GETTING STARTED

PACE Suite 5.1

Document version: 20190205

Table of Contents

1 Introduction 7
1.1 About This Document 7
1.2 Glossary 7
2 PACE Overview 8
2.1 Installation & Activation 9
2.1.1 Software Prerequisites 9
2.1.2 Hardware Prerequisites 9
2.1.3 Installing PACE Suite 10
2.1.4 Activating PACE Suite 14
3 How-To Manuals 16
3.1 Create MSI 16
3.1.1 Create MSI from Scratch 16
3.1.2 Repackage Installation to MSI 19
3.1.3 Capture System Changes to MSI 35
3.2 Create MST 48
3.2.1 Create Blank MST 48
3.2.2 Create Response MST 51
3.2.3 Capture System Changes to MST 57
3.3 Create MSP 72
3.3.1 Generate MSP on Difference 72
3.3.2 Save Changes as MSP 78
3.4 Create MSIX/APPX 85
3.4.1 Create MSIX/APPX from Scratch 85
3.4.2 Repackage Installation to MSIX/APPX 88
3.5 Create APPV 99
3.5.1 Create APPV from Scratch 99
3.5.2 Repackage Installation to APPV 102
3.6 Create THINAPP 113
3.6.1 Create THINAPP from Scratch 113
3.6.2 Repackage Installation to THINAPP 116
3.7 Edit MSI/MST 126
3.7.1 Pre-Condition 127
3.7.2 Application Details 129

3.7.3 Package IDs 130
3.7.4 Files and Folders 131
3.7.4.1 Add System Folder 132
3.7.4.2 Add Subfolder 134
3.7.4.3 Import Files and COM 137
3.7.4.4 Change Compression Type 143
3.7.4.5 Set Permissions 146
3.7.4.6 Remove Permissions 150
3.7.4.7 Move Files, Folders 153
3.7.4.8 Set INSTALLDIR 155
3.7.4.9 Control Empty Folder Creation 156
3.7.4.10 Control File/Folder Removal 159
3.7.4.11 Edit INI File 163
3.7.4.12 Edit File Properties 165
3.7.4.13 Edit Folder Properties 167
3.7.5 Shortcuts 169
3.7.5.1 Add Shortcut 169
3.7.5.2 Import Shortcuts 175
3.7.5.3 Move Shortcut 177
3.7.5.4 Edit Shortcut Properties 179
3.7.6 Registry 182
3.7.6.1 Add Registry Key 183
3.7.6.2 Add Registry Value 185
3.7.6.2.1 Add ActiveSetup 187
3.7.6.3 Add Value/Key Removal 192
3.7.6.4 Import Registry 194
3.7.6.5 Export Registry 198
3.7.6.6 Set Permissions 199
3.7.6.7 Remove Permissions 203
3.7.6.8 Move Value, Key 206
3.7.6.9 Edit Registry Key 208
3.7.6.10 Edit Registry Value 210
3.7.6.11 Edit Value/Key Removal 213
3.7.7 Custom Actions 217

3.7.7.1 Add Standard Actions 217
3.7.7.2 Add Custom Action 220
3.7.7.2.1 Run PowerShell script 225
3.7.7.3 Edit Custom Action 234
3.7.8 Drivers 239
3.7.8.1 Add Driver 239
3.7.8.2 Edit Driver 244
3.7.9 Features and Components 246
3.7.9.1 Add Feature 246
3.7.9.2 Add Component 249
3.7.9.3 Move Feature, Component 251
3.7.9.4 Edit Feature Properties 253
3.7.9.5 Edit Component Properties 255
3.7.9.6 Detect Empty Components 259
3.7.9.7 Review Component Resources 261
3.7.10 ARP Appearance 264
3.7.11 Upgrades 267
3.7.11.1 Import Upgrades 267
3.7.11.2 Add Upgrade 269
3.7.11.3 Edit Upgrade 272
3.7.12 Properties 275
3.7.12.1 Add Property 275
3.7.12.2 Edit Property 277
3.7.13 Summary Information 279
3.7.14 Tables 282
3.7.14.1 Add Tables 282
3.7.14.2 Drop Table 284
3.7.14.3 Import Tables (IDT) 286
3.7.15 Merge Modules 289
3.7.15.1 Create MSM 289
3.7.15.2 Import MSM to Package 292
3.8 Edit Project 296
3.8.1 Pre-Condition 296
3.8.2 File and Folders 299

3.8.2.1 Exclude File/Folder 299
3.8.2.2 Include Excluded File/Folder 300
3.8.2.3 Add Folder 301
3.8.2.4 Import Files 303
3.8.3 Registry 305
3.8.3.1 Exclude Registry 305
3.8.3.2 Include Excluded Registry 306
3.8.3.3 Import Registry 307
3.8.3.4 Export Registry 309
3.8.4 Services 311
3.8.4.1 Exclude Service 311
3.8.4.2 Include Excluded Service 312
3.8.4.3 Edit Service 313
3.8.5 Environment Variables 315
3.8.5.1 Add Environment Variable 315
3.8.5.2 Remove Environment Variable 317
3.8.6 MSI Settings 318
3.8.6.1 Set Application Details 319
3.8.6.2 Set INSTALLDIR 320
3.8.6.3 Manage Shortcuts 321
3.8.6.3.1 Exclude Shortcut 321
3.8.6.3.2 Include Excluded Shortcut 322
3.8.6.3.3 Edit Shortcut 323
3.8.6.3.4 Import Shortcut 324
3.8.6.4 Control File/Folder Permissions 326
3.8.6.5 Control Registry Permissions 327
3.8.6.6 Select Compression Type 328
3.8.6.7 Use Advertising Tables 330
3.8.6.8 Add Active Setup 332
3.8.6.9 Set Summary Information 334
3.8.7 MSIX/APPX Settings 336
3.8.7.1 Create Modification Package 337
3.8.7.2 Set Display Properties 339
3.8.7.3 Set Identity Properties 340

3.8.7.4 Declare Capabilities 341
3.8.7.5 Set Supported Platforms 343
3.8.7.6 Manage Package Architectures 345
3.8.7.7 Create Bundle 346
3.8.7.8 Sign Package/Bundle 348
3.8.7.9 Edit Manifest Manually 350
3.8.7.10 Manage Apps and Visual Assets 351
3.8.7.10.1 Exclude App 352
3.8.7.10.2 Include Excluded App 353
3.8.7.10.3 Add App 354
3.8.7.10.4 Set App Logos 355
3.8.7.11 Generate AppInstaller 357
3.8.8 APPV Settings 359
3.8.8.1 Set Application Details 359
3.8.8.2 Manage Package Settings 360
3.8.8.3 Set Folder Virtualization Level 361
3.8.8.4 Set Registry Virtualization Level 362
3.8.8.5 Manage Apps, FTA, Shortcuts 363
3.8.8.5.1 Exclude App 364
3.8.8.5.2 Include Excluded App 365
3.8.8.5.3 Add App 366
3.8.8.5.4 Add App Shortcut 368
3.8.8.5.5 Add App FTA 369
3.8.9 THINAPP Settings 371
3.8.9.1 Set Entry Points 371
3.8.9.2 Define Primary Data Container 372
3.9 Edit APPV 374
3.10 Deploy MSI to SCCM 377
3.11 Generate MSI Package Report 382
3.12 Validate MSI Package 385
3.13 Get MSI Package Complexity 388
3.14 Create Discovery Documentation 392
4 References 396
4.1 Contact 396
4.2 Social Networks 396

1 Introduction

1.1 About This Document

The Getting Started document describes how to perform common application packaging tasks with the help of PACE Suite.

1.2 Glossary

MSI	MSI is an installer package file format used by Windows. Its name comes from the program's original title, Microsoft Installer, which has changed to Windows Installer. MSI files are used for installation, storage, and removal of programs.
MST	MST is a settings file used by Microsoft Windows Installer, a component of the Windows operating system that enables software installations. It contains software configuration options, allows custom parameters for the installation.
APPV	Microsoft Application Virtualization (also known as App-V; formerly Softricity SoftGrid) is an application virtualization and application streaming solution from Microsoft.
APPX	APPX is the file format used to distribute and install Universal Windows Platform (UWP) apps on Windows 8.x and 10, Windows Phone 8.1, Windows 10 Mobile, Xbox One, Hololens, and Windows 10 IoT Core.
MSIX	MSIX is the file format used to distribute and install Universal Windows Platform (UWP) apps on Windows 10, 8/8.1, 7, MacOS, iOS, Android, Linux, Web.

2 PACE Overview

PACE Suite consists of the following components:

Tool	Major functions
	• Repackage any existing installation into Windows Installer (MSI), UWP app package (MSIX/APPX) or a virtualization format (Microsoft App-V 5.x, and VMware ThinApp). • Auto-detection of embedded installers. • Create a response transform • Include the excluded files/registry back to a package.
	• Manage the contents of an MSI in a tree view • Import the desired resources • Manage Custom Actions • Integrate Device Drivers • Use a smart and advanced MSI database editor, with formatted string autocompleting, Excel-like formula bar, row reference tracking, and more. • Create patches (MSP) and Merge Modules (MSM) • Manage permission settings for file system and registry • Handle upgrades easily – just let MSI Editor know which MSI you want to be upgraded at runtime. • Publish your applications to Microsoft SCCM 2007, 2012, 2016. • See the estimated complexity of an app right away. • Generate package documentation containing the details about your package and configuration. • Undo-redo any manipulation and see the changes highlighted in the MSI tables
	• Automatically record your on-screen actions into a nice-looking document with screenshots and annotations

2.1 Installation & Activation

2.1.1 Software Prerequisites

Ensure that a target system contains necessary software prerequisites:

OS	Windows 10/8.1/8/7 SP1, Windows Server 2016/2012/2008 R2 SP1 NOTE Building Universal Windows Platform app packages (MSIX/APPX) is available under Windows 10 or Windows Server 2016 and newer.
Middleware	Microsoft .NET Framework 4.7.2 (https://dotnet.microsoft.com/download/dotnet-framework-runtime) or higher. NOTE For building THINAPP packages, ensure that your system contains VMware ThinApp (5.0.0/5.0.1/5.1.0/5.1.1/5.2.0) installed and activated with the valid license. NOTE For publishing MSI packages to Microsoft SCCM 2007 (SP1/SP2/R1/R2) or SCCM 2012 (SP1/SP2/R2/R2 SP1) server, ensure that your system contains Microsoft Management Framework 3.0 and Windows Remote Management (WinRM) service enabled.
Privileges	Administrative privileges on the system.

2.1.2 Hardware Prerequisites

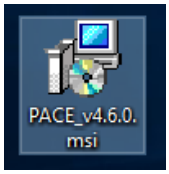
Ensure that a target system fulfils the minimum hardware conditions:

Processor	32-bit (x86) or 64-bit (x64) processor at 2 GHz or greater NOTE To repackage 64-bit applications or create 64-bit App-V packages, use PACE Suite on a 64-bit Windows operating system.
RAM	2 GB
Hard drive free space	1 GB NOTE Additional storage is required for your projects and packages and depends on their size and quantity.
Display	1280 x 768 resolution or higher.

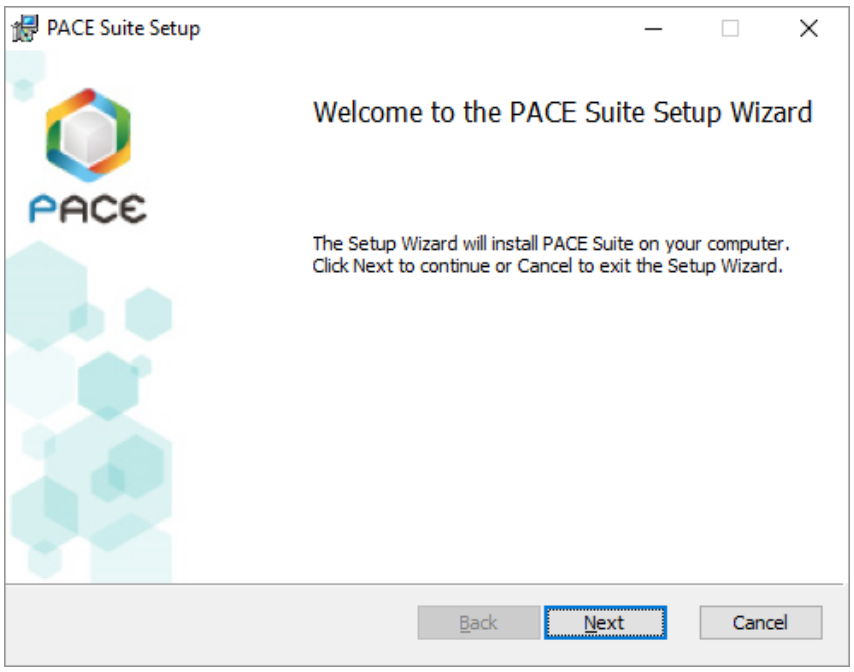
2.1.3 Installing PACE Suite

PACE Suite installation checks whether your license is compatible with the current PACE Suite version and whether your system meets minimum software requirements. Note that the installation requires the administrator privileges.

- Run the downloaded PACE Suite installater.

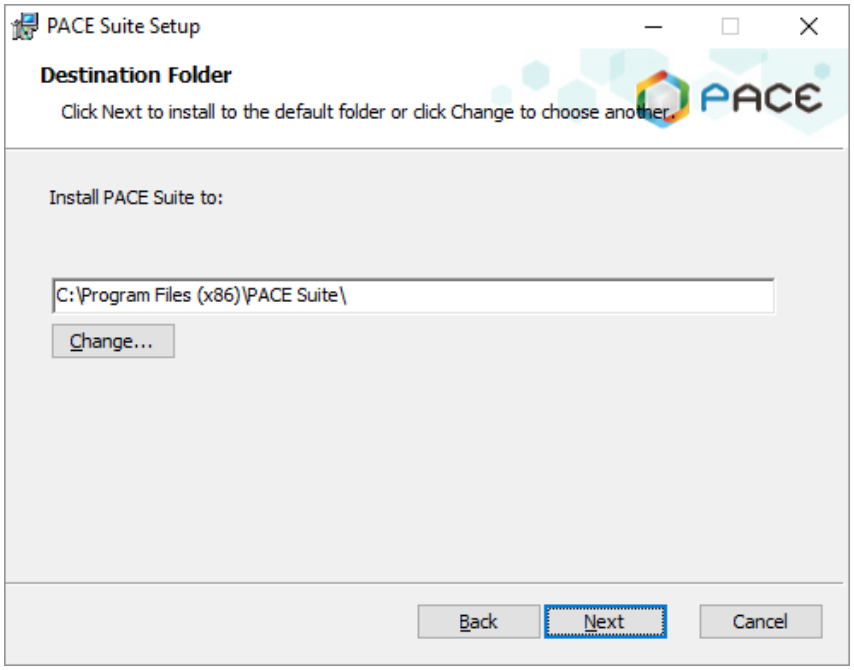


- Click Next.

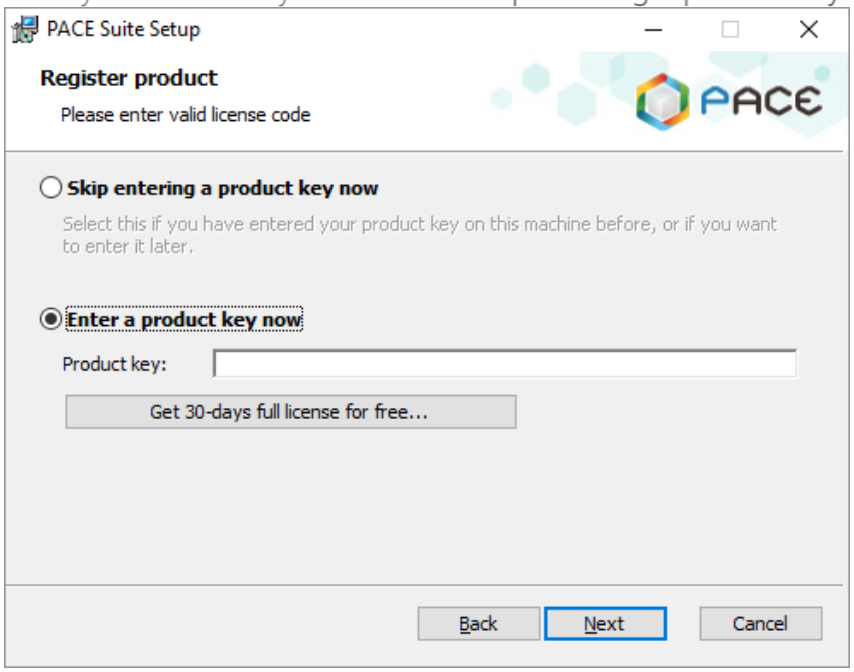


- Read and accept the END USER LICENSE AGREEMENT and then click Next.

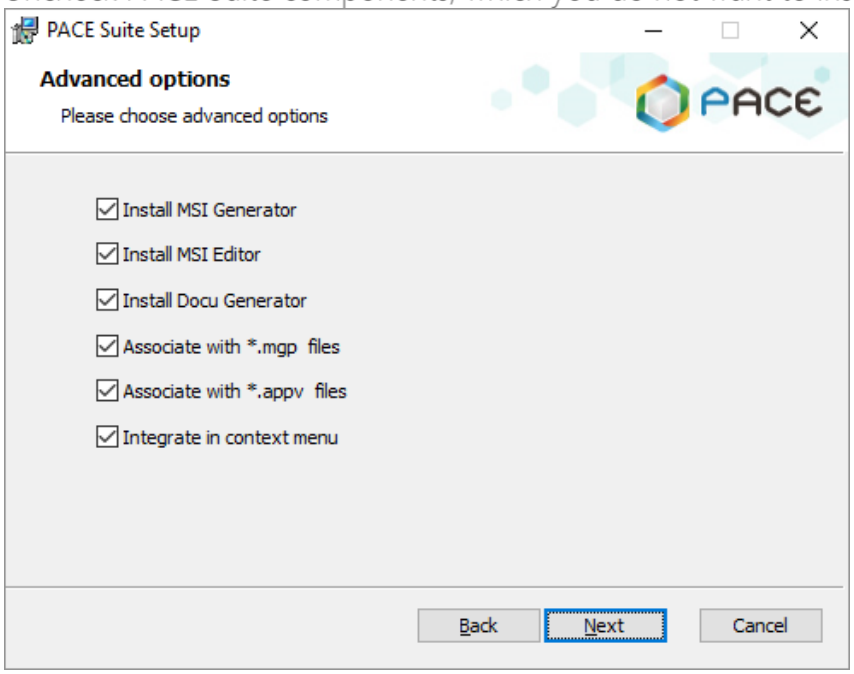
- Leave the default installation location or choose an alternative one and click Next.



