .exe, .msi)

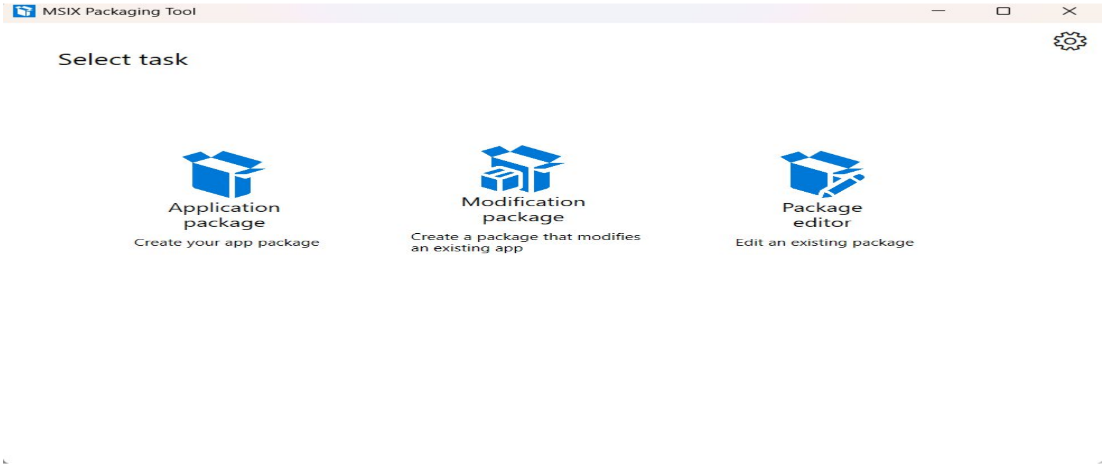
Steps to Create an MSIX Package

Step 1: Launch MSIX Packaging Tool

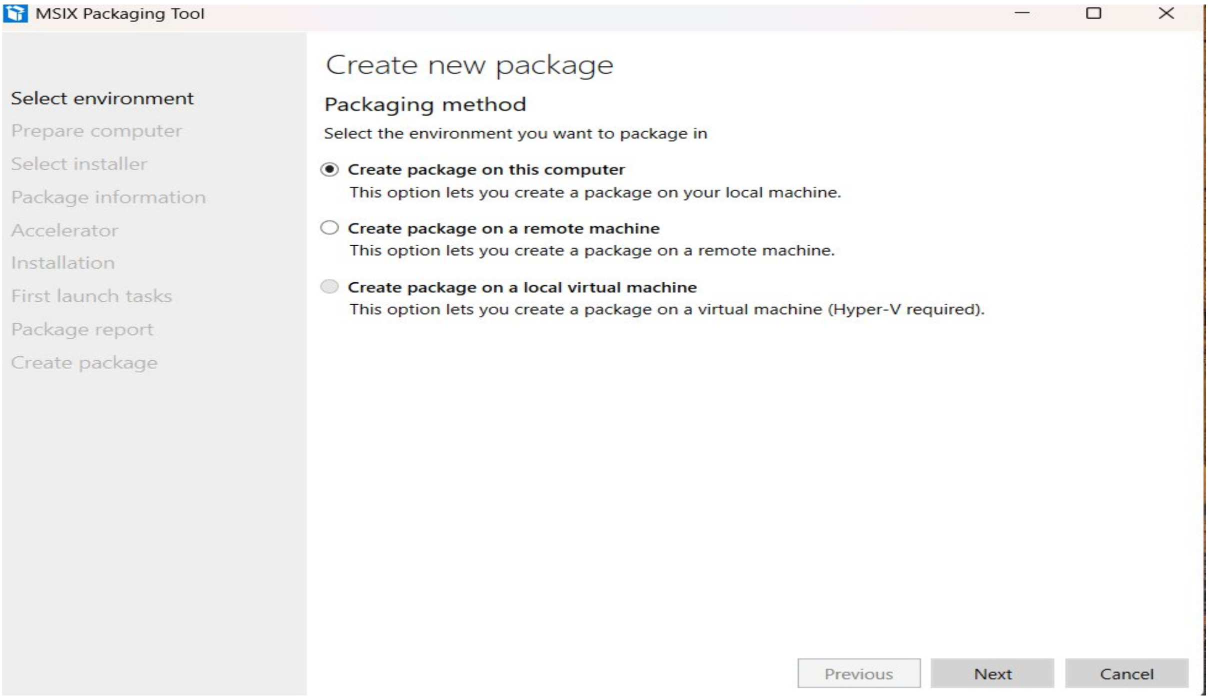
Open the tool from the Start Menu: **MSIX Packaging Tool**