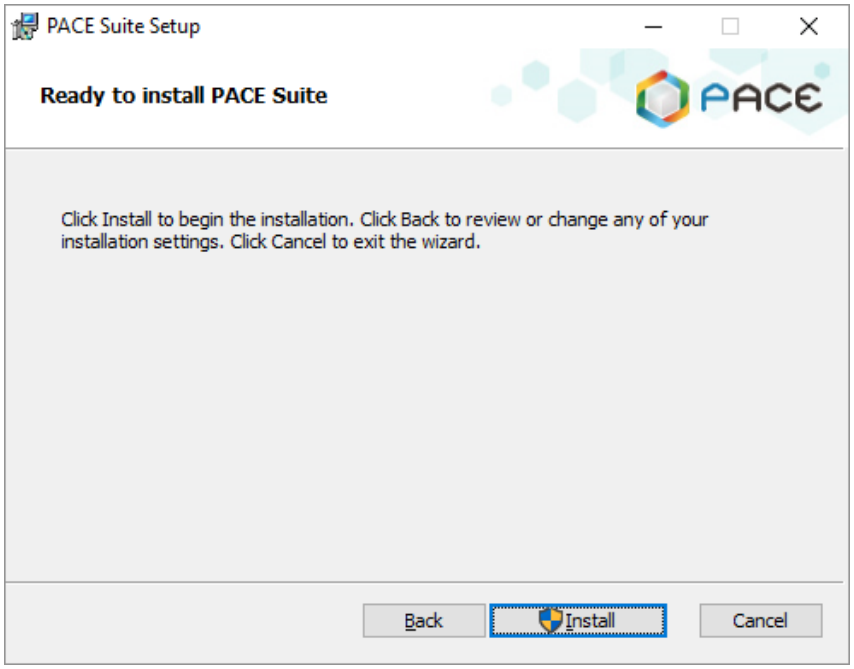
- Enter your license key or select the Skip entering a product key now option and click Next.



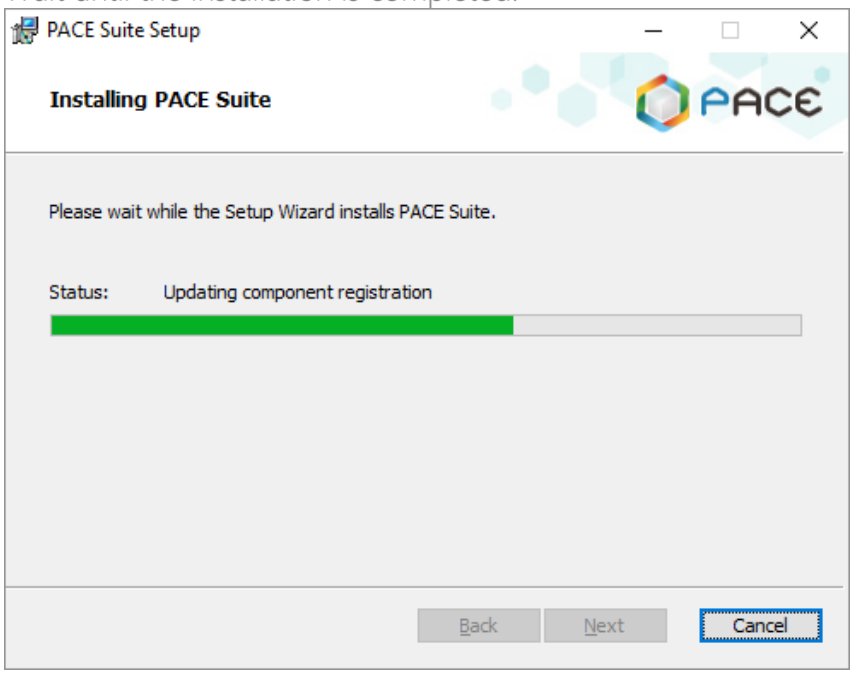
- Uncheck PACE Suite components, which you do not want to install, and click Next.



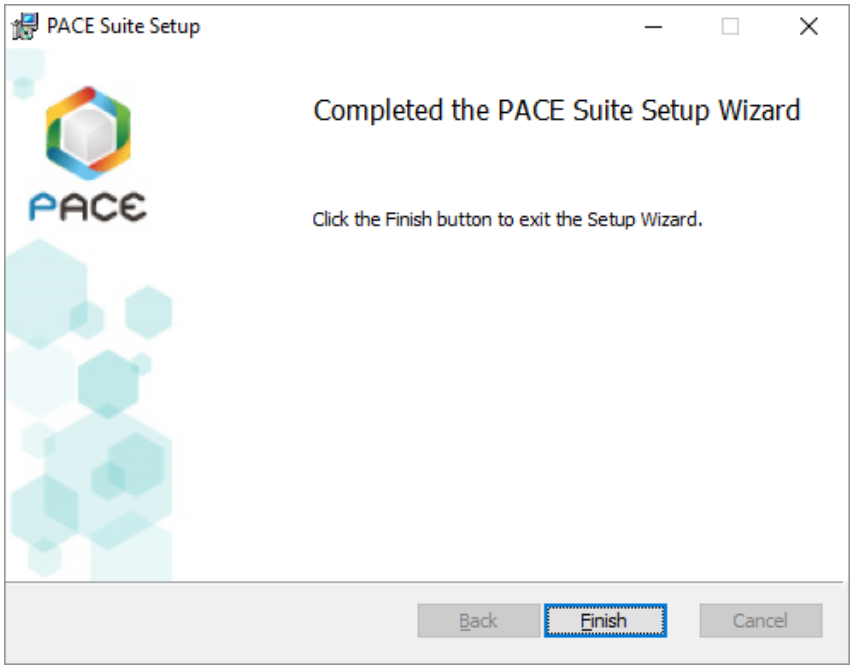
- Click Install.



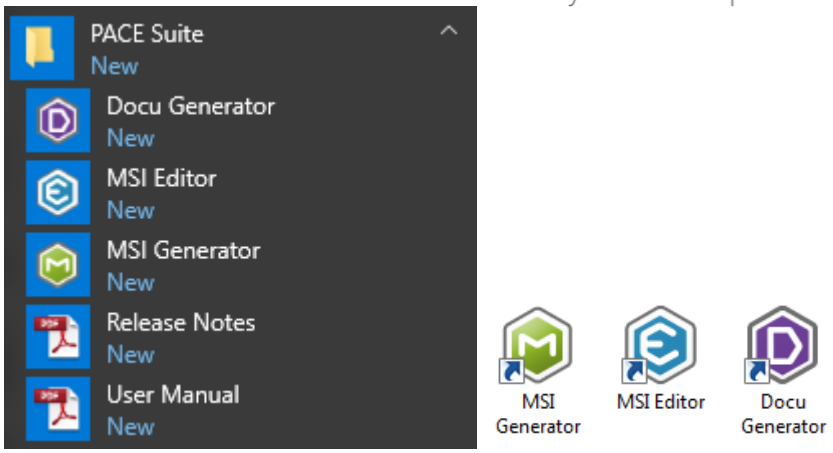
- Wait until the installation is completed.



- Click Finish.



- Find the installed PACE Suite shortcuts on your Desktop and in the Start Menu.

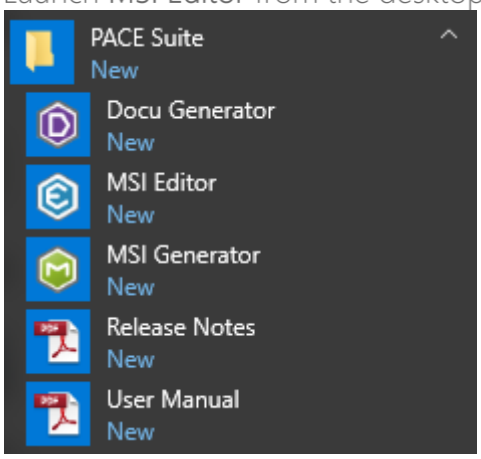


2.1.4 Activating PACE Suite

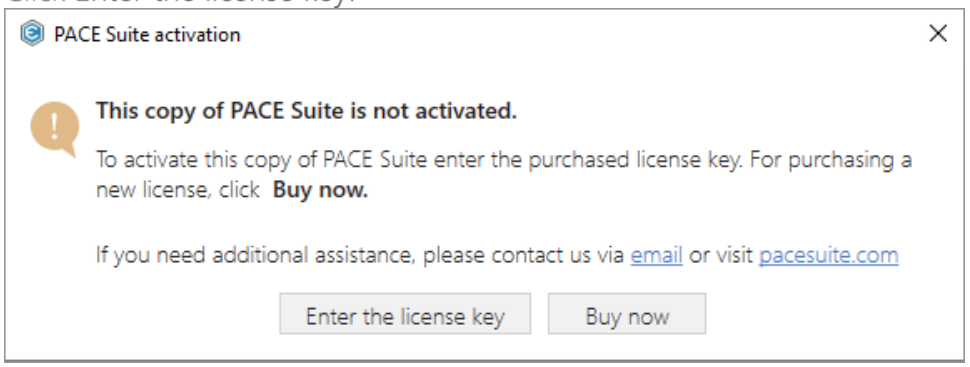
If you have installed PACE Suite with the **Skip entering a product key now** option or your current license is expired, you can activate your copy of PACE Suite with the provided license key.

As PACE Suite consists of MSI Editor, MSI Generator and Docu Generator, you have to activate all these products separately.

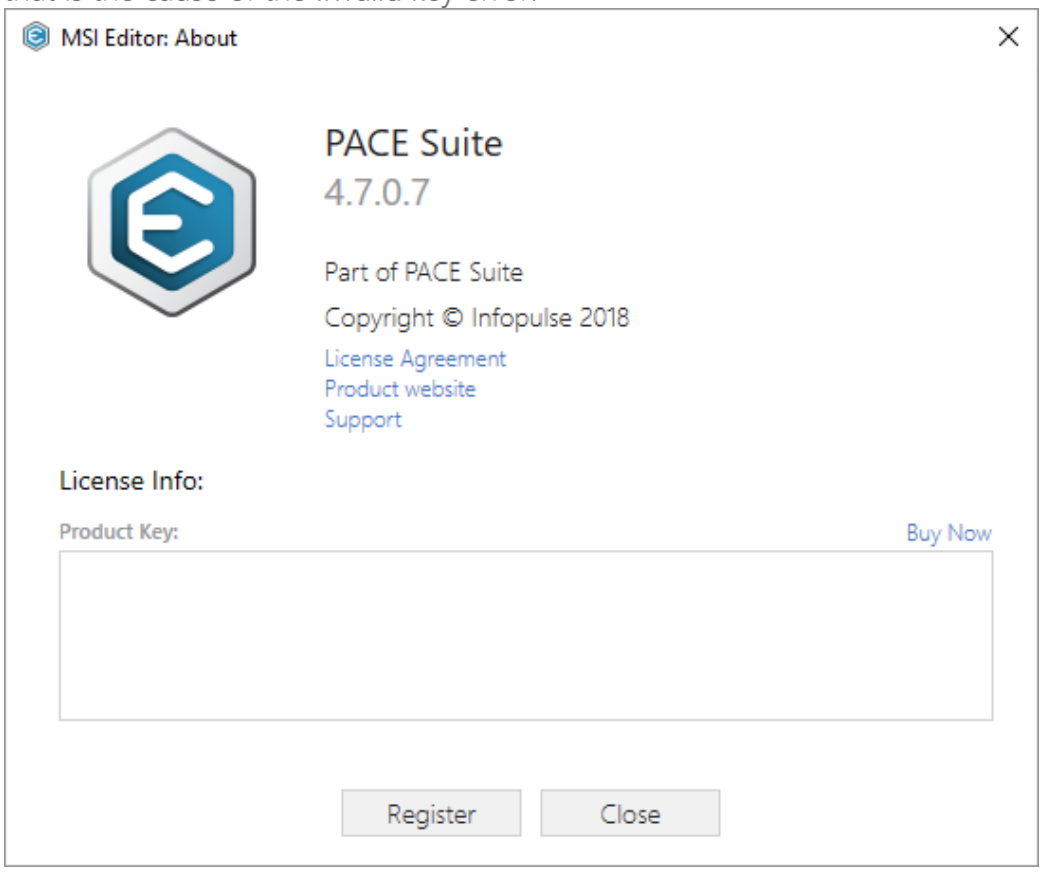
- Launch **MSI Editor** from the desktop or the start menu shortcut.



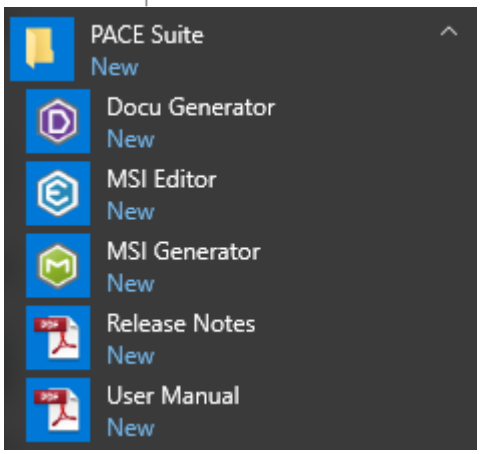
- Click **Enter the license key**.



- Enter your license key and click **Register**.
  NOTE If you are using the desktop version of Microsoft Outlook, ensure that the Auto Remove Line Breaks feature is disabled. Otherwise, your keys may be displayed incorrectly, and often that is the cause of the **Invalid key** error.



- Repeat the activation process for **MSI Generator** and **Docu Generator**, launching them from the desktop or the start menu shortcut.



3 How-To Manuals

We regularly publish PACE Suite educational videos on YouTube. Please subscribe to our channel.

3.1 Create MSI

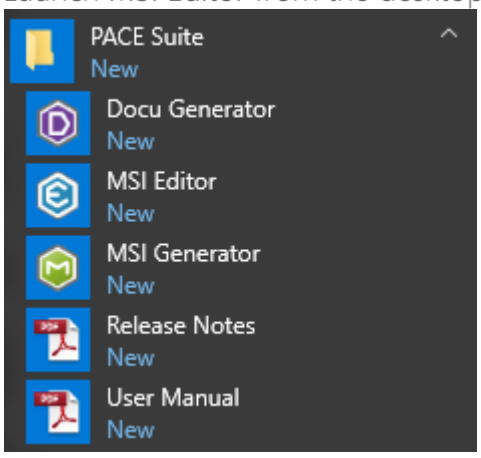
Choose a scenario that better suits your needs:

- Create MSI from Scratch, described in section 3.1.1
  Create a blank MSI in MSI Editor, add folders, files, registry entries and shortcuts directly to the MSI package.
- Repackage Installation to MSI, described in section 3.1.2
  Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into MSI package using MSI Generator. Along with files and registry, services and environment variables, you can capture the modified permissions to the file system and registry.
- Capture System Changes to MSI, described in section 3.1.3
  Capture system changes, which were made to the file system and registry, by script, by application or manually and save them into MSI package using MSI Generator. Along with files and registry, services and environment variables, you can capture the modified permissions to the file system and registry.

3.1.1 Create MSI from Scratch

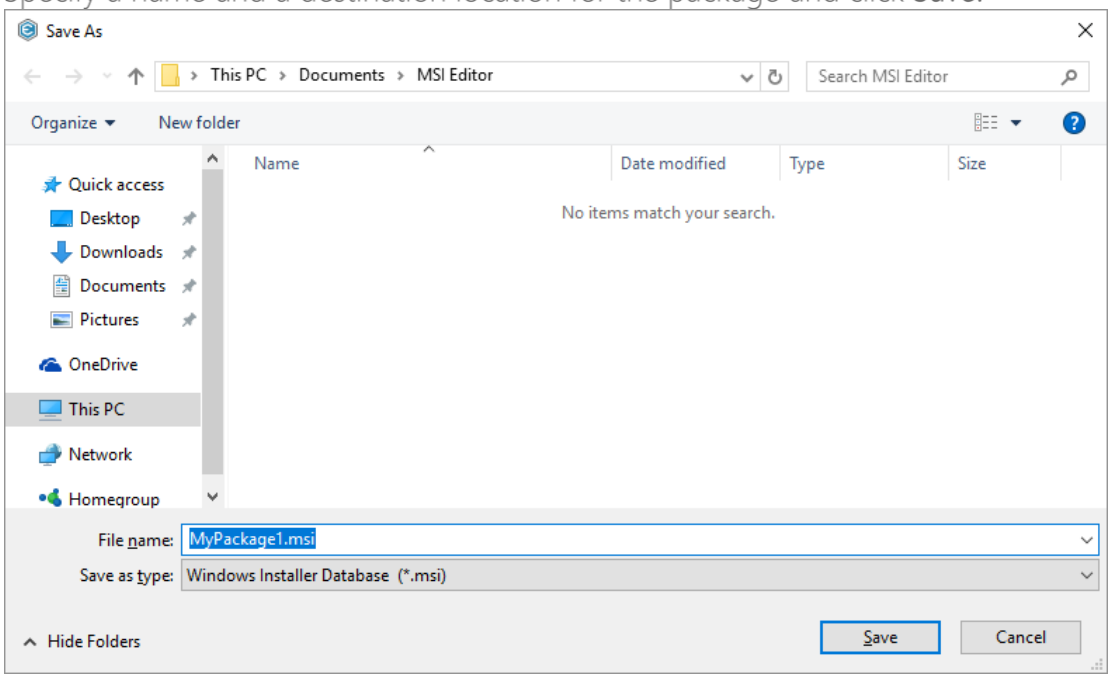
Create a blank MSI in MSI Editor, add folders, files, registry entries and shortcuts directly to the MSI package.

- Launch MSI Editor from the desktop or the start menu shortcut.



- Select New MSI... from the MENU to create a new empty MSI package.

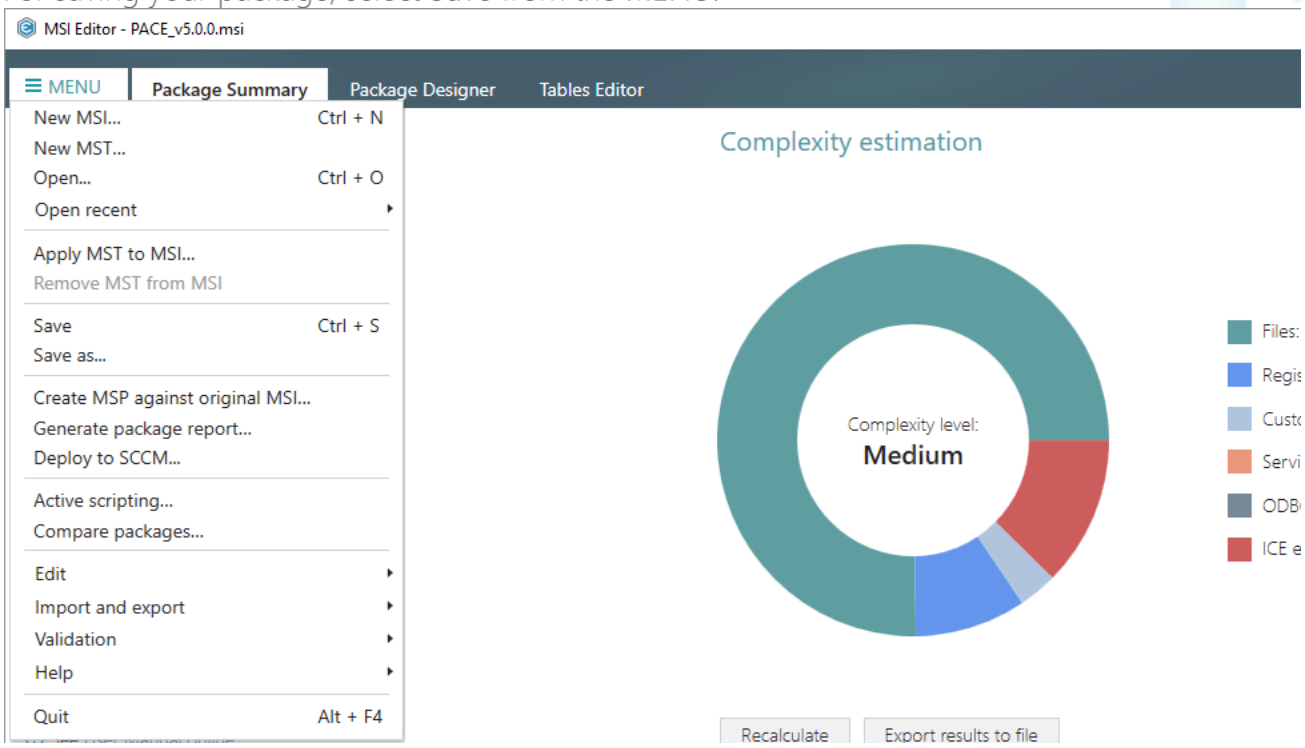
- Specify a name and a destination location for the package and click Save.



- For managing package resources and settings, refer to the respective section of this document:
    - For managing Application Details see section 3.7.2
      Manage product name, version, language and manufacturer of the application.
    - For managing Package IDs see section 3.7.3
      Manage product, upgrade and package codes.

- For managing Files and Folders see section 3.7.4
  Review package files and folders, add new folders, import files, set permissions, register COM objects and .NET assemblies, set the INSTALLDIR folder and the like.
- For managing Shortcuts see section 3.7.5
  Review package shortcuts, add new or import existing shortcuts, manage their properties.
- For managing Registry see section 3.7.6
  Review package registry, add new or import from REG file registry keys and values, integrate ActiveSetup, set permissions and the like.
- For managing Custom Actions see section 3.7.7
  Manage custom and standard actions of the package, add new one to run scripts, EXE, etc.
- For managing Drivers see section 3.7.8
  Review the package device drivers, modify and add new ones.
- For managing Features and Components see section 3.7.9
  Review the feature-component structure and resources, linked with components. Add new features, components and manage their properties.
- For managing ARP Appearance see section 3.7.10
  Control how the package will be displayed in the App list (Add & Remove Programs).
- For managing Upgrades see section 3.7.11
  Manage the package upgrade settings to automatically upgrade previous MSI packages.
- For managing MSI Properties see section 3.7.12
  Manage package properties, defined in the Property table and those, which will be updated by the CustomAction, AppSearch, Control, ComboBox, CheckBox, RadioButton tables.
- For managing Summary Information see section 3.7.13
  Manage Summary Information of the opened MSI/MST.
- For managing MSI Tables see section 3.7.14
  Add, import, drop MSI tables.
- For managing Merge Modules, described in section 3.7.15
  Create new MSM or importing existing MSM to the package.

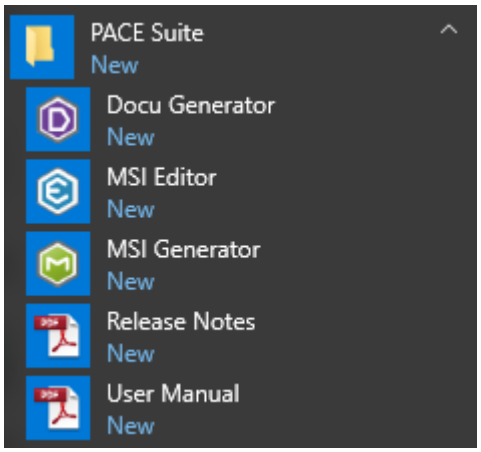
- For saving your package, select Save from the MENU.



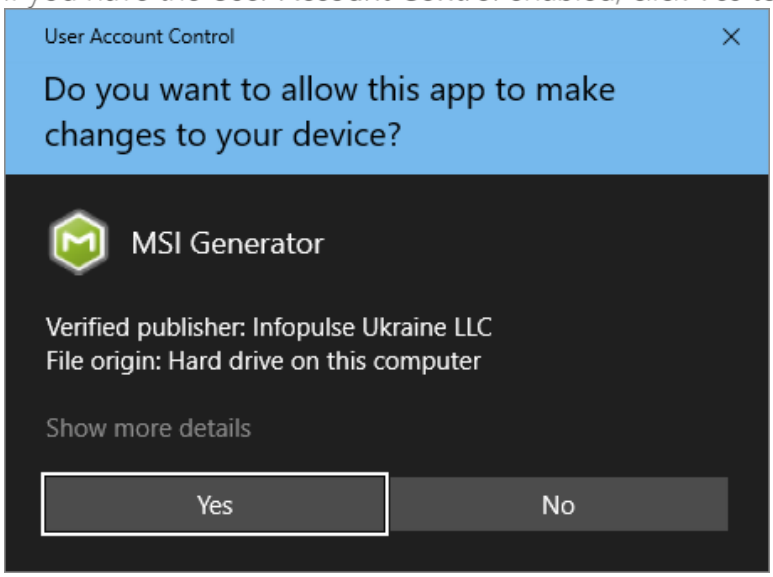
3.1.2 Repackage Installation to MSI

Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into MSI package using MSI Generator. Along with files and registry, services and environment variables, you can capture the modified permissions to the file system and registry.

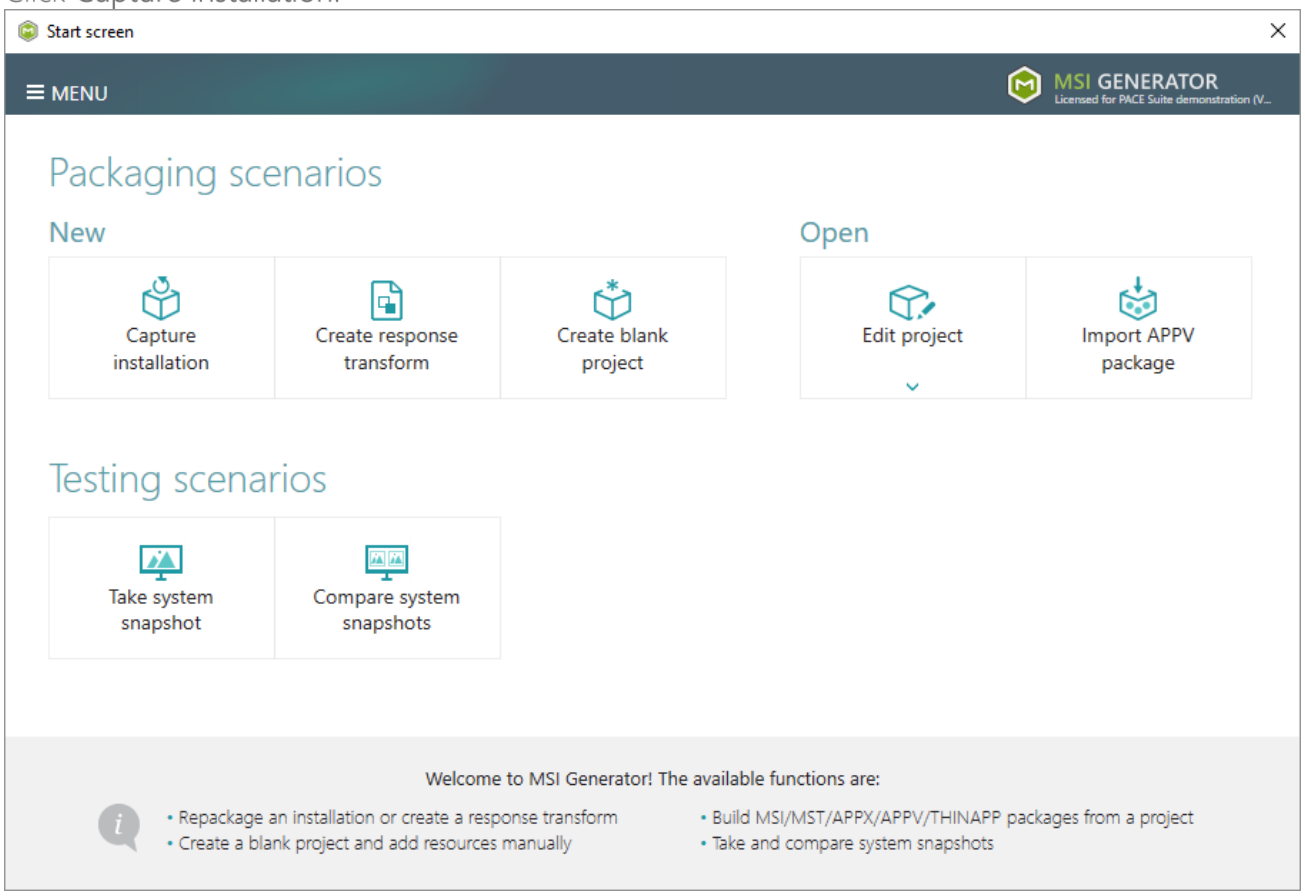
- Launch MSI Generator from the desktop of the start menu shortcut.



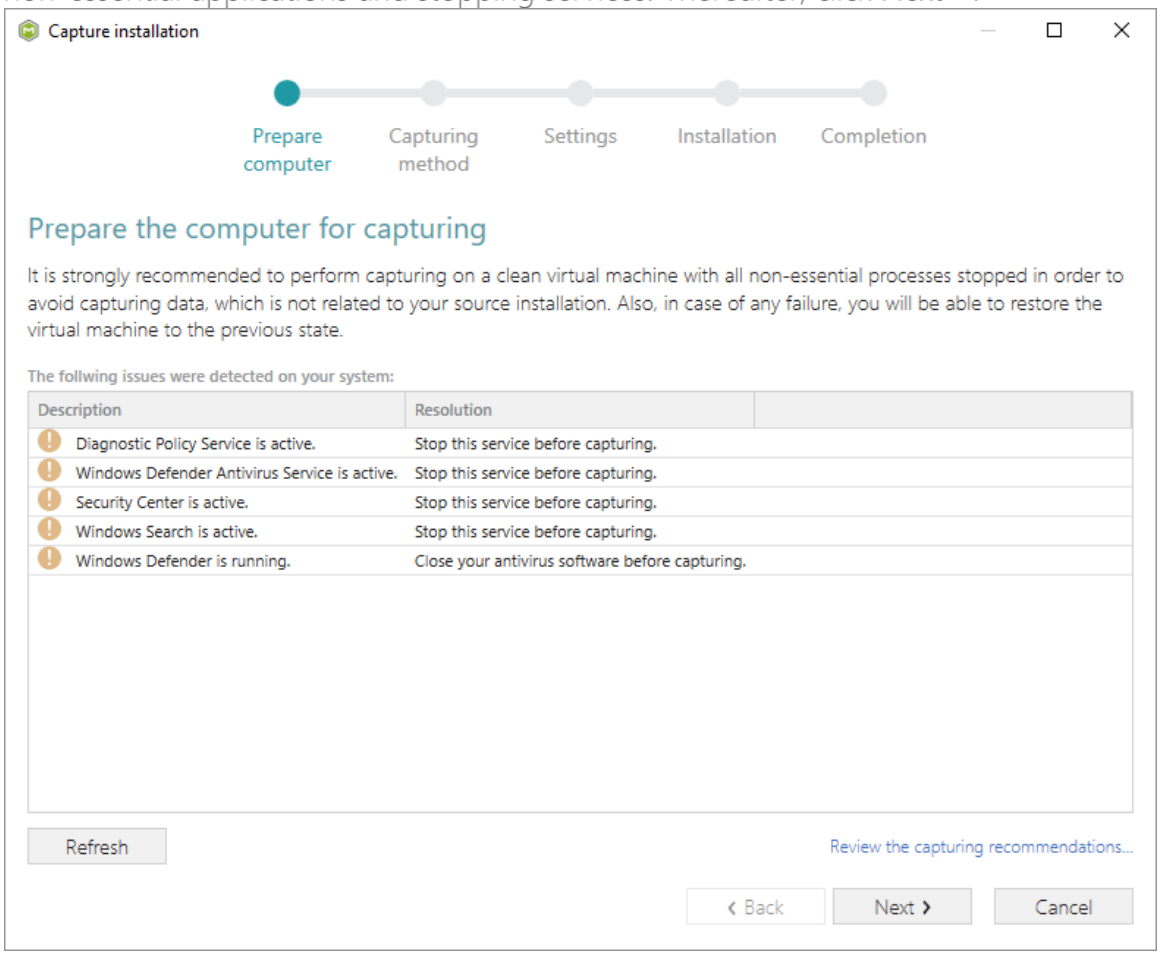
- If you have the User Account Control enabled, click Yes to launch the application.



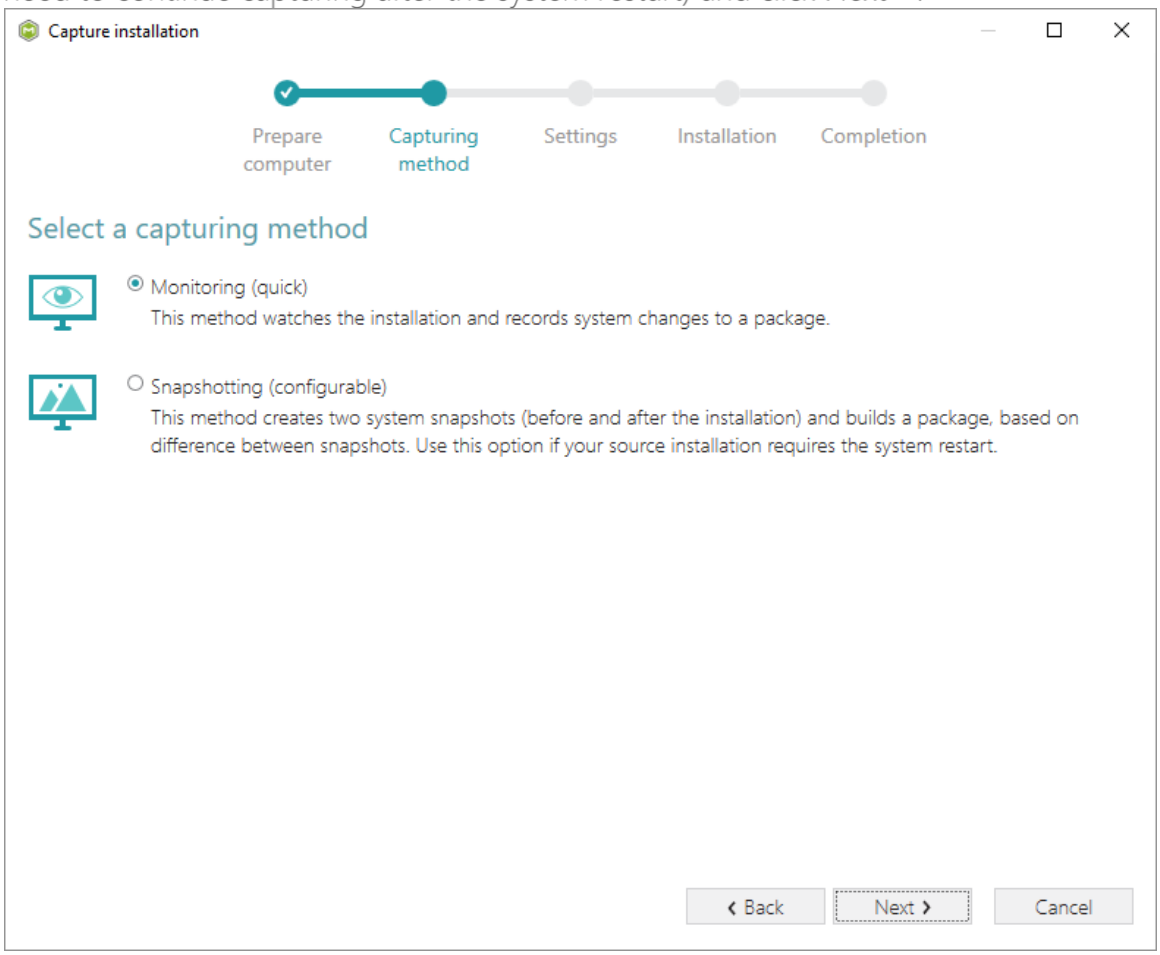
- Click Capture installation.



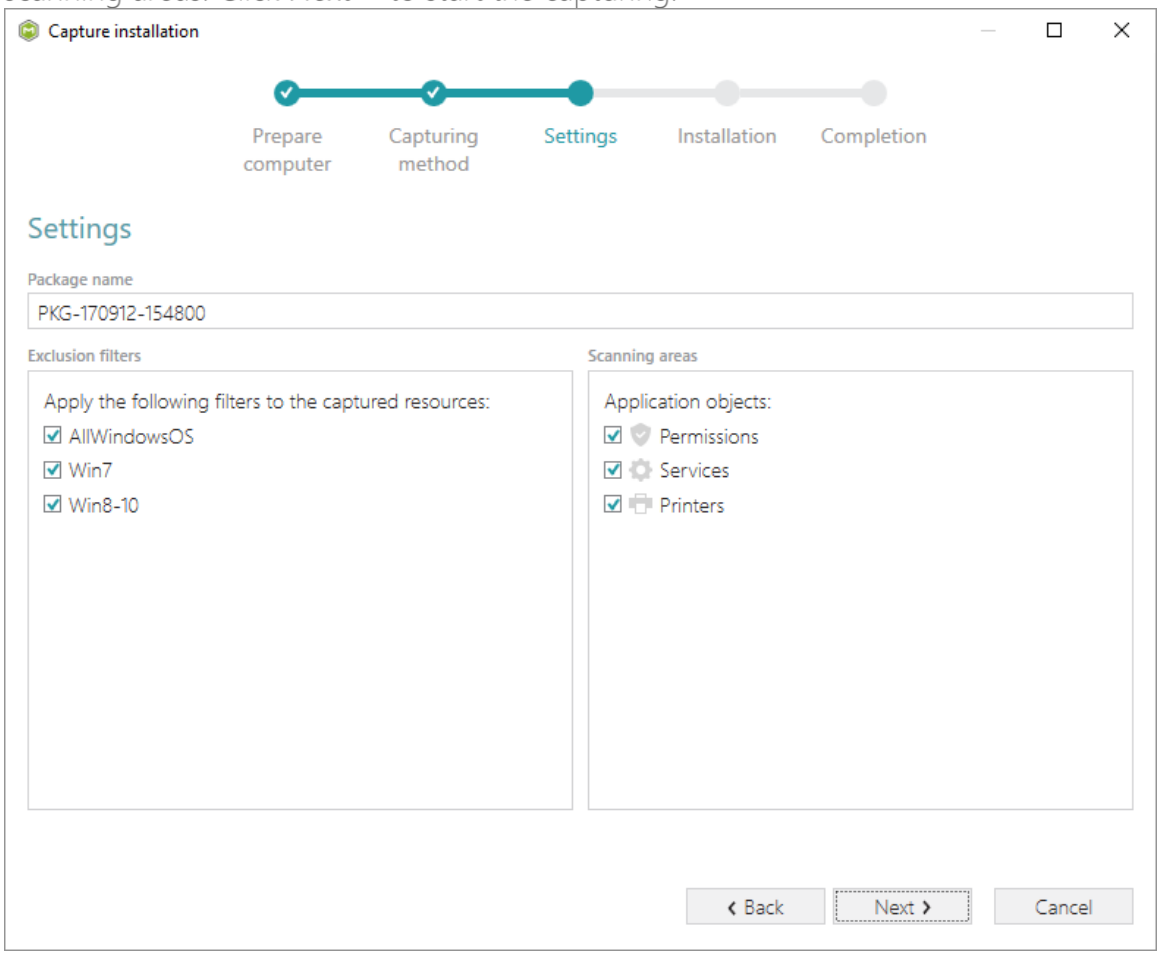
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