Step 2: Start a New Package

Click on “Application Package”



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



Create new package

Packaging method

Select the environment you want to package in

● Create package on this computer

This option lets you create a package on your local machine.

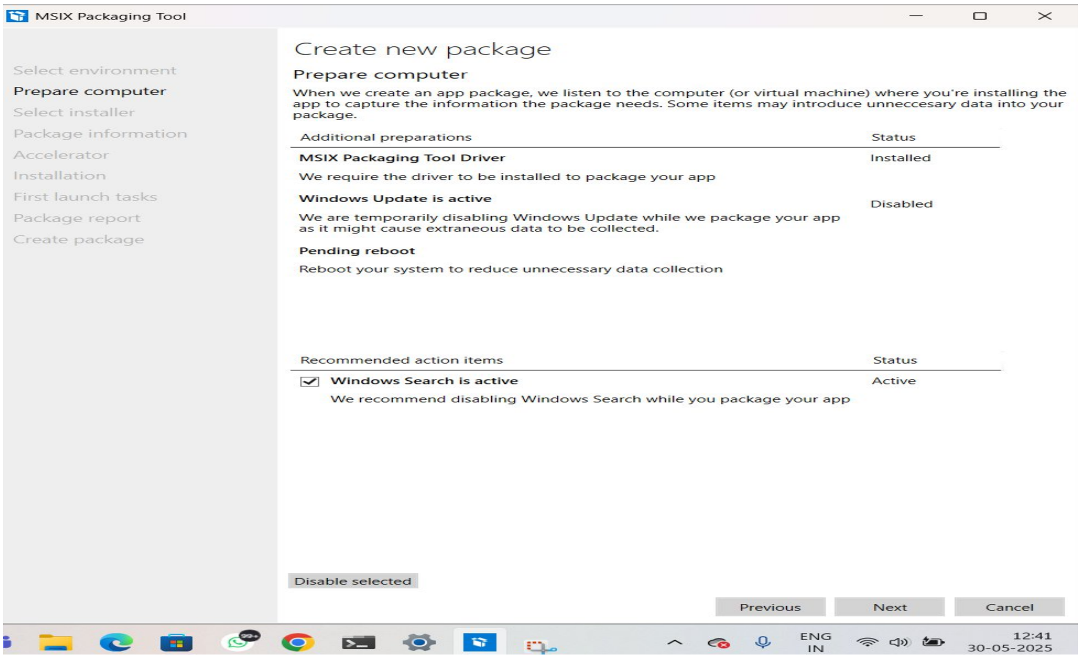
○ Create package on a remote machine

This option lets you create a package on a remote machine.

○ Create package on a local virtual machine

This option lets you create a package on a virtual machine (Hyper-V required).