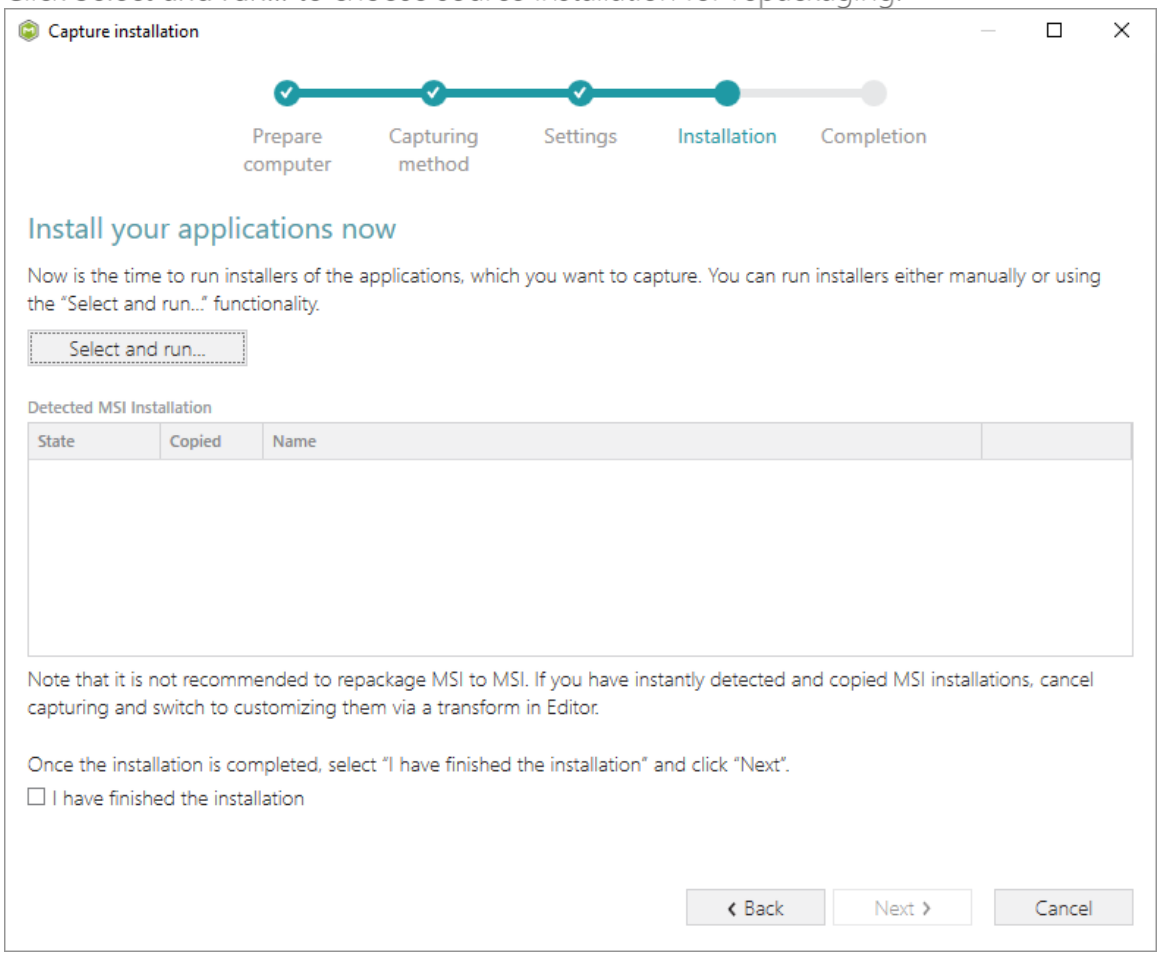
- Select the **Monitoring** method for the quicker capturing (or use the **Snapshotting** one if you need to continue capturing after the system restart) and click **Next >**.



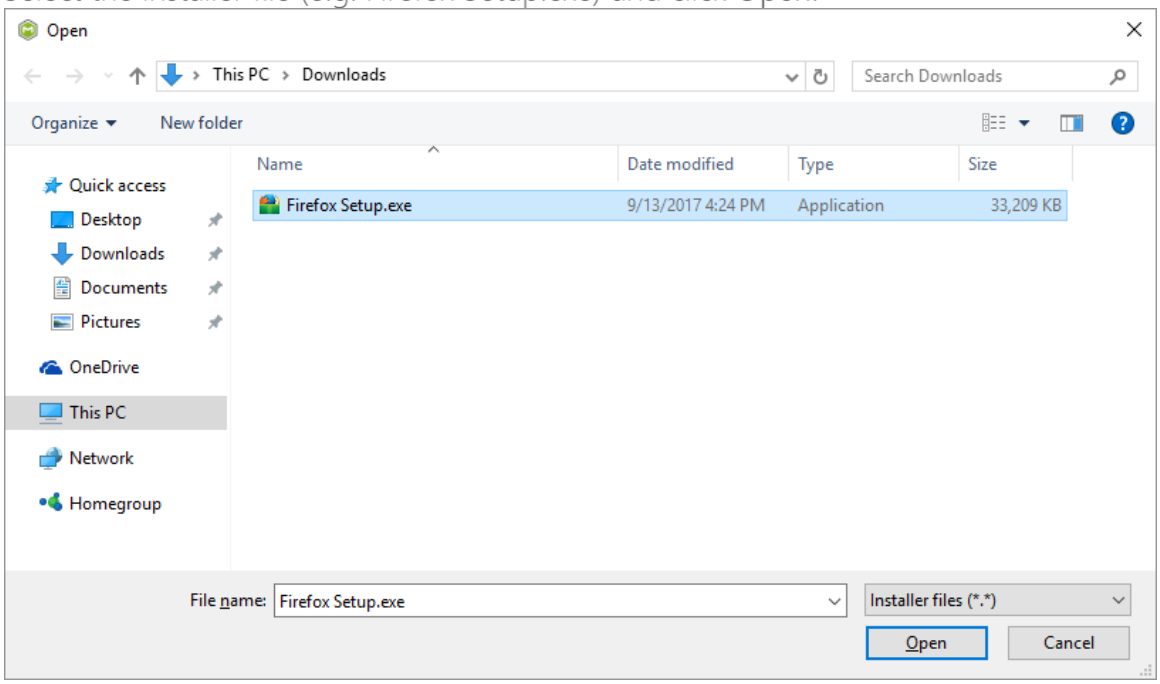
- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.



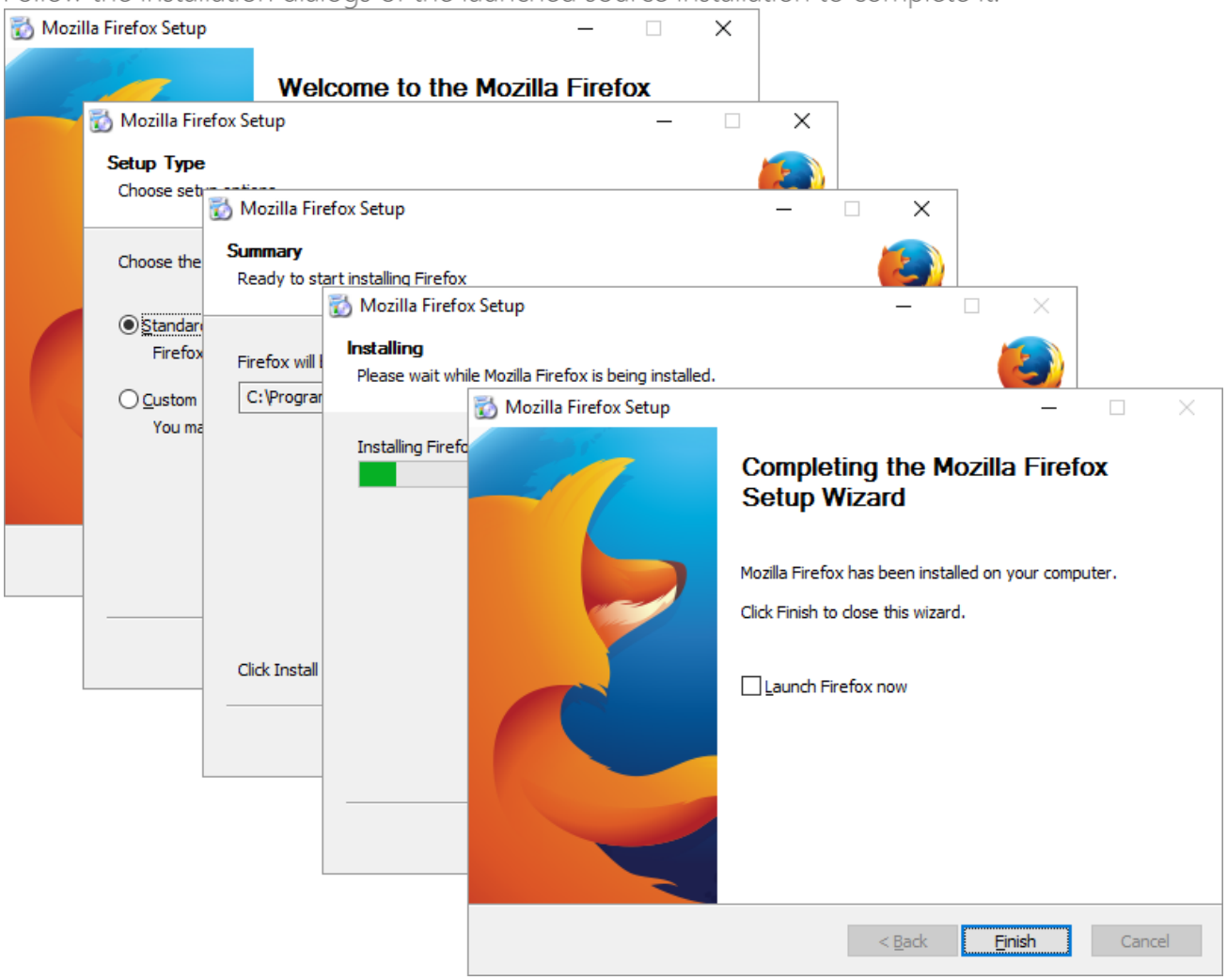
- Click Select and run... to choose source installation for repackaging.



- Select the installer file (e.g. Firefox Setup.exe) and click Open.

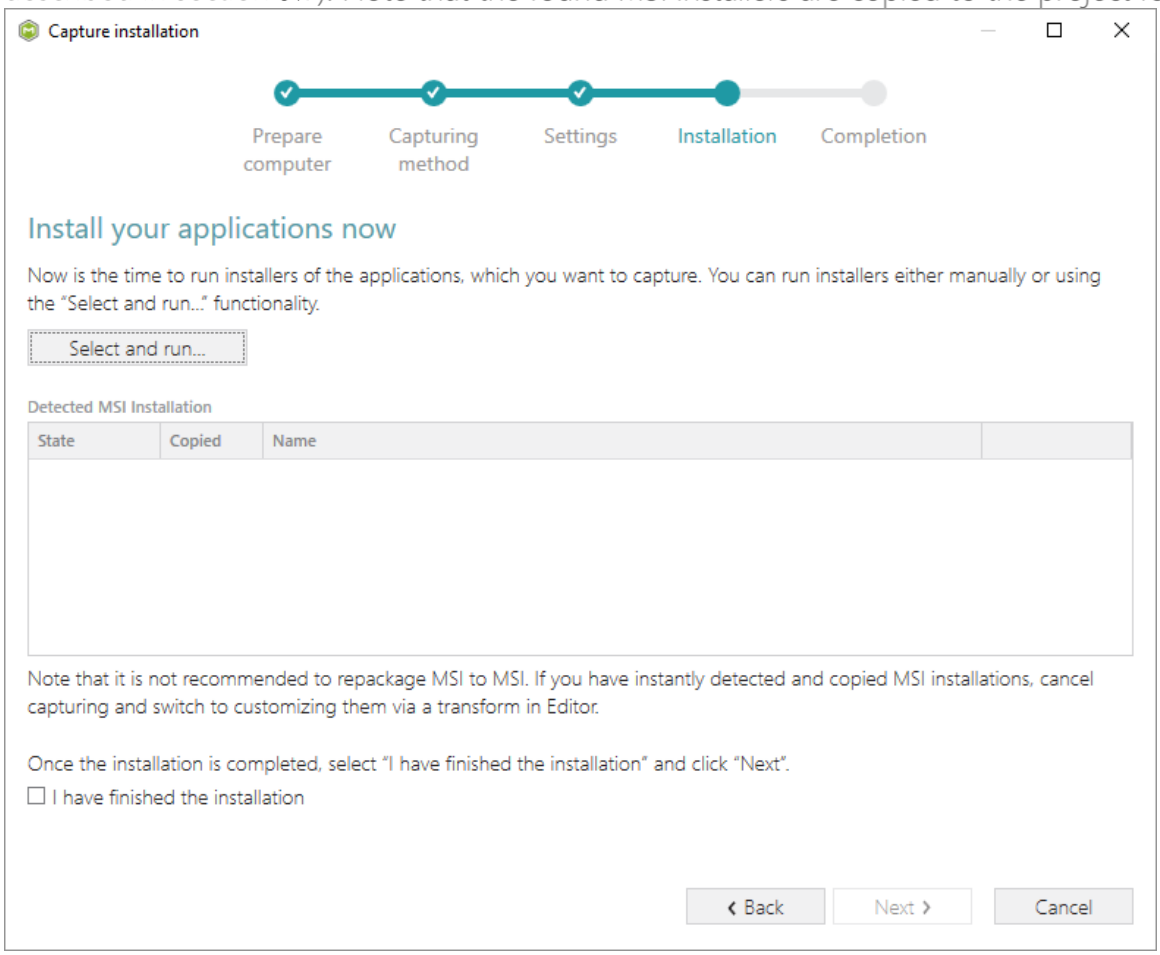


- Follow the installation dialogs of the launched source installation to complete it.



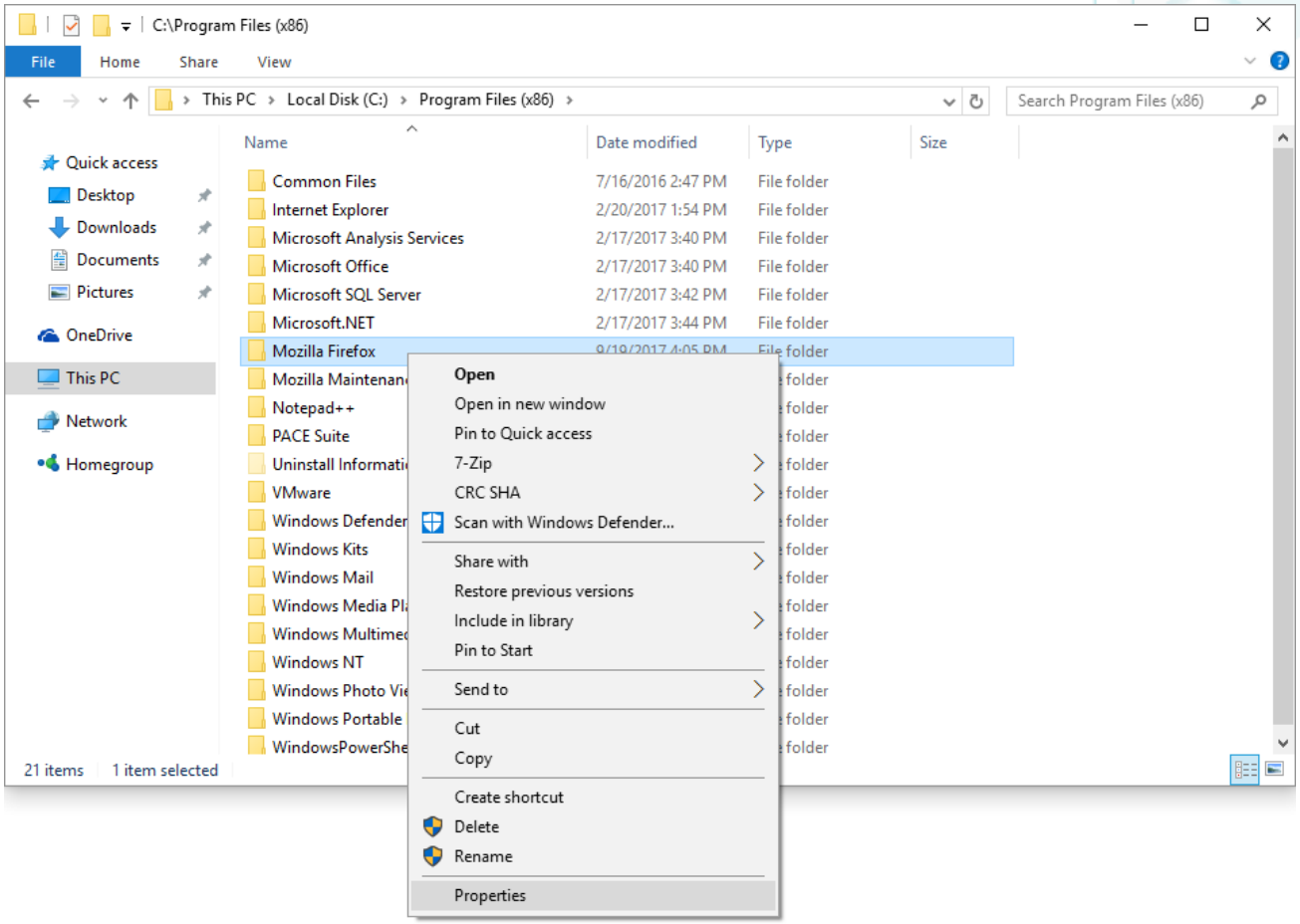
- Once the source installation is completed, we recommend to check if the Detected MSI installations tab does not contain found MSI installers. Even if your sources installation is an EXE file, it could contain embedded MSI installers, launched hiddednly. As you may know, repackaging existing vendor MSI installers into MSI is against Microsoft best practices. Thus, if the Detected MSI installations tab contains a found MSI installer, consider canceling

this capturing (repackaging) process and switching to editing it in MSI Editor (Edit MSI/MST is described in section 3.7). Note that the found MSI installers are copied to the project folder.

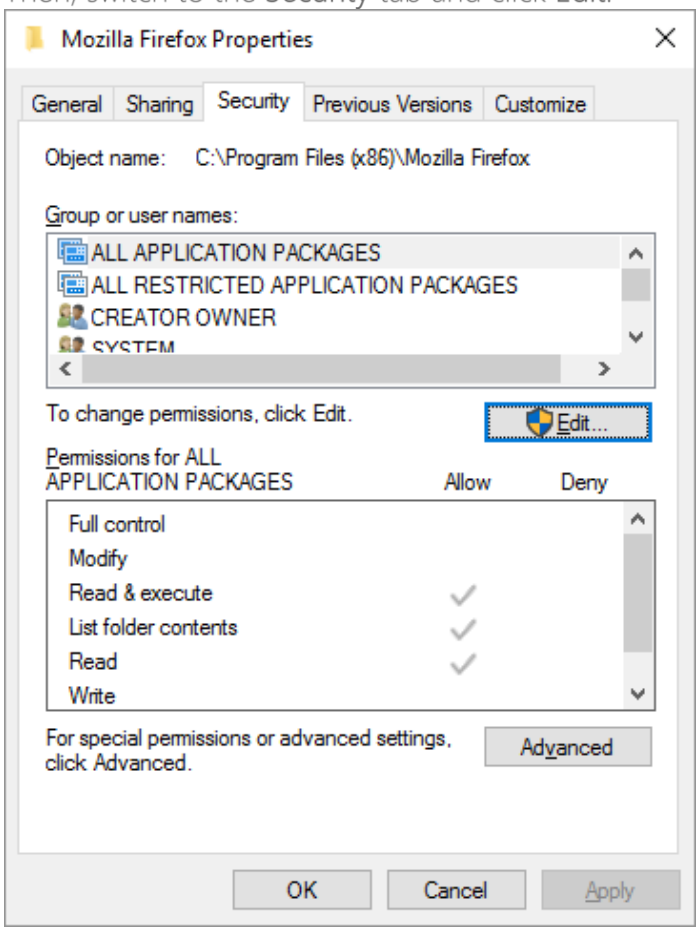


- Now you can make any additional changes to the file system and registry, which you want to capture along with source installation and include to the package. For instance, you can copy or edit configuration files, import REG file to the system registry, change file, folder permissions, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.

  In case of necessity, to change the default permission settings for a folder or a file, select Properties from the Windows Explorer context menu of a folder, for which you want to set new permissions.

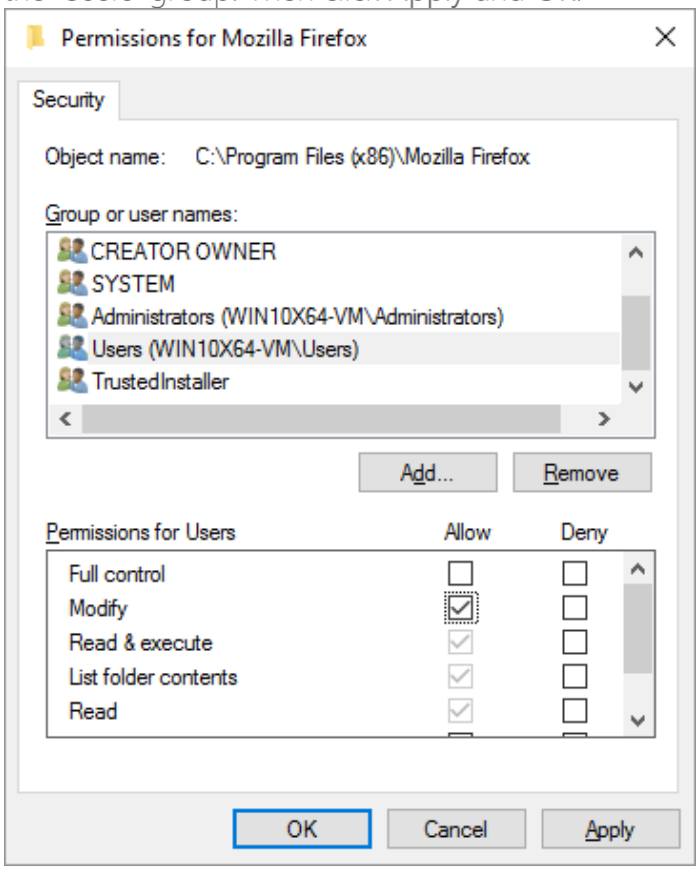


Then, switch to the Security tab and click Edit.

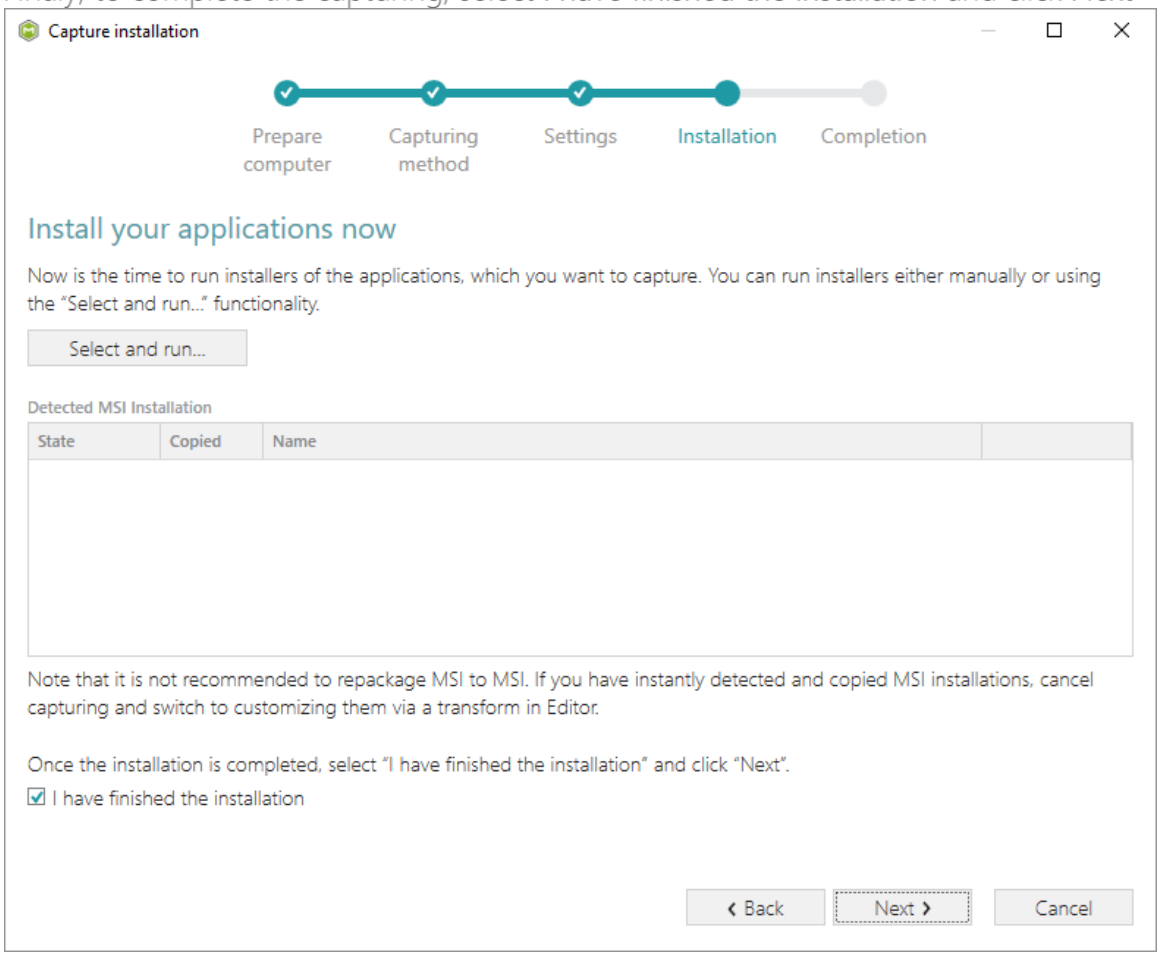


And select a group or a user, for which you want to change the permissions and then select the necessary access rights. For example, lets set the 'Write' and the 'Modify' access rights for

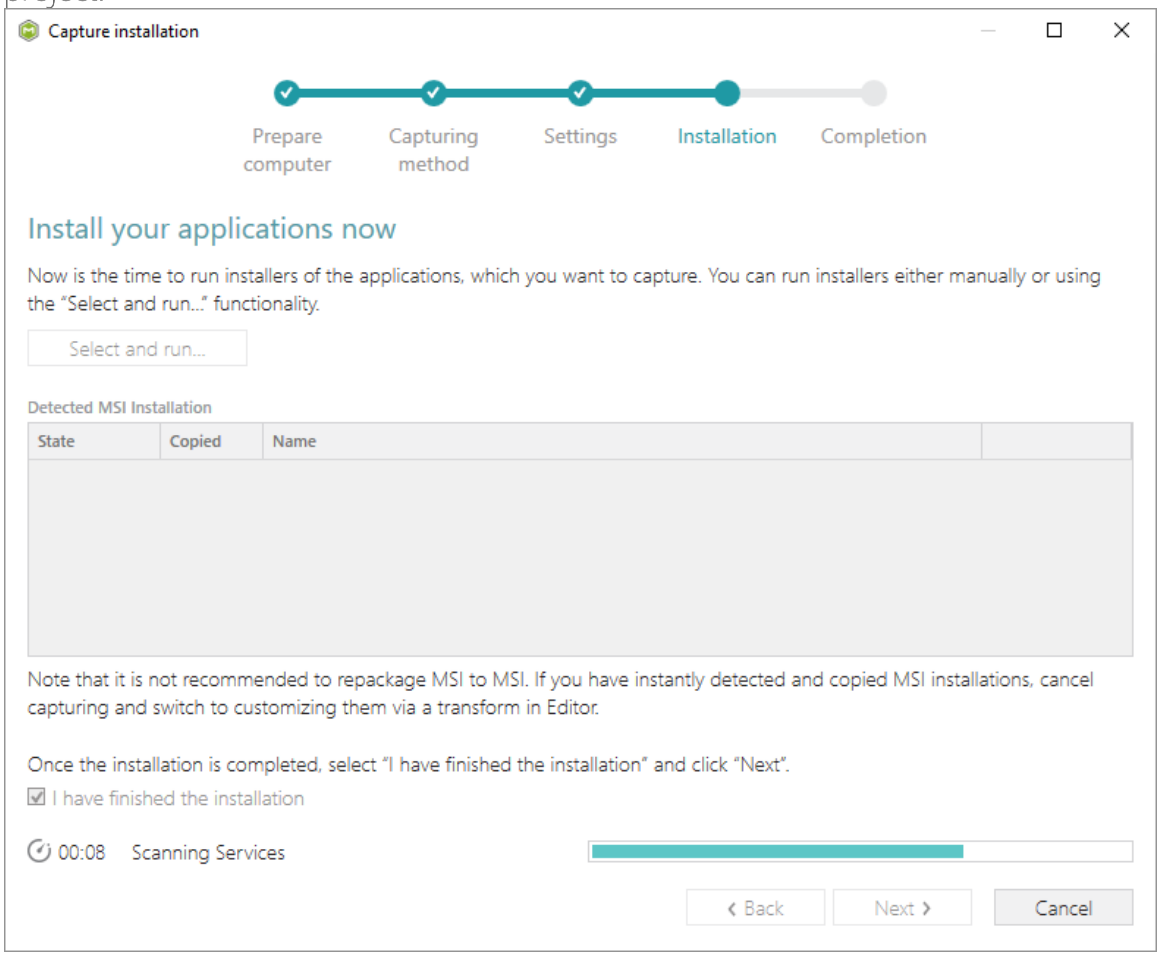
the 'Users' group. Then click Apply and OK.



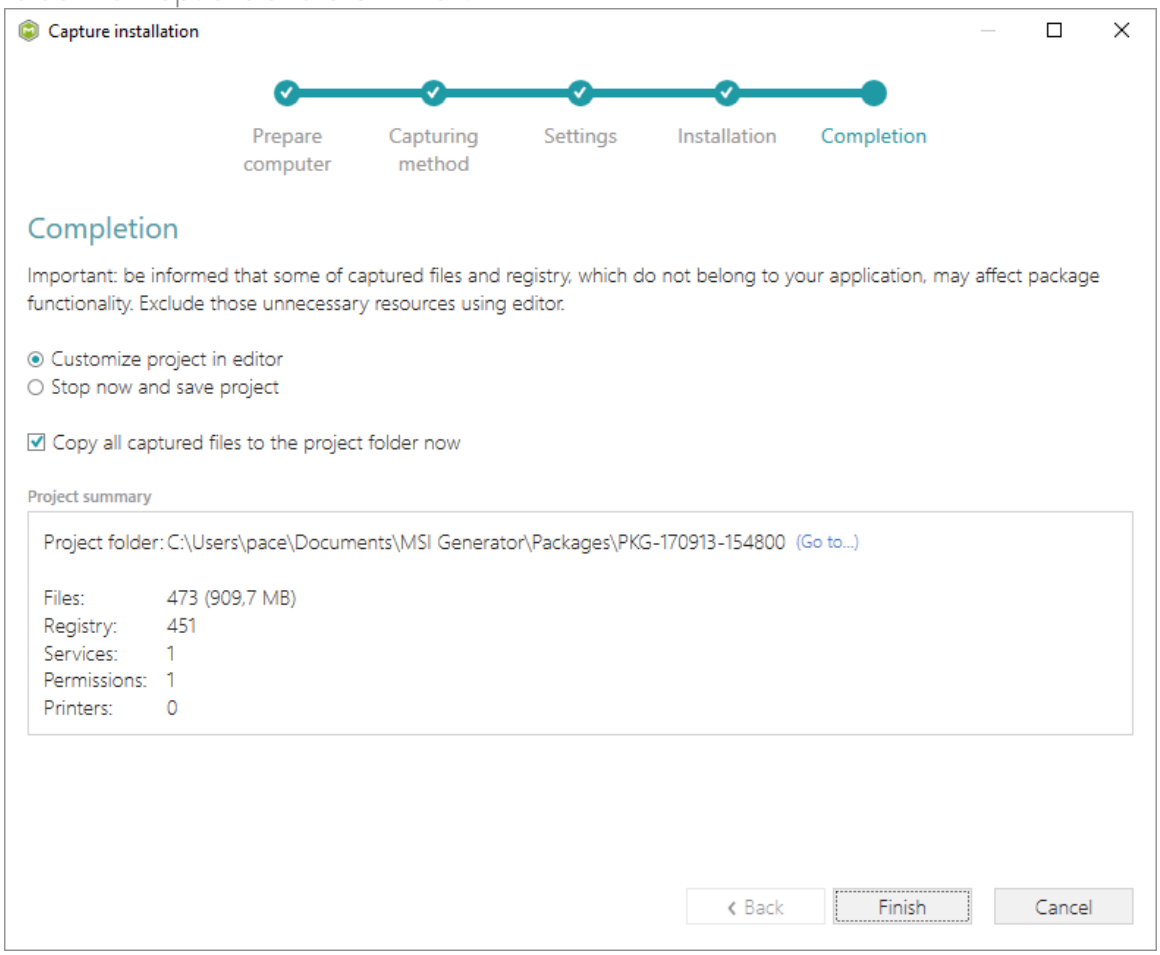
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



- Wait a little, while the capturing process is finishing, filtering captured data and creating the project.

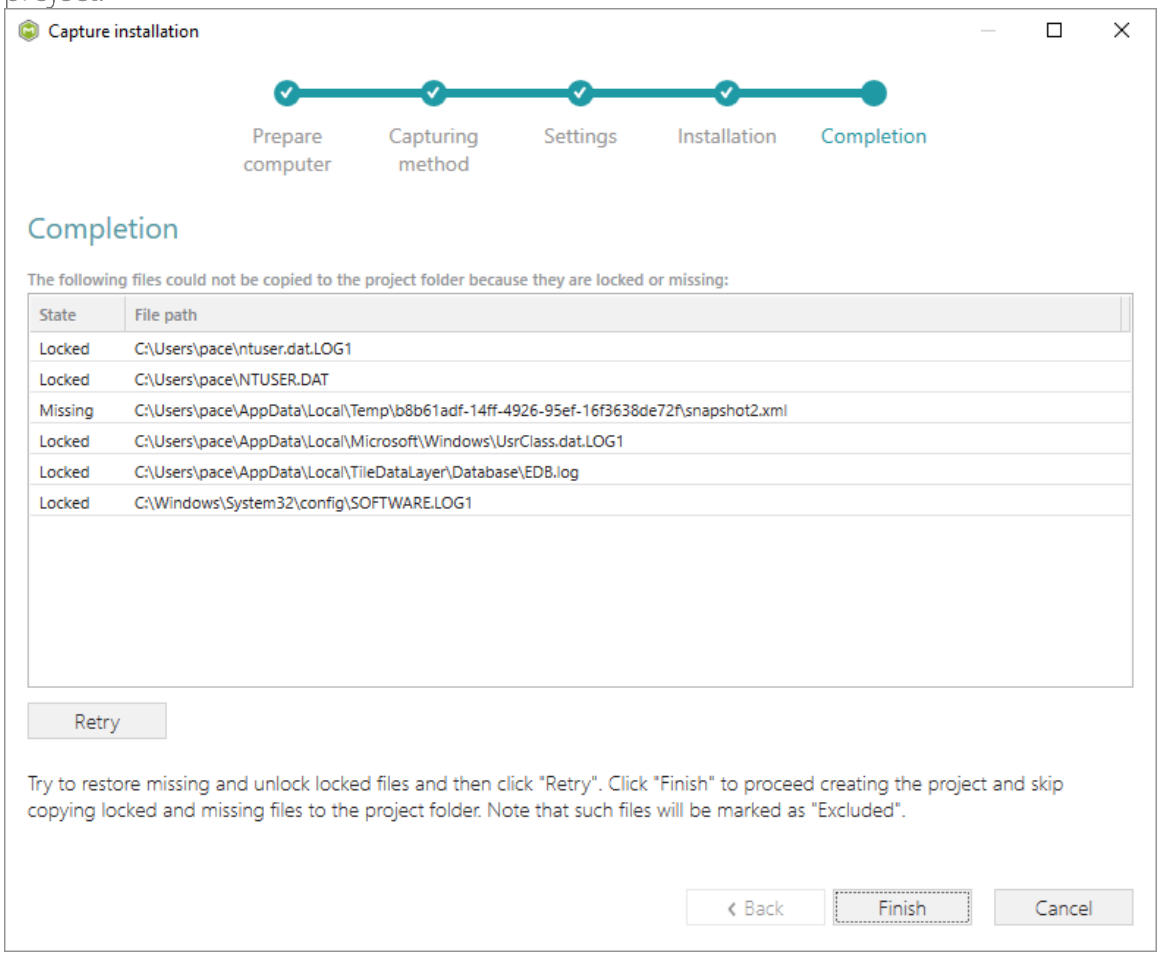


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



- If some of captured files are locked or missing and as result they could not be copied to the project folder, they will be displayed as illustrated below. Usually, the displayed files are system noise and do not refer to the captured application. Click Finish to continue with skiping locked and missing files. Note that such files will be automatically marked as 'Excluded' in your

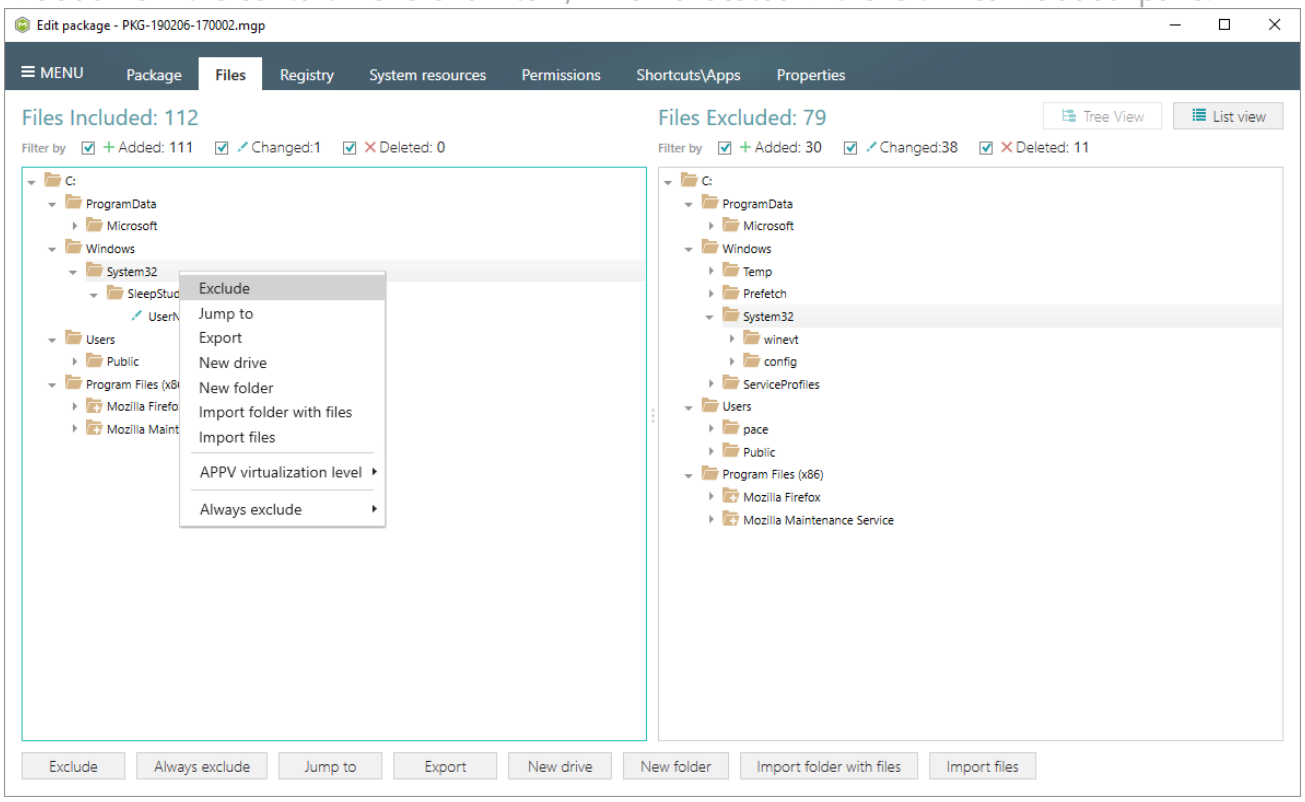
project.