Step 4: Install the MSIX Packaging Tool Driver



Create new package

Prepare computer

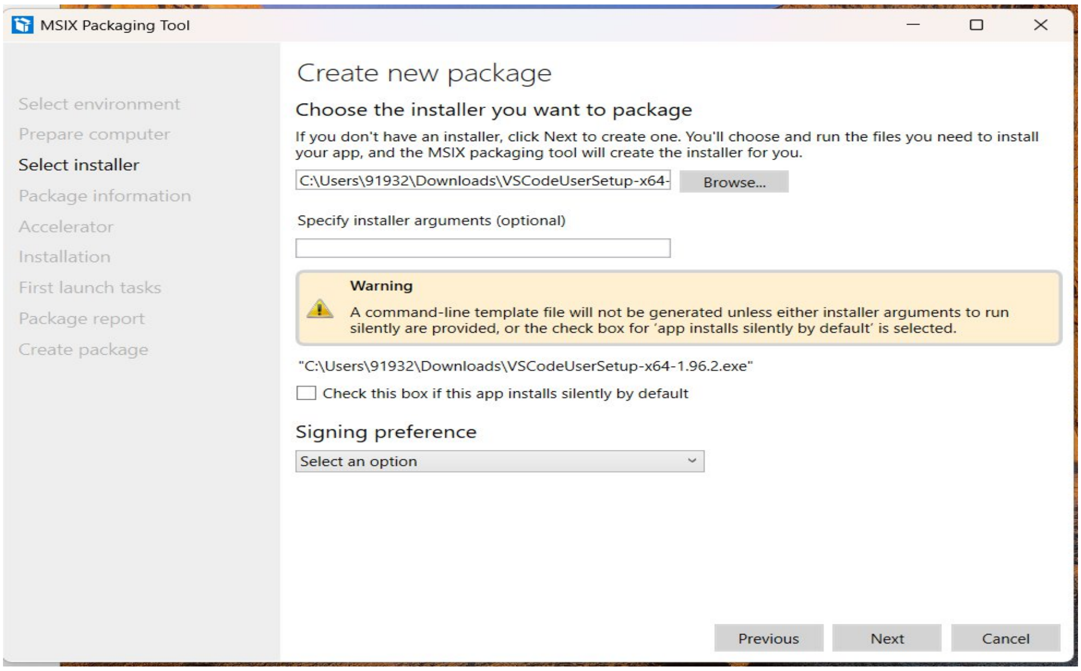
When we create an app package, we listen to the computer (or virtual machine) where you're installing the app to capture the information the package needs. Some items may introduce unneccesary data into your package.

Additional preparations	Status
MSIX Packaging Tool Driver We require the driver to be installed to package your app	Installed
Windows Update is active We are temporarily disabling Windows Update while we package your app as it might cause extraneous data to be collected.	Disabled
Pending reboot Reboot your system to reduce unnecessary data collection	

Recommended action items	Status
☑ **Windows Search is active** We recommend disabling Windows Search while you package your app	Active

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

And it’s Optional to specify any installer arguments



Create new package

Choose the installer you want to package

If you don't have an installer, click Next to create one. You'll choose and run the files you need to install your app, and the MSIX packaging tool will create the installer for you.

C:\Users\91932\Downloads\VSCodeUserSetup-x64-

Specify installer arguments (optional)

Warning

A command-line template file will not be generated unless either installer arguments to run silently are provided, or the check box for ‘app installs silently by default’ is selected.

"C:\Users\91932\Downloads\VSCodeUserSetup-x64-1.96.2.exe"

□ Check this box if this app installs silently by default

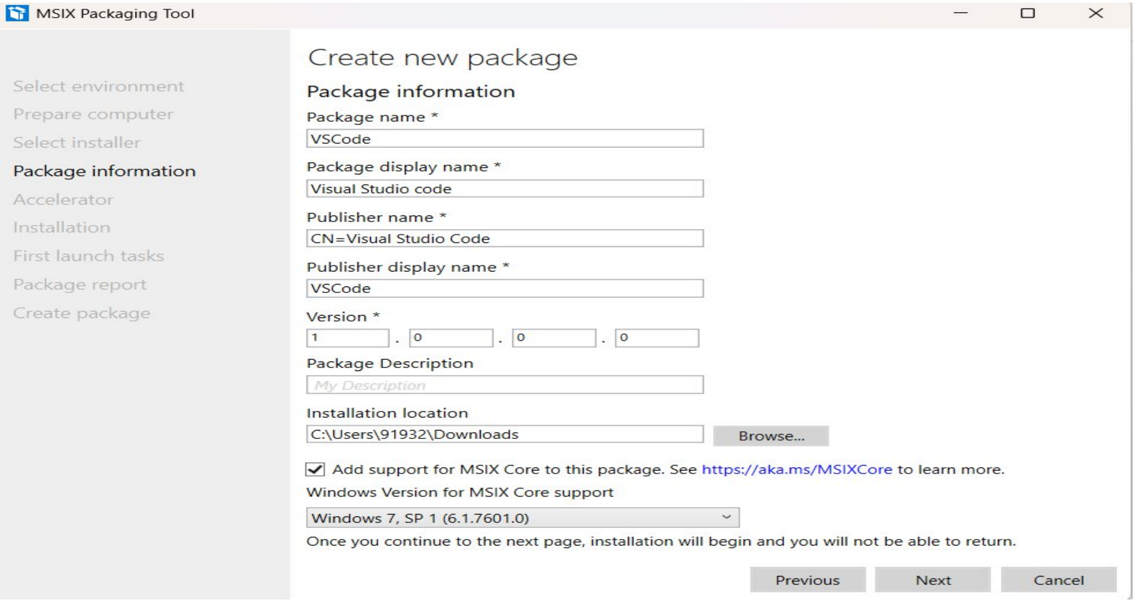
Signing preference

Select an option

Step 6:Provide the following:

- Package Name
- Package Display Name
- Publisher Name
- Publisher Display Name
- Version
- Installation Location

Click “ Next”.



MSIX Packaging Tool

Create new package

Package information

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

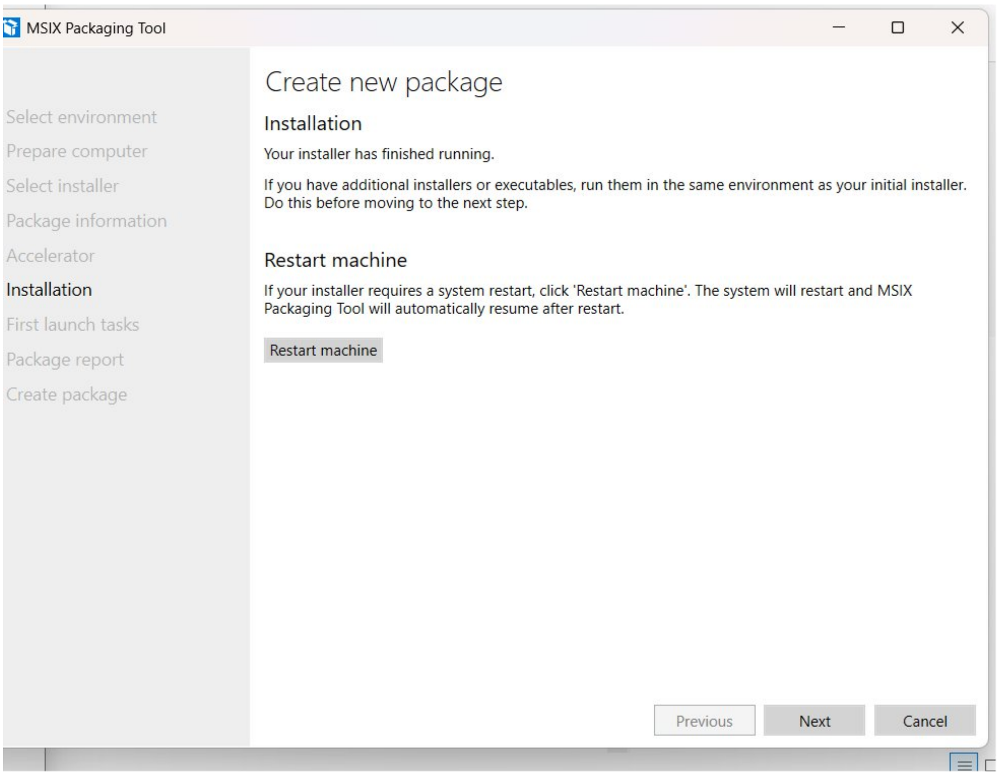
Installation location
C:\Users\91932\Downloads Browse...

☑ 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)

Once you continue to the next page, installation will begin and you will not be able to return.