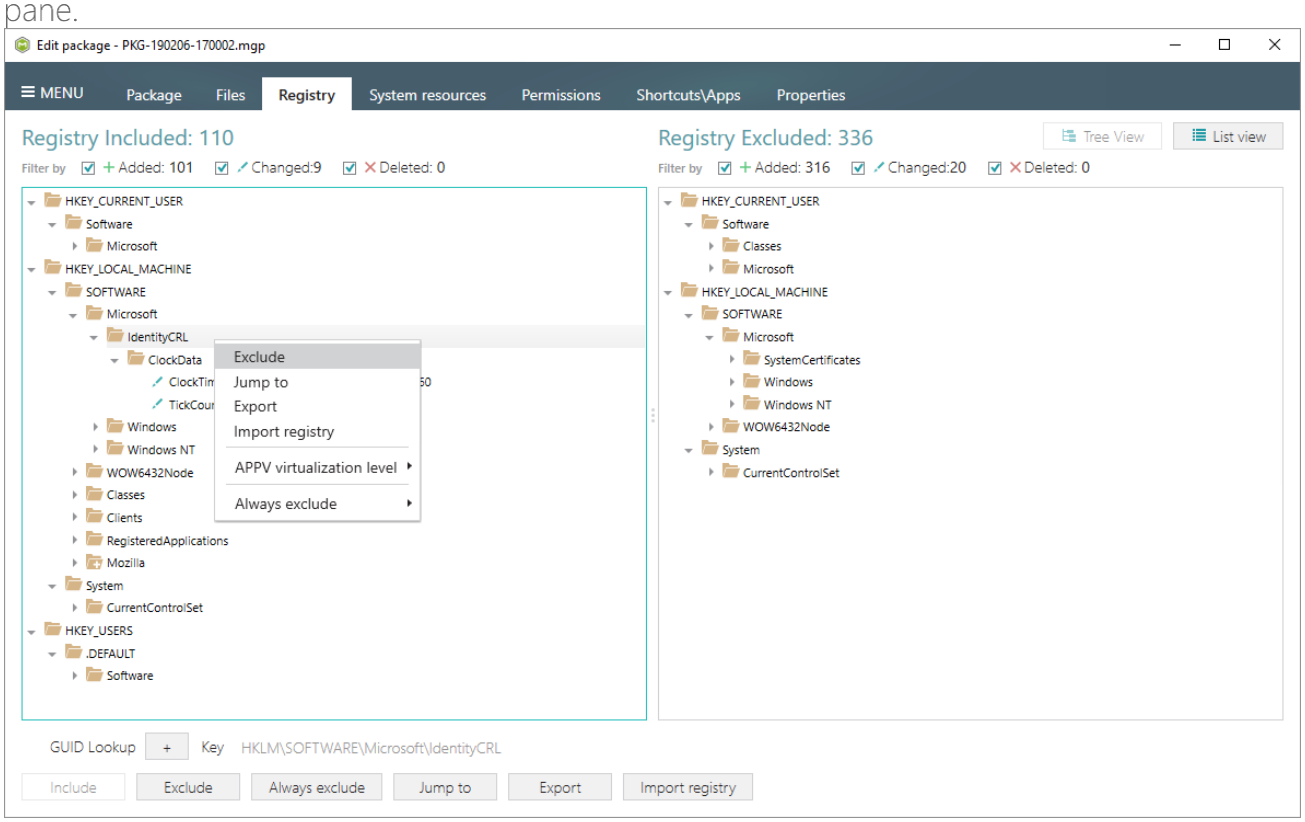
- Once the project is opened in the project editor, it is strongly recommended to review the Files, Registry, System resources, Permissions, and Shortcuts\Apps tabs and exclude the accidentally captured resources, which do not belong to the application. Such accidentally captured resources are also called system noise and it could be Windows Updates, Search indexing, Antivirus software and so on.

  An MSI package with such resources may fail during the installation. The common reason for such fails is trying to write a registry value to the system-protected location, strongly prohibited from writing by third-party installations.

- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.

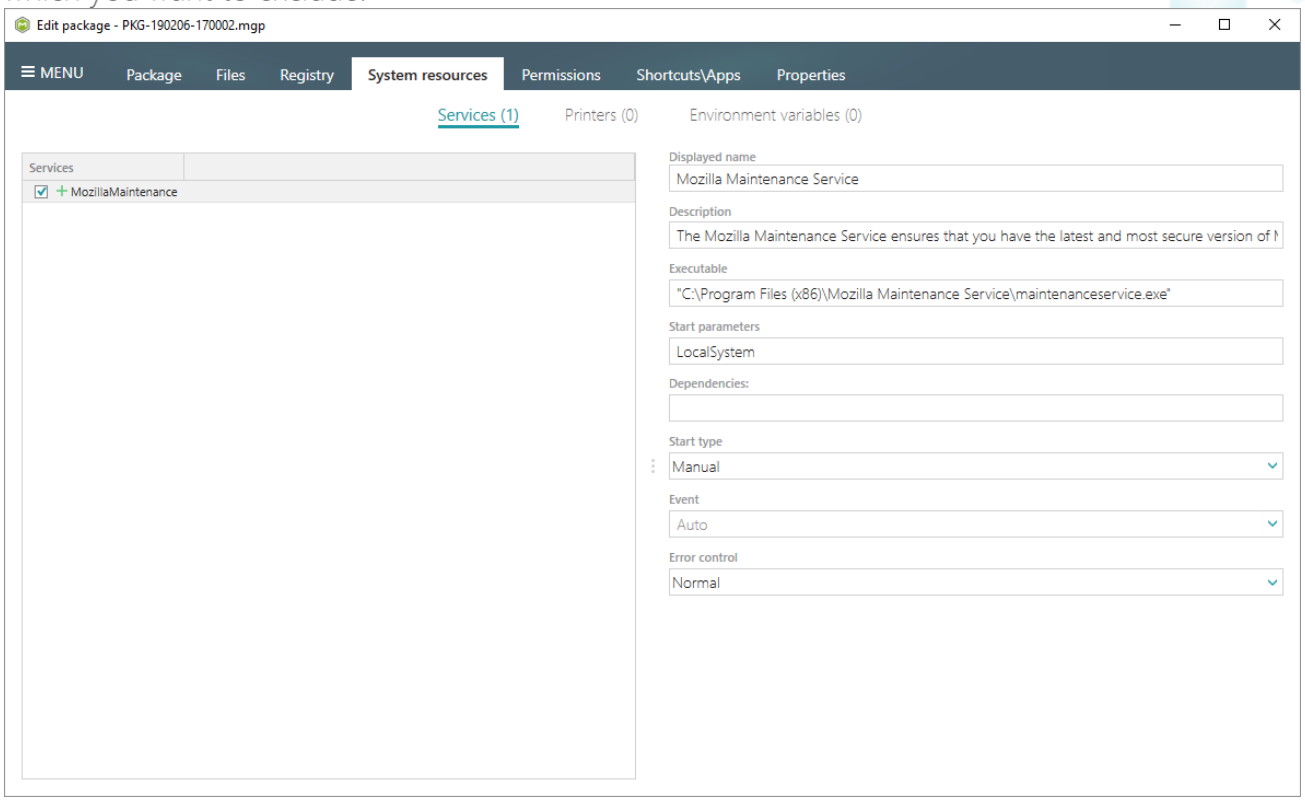


- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.

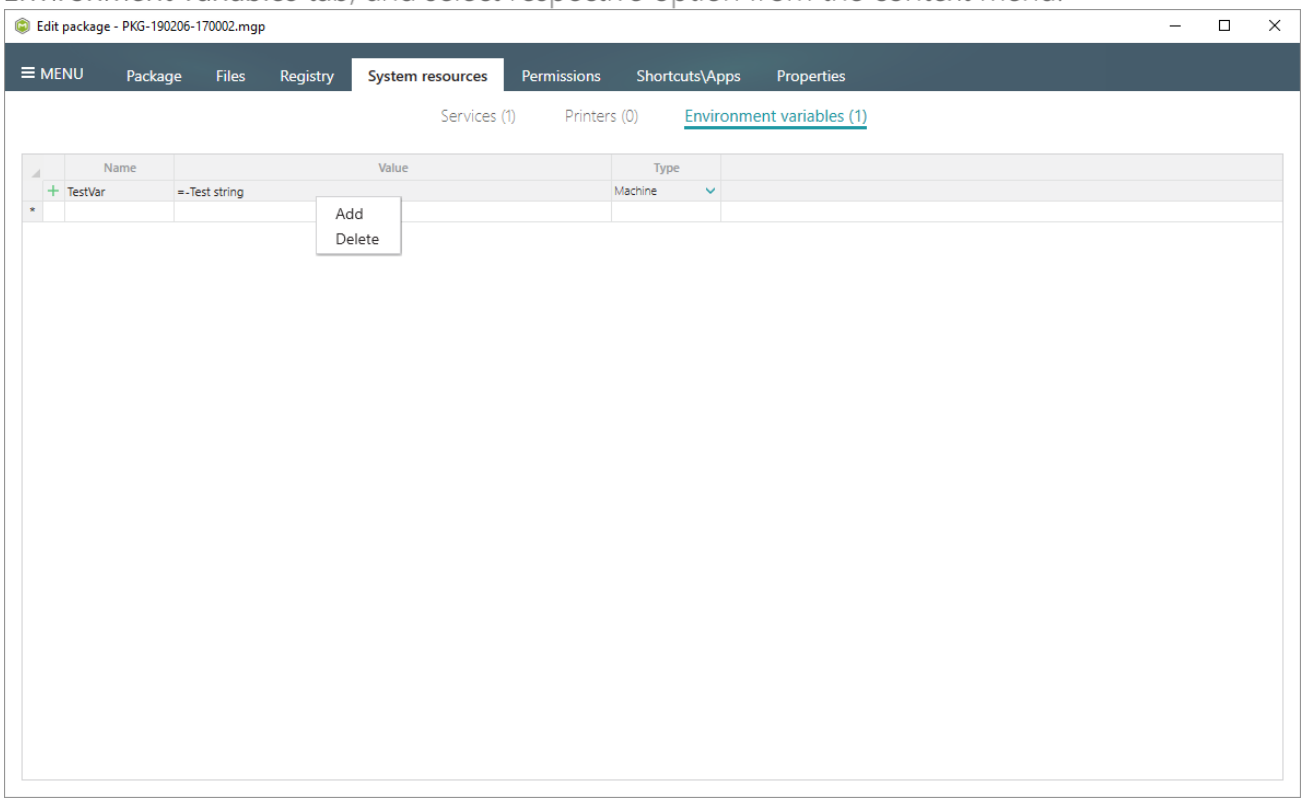


- In order to review and exclude unnecessary services, go to the **System resources** -> **Services** tab, and uncheck the checkbox, located before the service name in the list, for those services,

which you want to exclude.

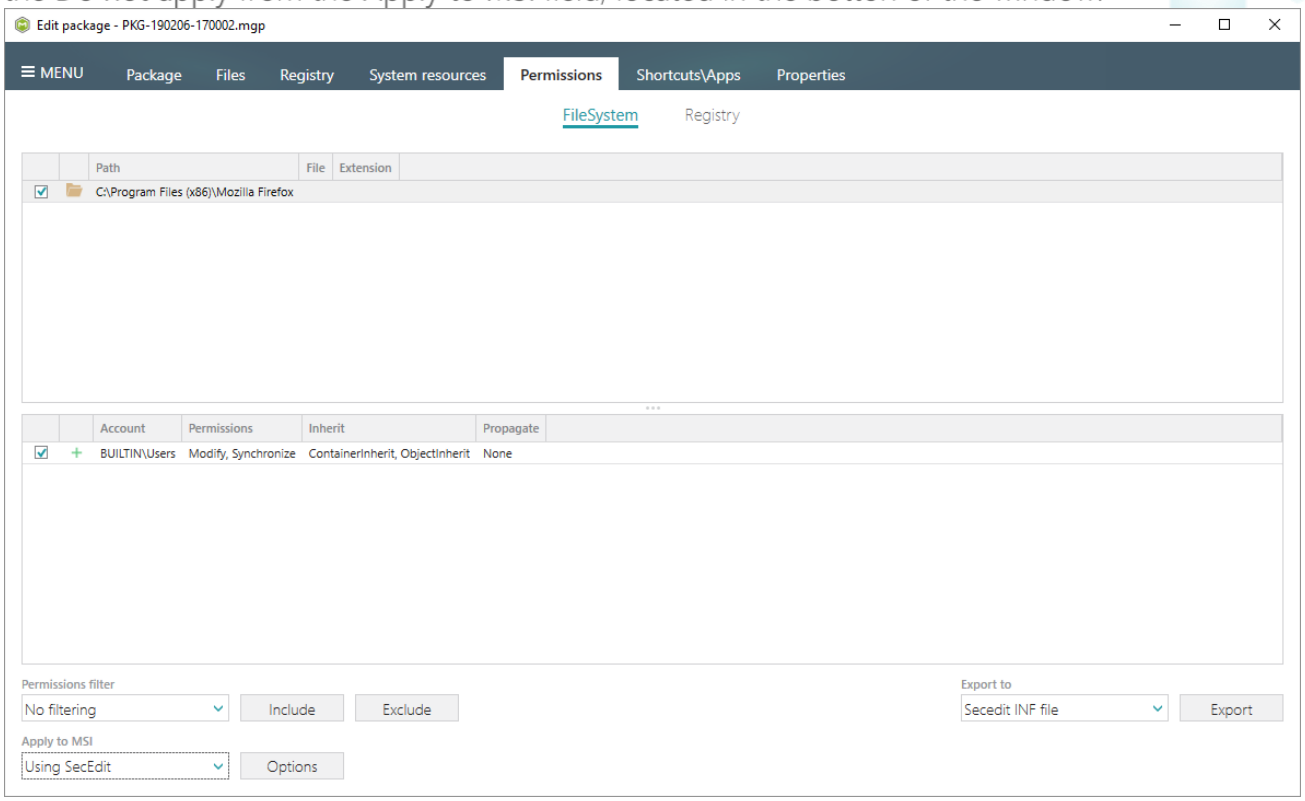


- In order to add and remove Environment variables, go to the System resources -> Environment variables tab, and select respective option from the context menu.

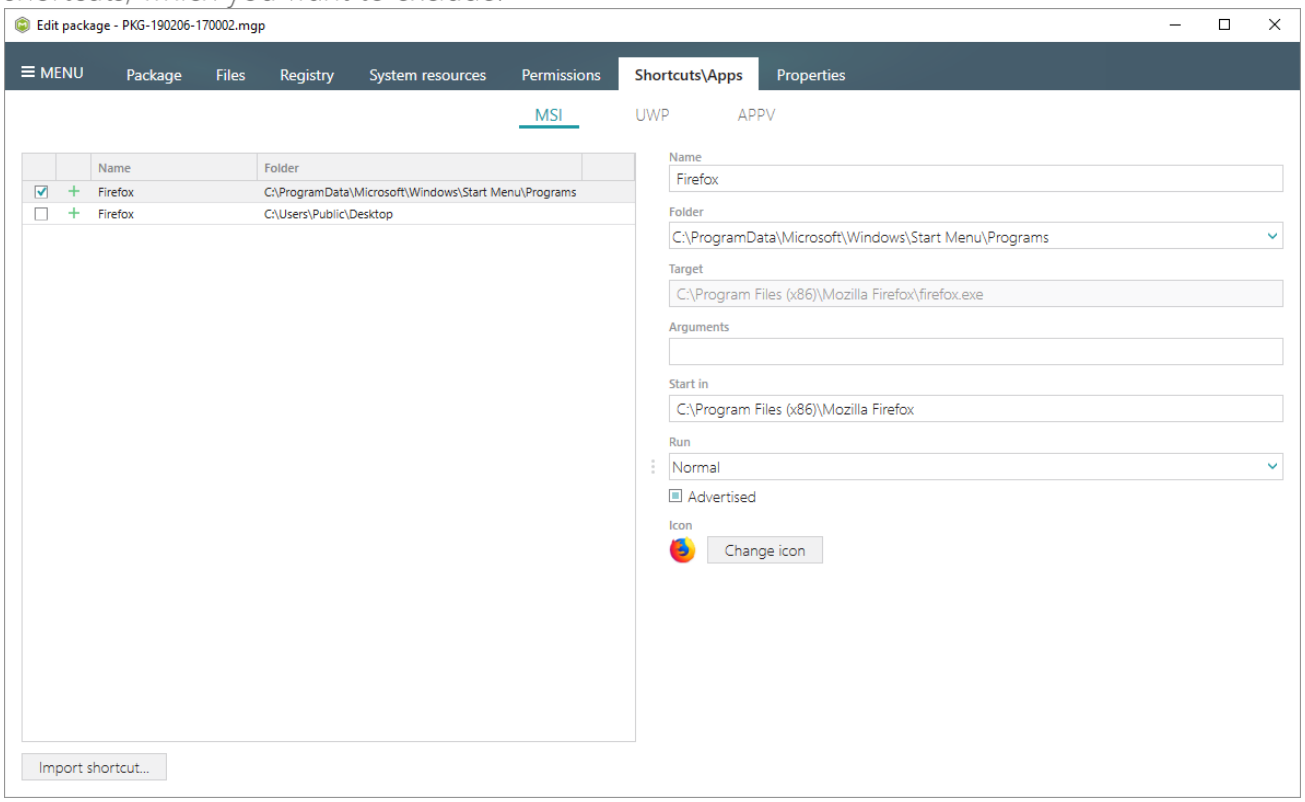


- On the Permissions tab you will find the captured permissions for the file system and registry. For excluding the unnecessary permission changes, either untick the each checkbox, located before the path, for which permission changes were detected, or to skip all permissions select

the Do not apply from the Apply to MSI field, located in the botton of the window.