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



MSIX Packaging Tool

Create new package

Installation

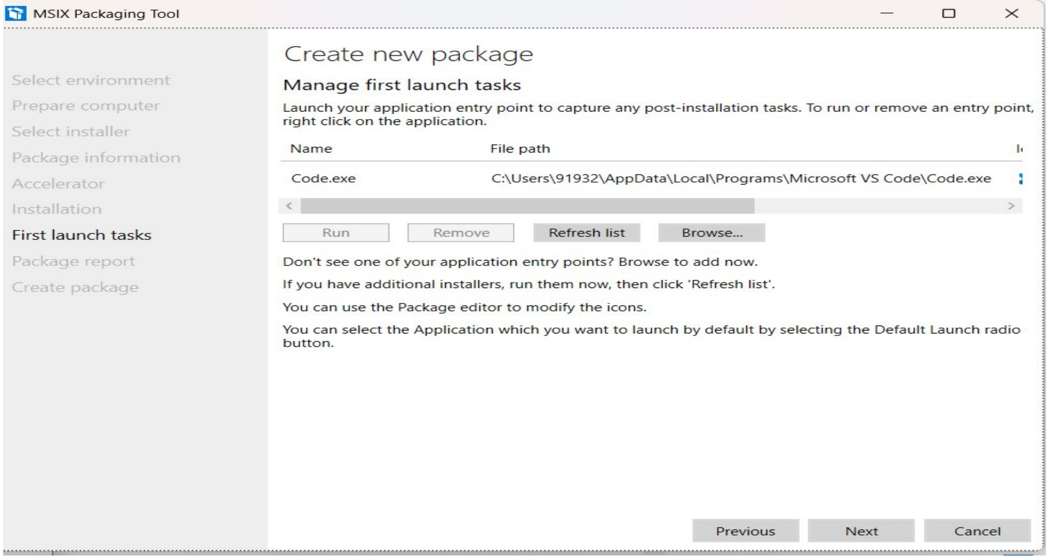
Your installer has finished running.

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.

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.



Create new package

Manage first launch tasks

Launch your application entry point to capture any post-installation tasks. To run or remove an entry point, right click on the application.

Name	File path
Code.exe	C:\Users\91932\AppData\Local\Programs\Microsoft VS Code\Code.exe

Don't see one of your application entry points? Browse to add now.

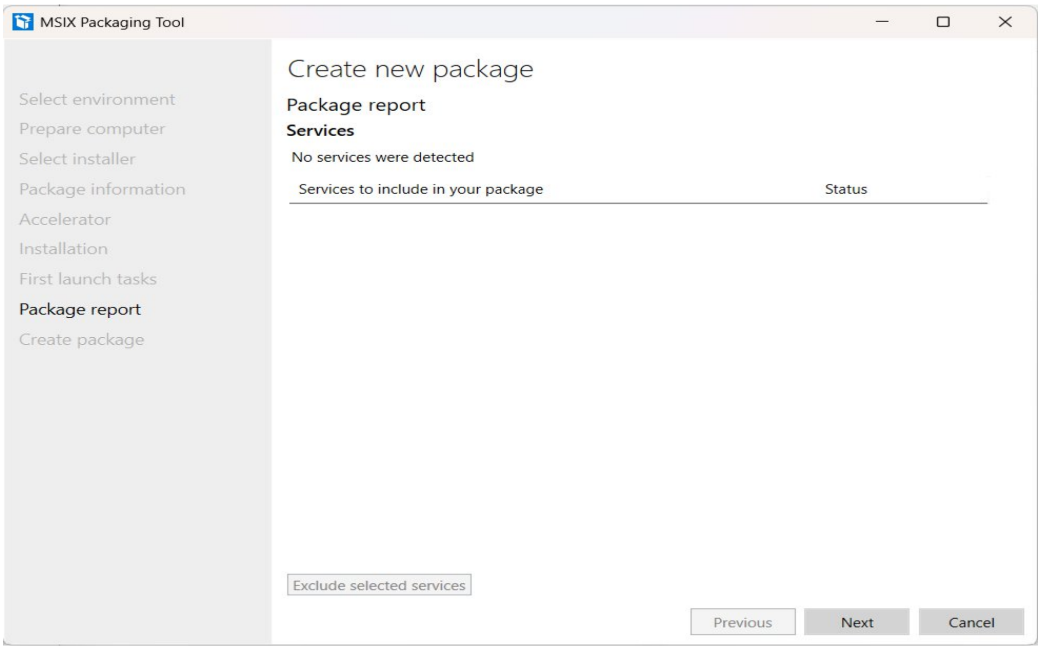
If you have additional installers, run them now, then click 'Refresh list'.