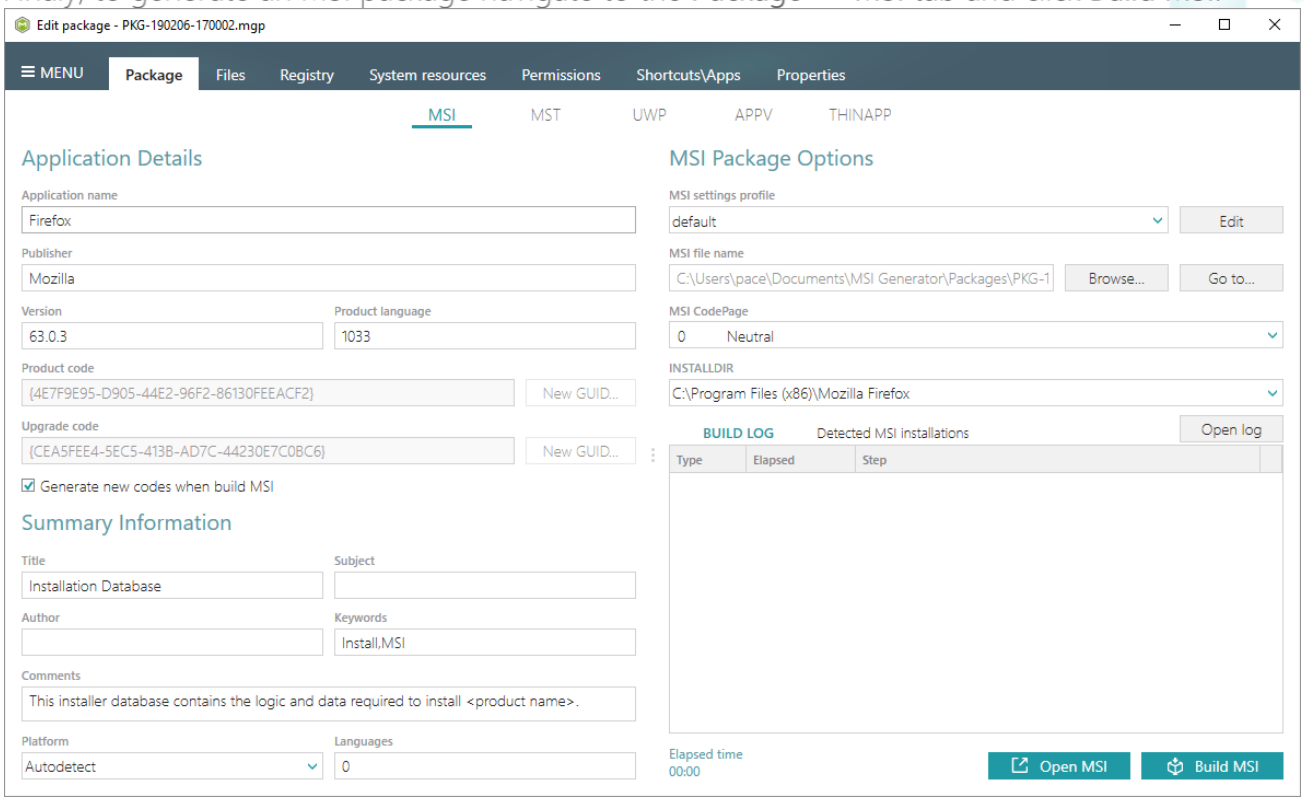


- In order to review and exclude the unnecessary MSI shortcuts, go to the Shortcuts\Apps -> MSI tab, and untick the checkbox, located before the shortcut name in the list, for those shortcuts, which you want to exclude.

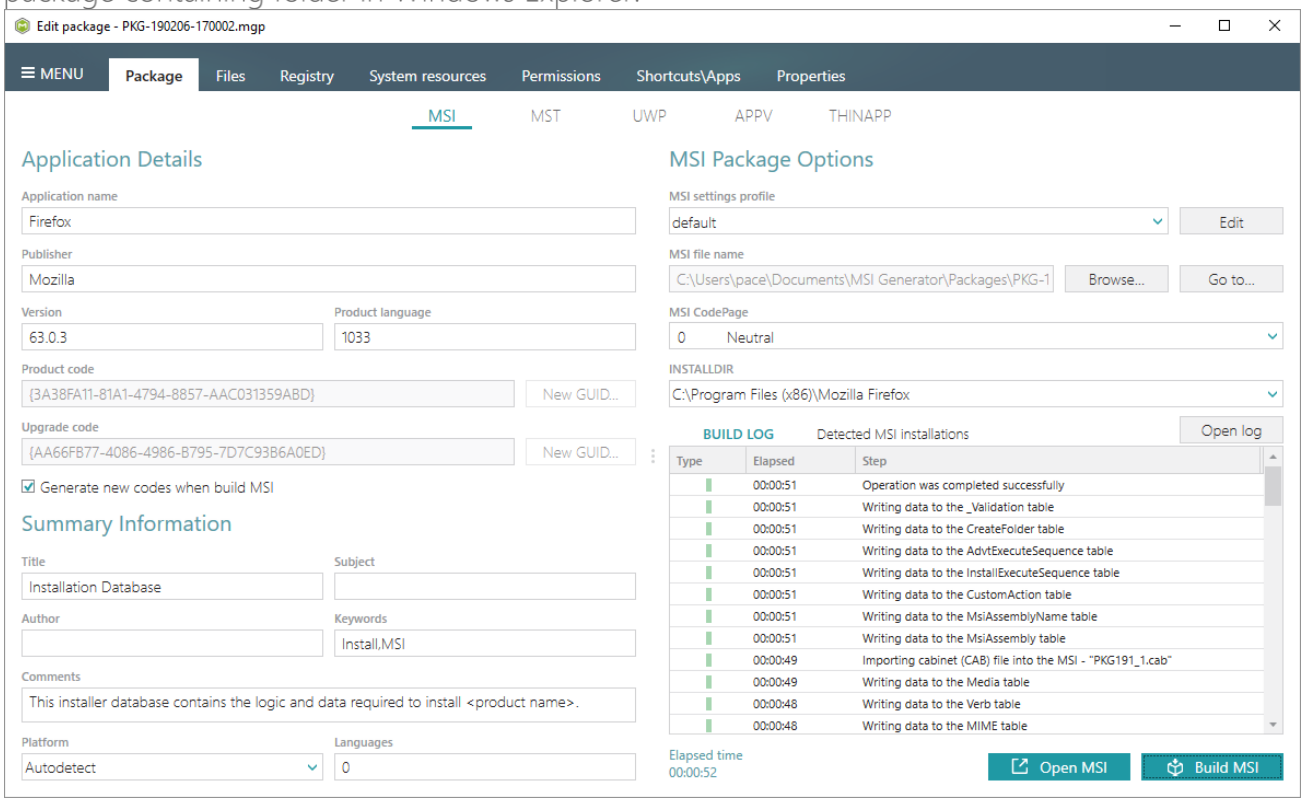


- For managing MSI Settings such as application details, compression type, use of advertised tables, adding, active setup and the like, see section 3.8.6

- Finaly, to generate an MSI package navigate to the **Package** -> **MSI** tab and click **Build MSI**.



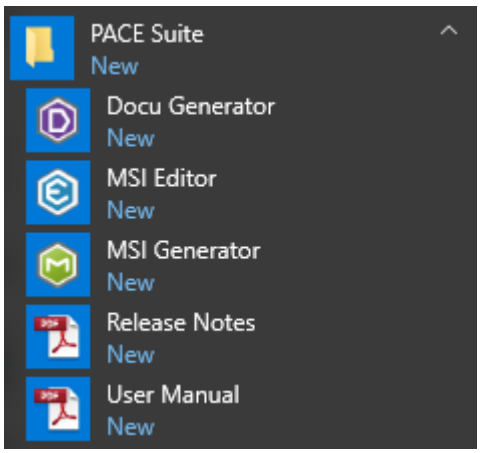
- Once the package is built, click **Go to...**, located next to the **MSI file name** field, to open the package containing folder in Windows Explorer.



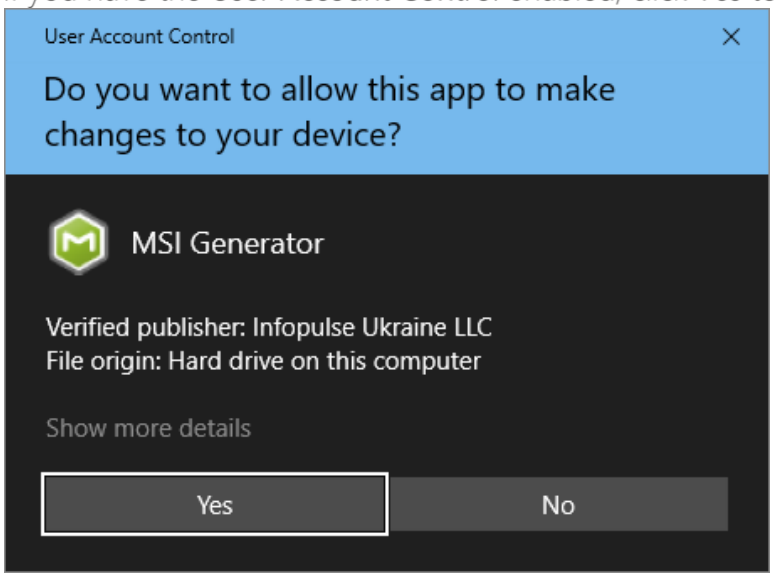
3.1.3 Capture System Changes to MSI

Capture system changes, which were made to the file system and registry, by script, by application or manually and save them into MSI package using MSI Generator. Along with files and registry, services and environment variables, you can capture the modified permissions to the file system and registry.

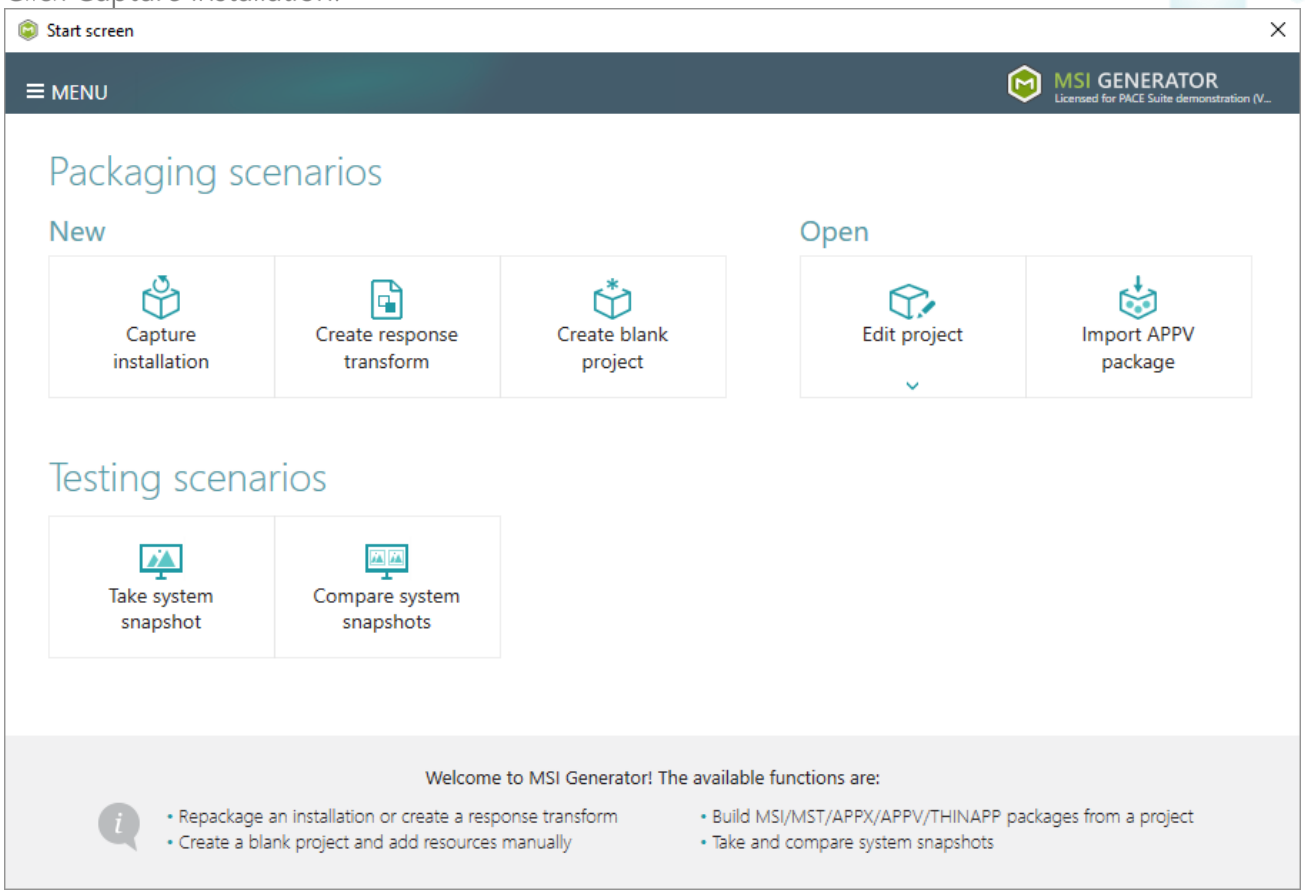
- Launch MSI Generator from the desktop of the start menu shortcut.



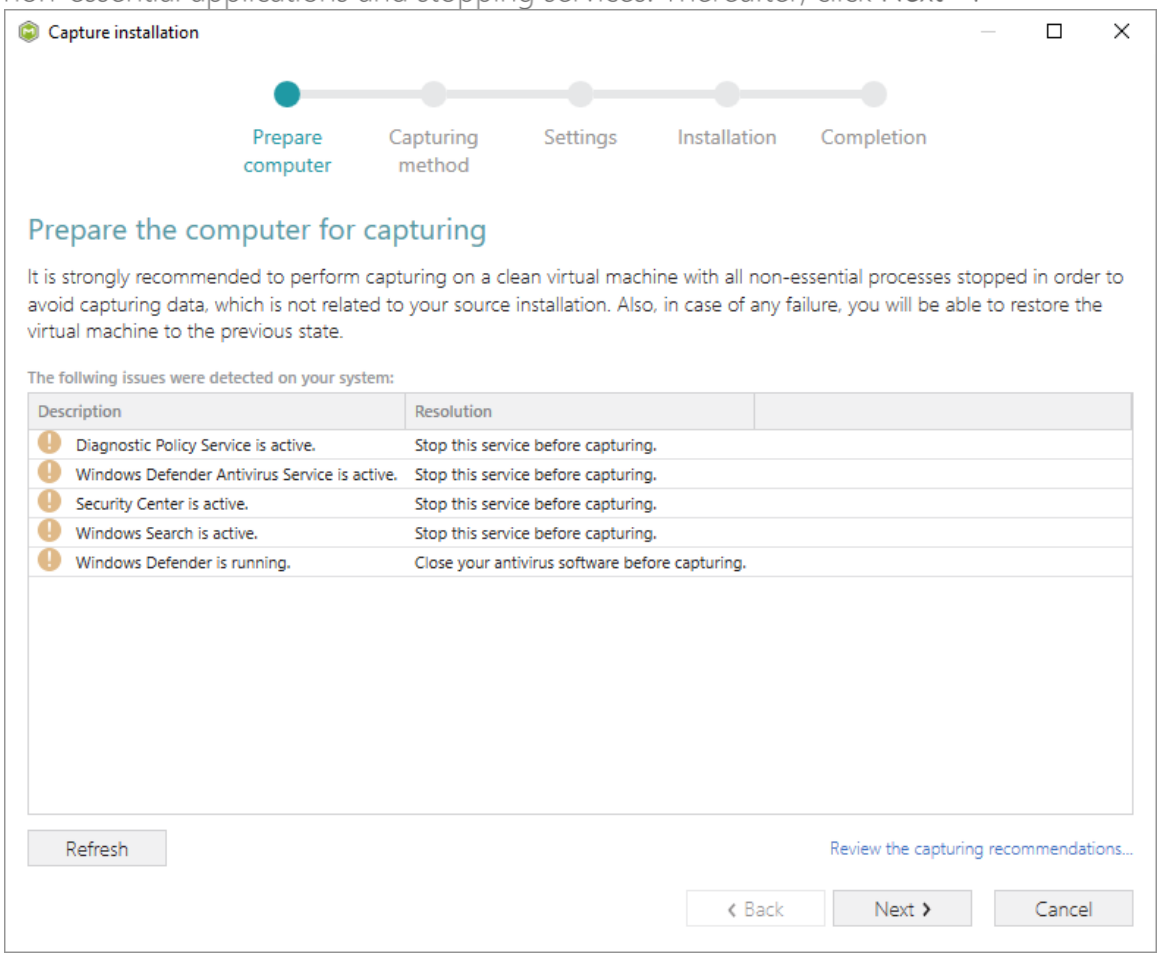
- If you have the User Account Control enabled, click Yes to launch the application.



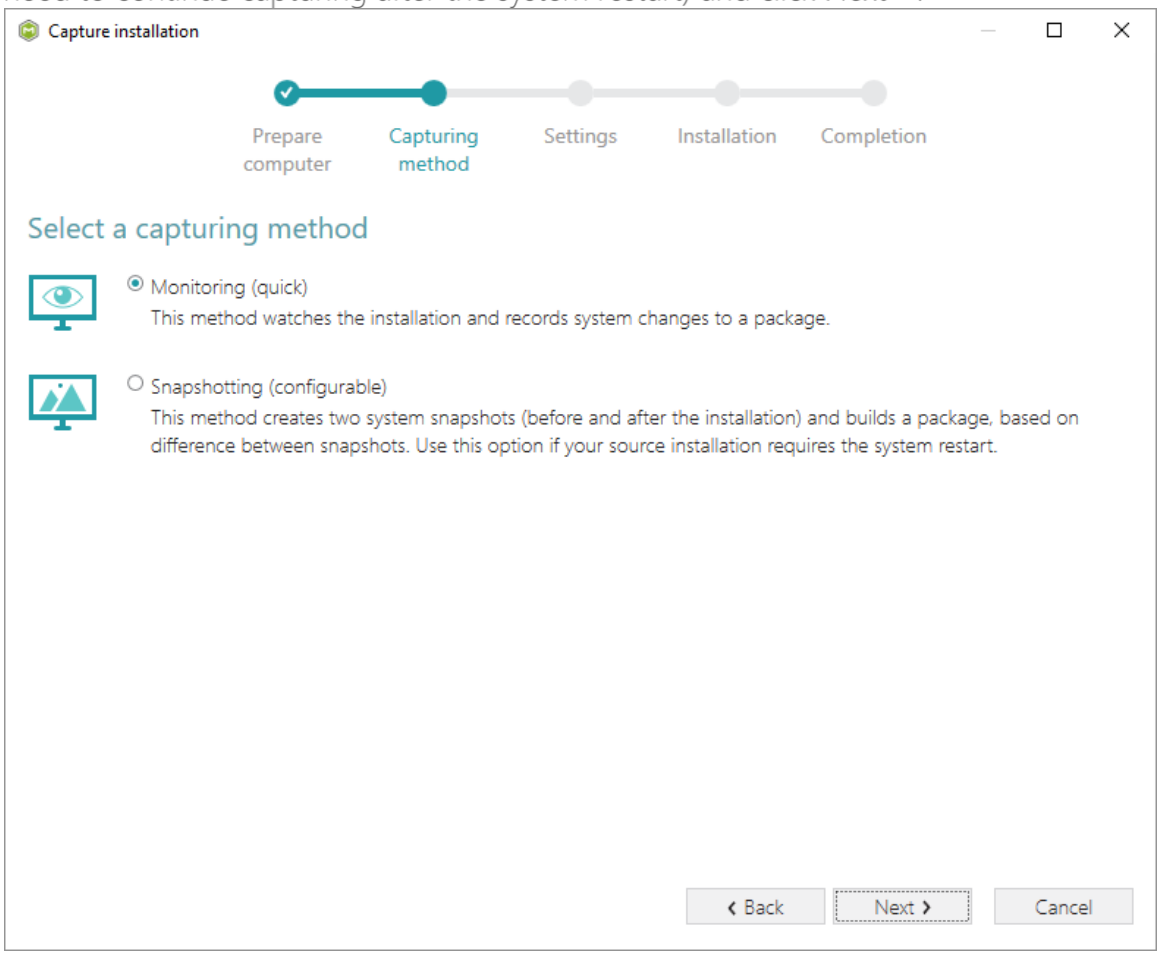
- Click Capture installation.



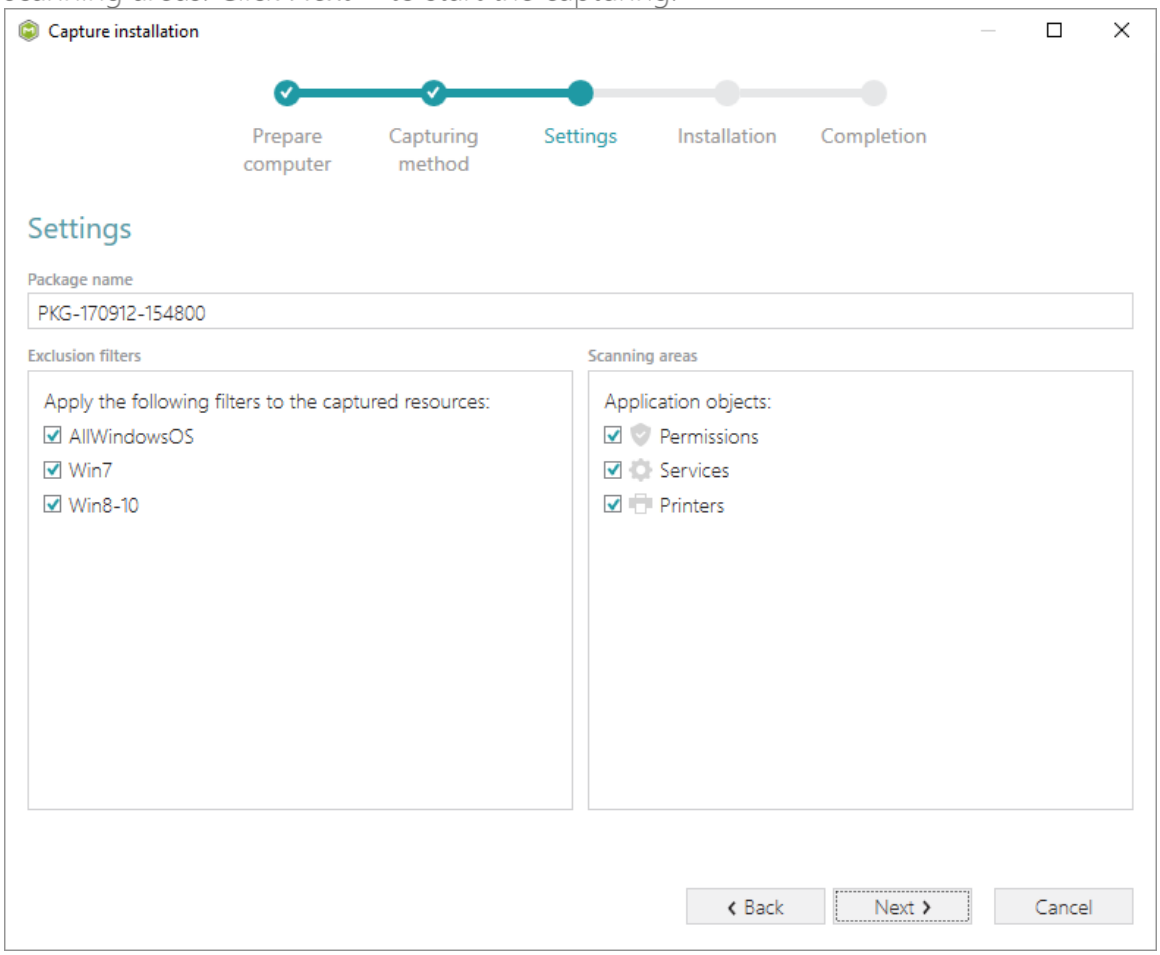
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



- Select the **Monitoring** method for the quicker capturing (or use the **Snapshotting** one if you need to continue capturing after the system restart) and click **Next >**.

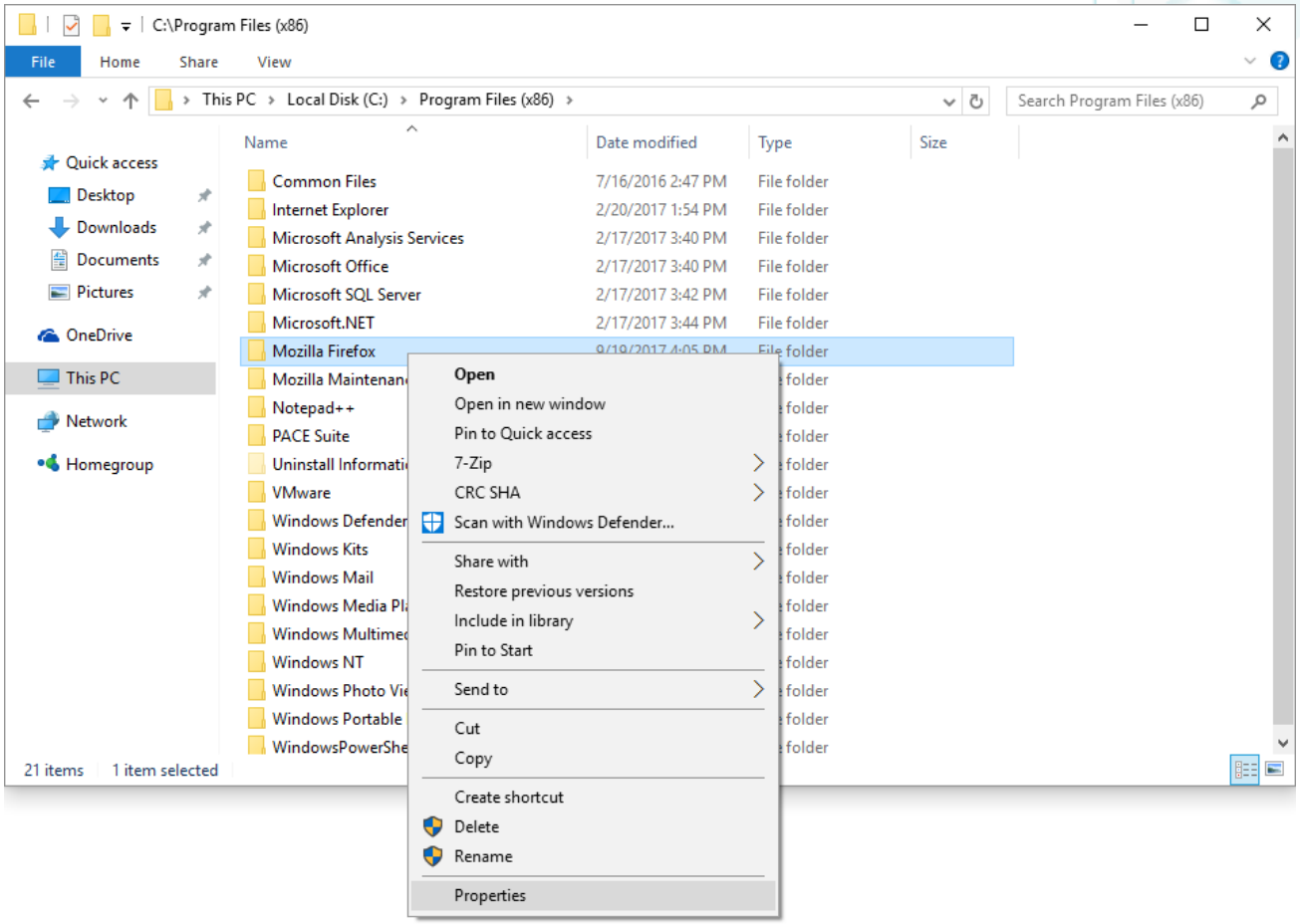


- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.

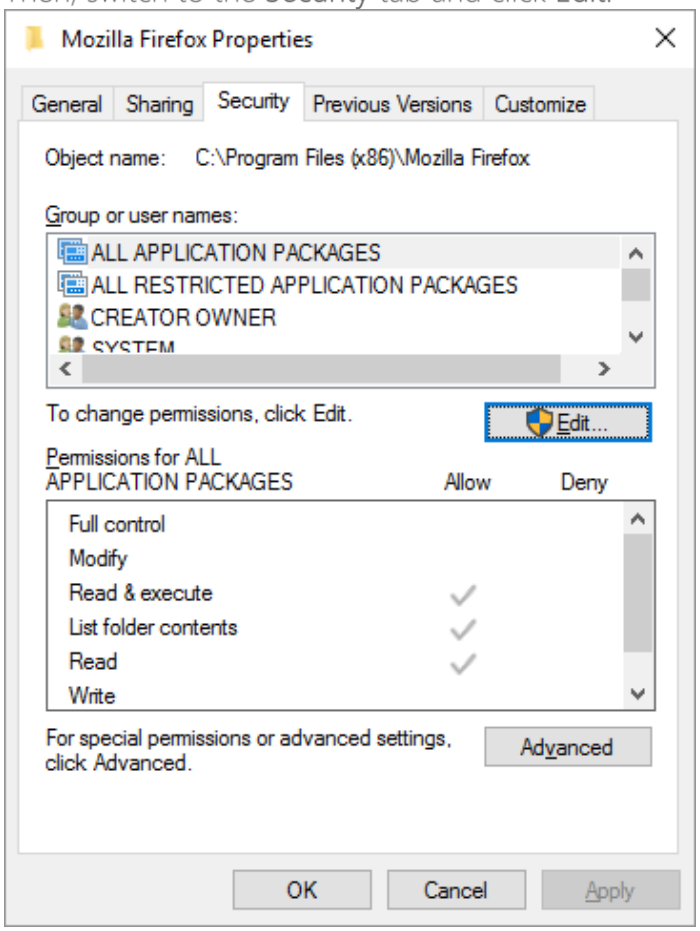


- Now you can make any changes to the file system and registry, which you want to capture and include to the package. For instance, you can copy files, import REG file to the system registry, change file, folder permissions, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.

  In case of necessity, to change the default permission settings for a folder or a file, select Properties from the Windows Explorer context menu of a folder, for which you want to set new permissions.

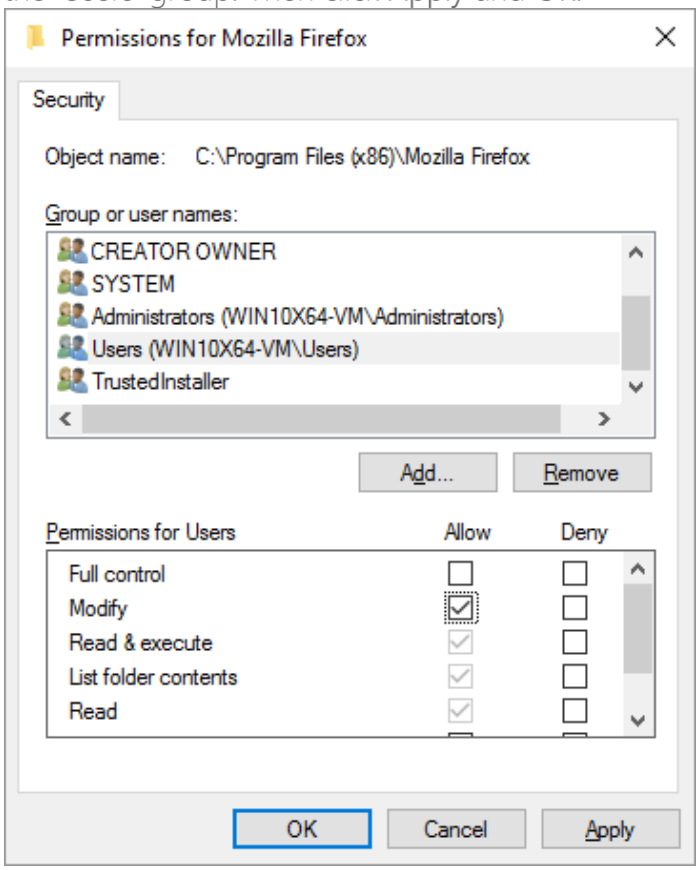


Then, switch to the Security tab and click Edit.

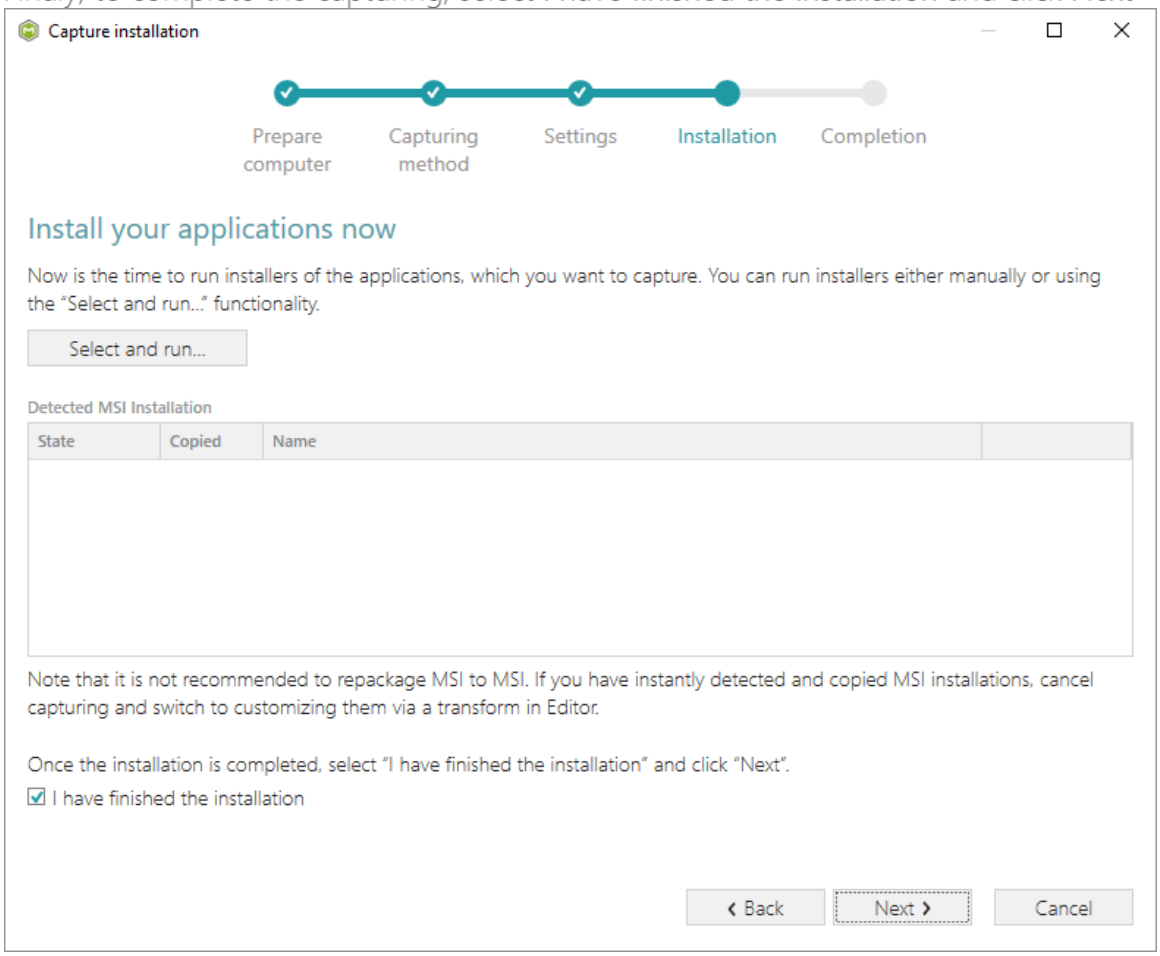


And select a group or a user, for which you want to change the permissions and then select the necessary access rights. For example, lets set the 'Write' and the 'Modify' access rights for

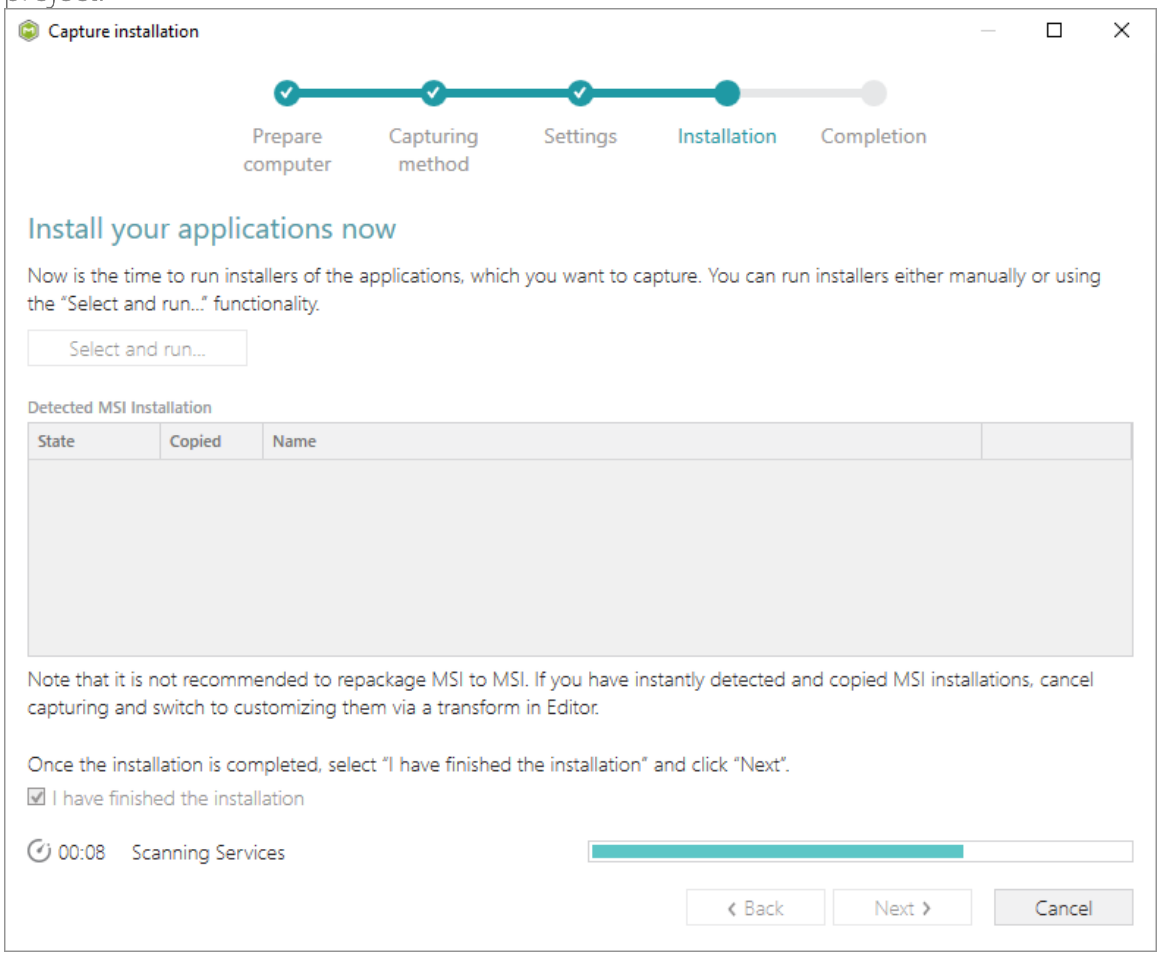
the 'Users' group. Then click Apply and OK.