You can use the Package editor to modify the icons.

You can select the Application which you want to launch by default by selecting the Default Launch radio button.

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



Create new package

Package report

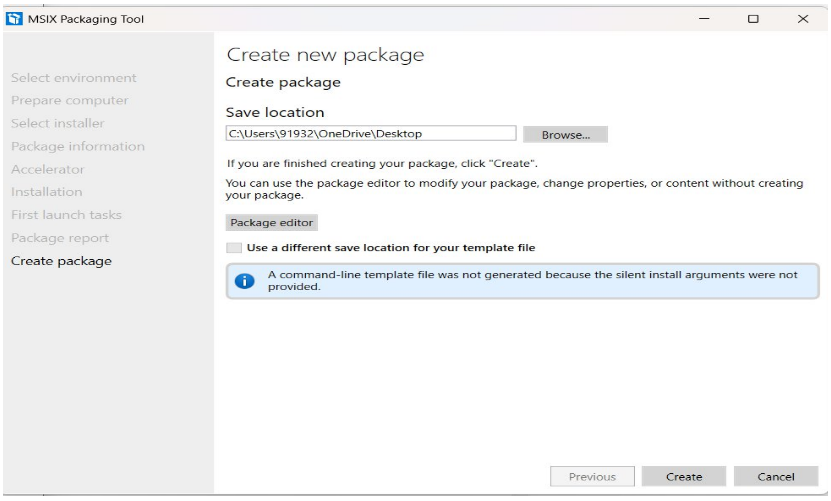
Services

No services were detected

Services to include in your package	Status

Step 10: Create The Package

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



Create new package

Create package

Save location

C:\Users\91932\OneDrive\Desktop

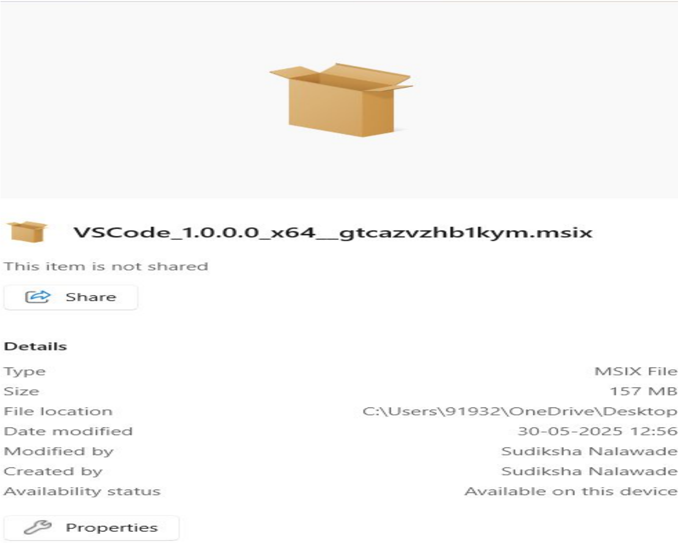
If you are finished creating your package, click "Create".

You can use the package editor to modify your package, change properties, or content without creating your package.

□ Use a different save location for your template file

A command-line template file was not generated because the silent install arguments were not provided.

You will get an **.msix** file



VSCode_1.0.0.0_x64__gtcazvzhb1kym.msix

This item is not shared

Details

Type	MSIX File
Size	157 MB
File location	C:\Users\91932\OneDrive\Desktop
Date modified	30-05-2025 12:56
Modified by	Sudiksha Nalawade
Created by	Sudiksha Nalawade
Availability status	Available on this device

Step 12: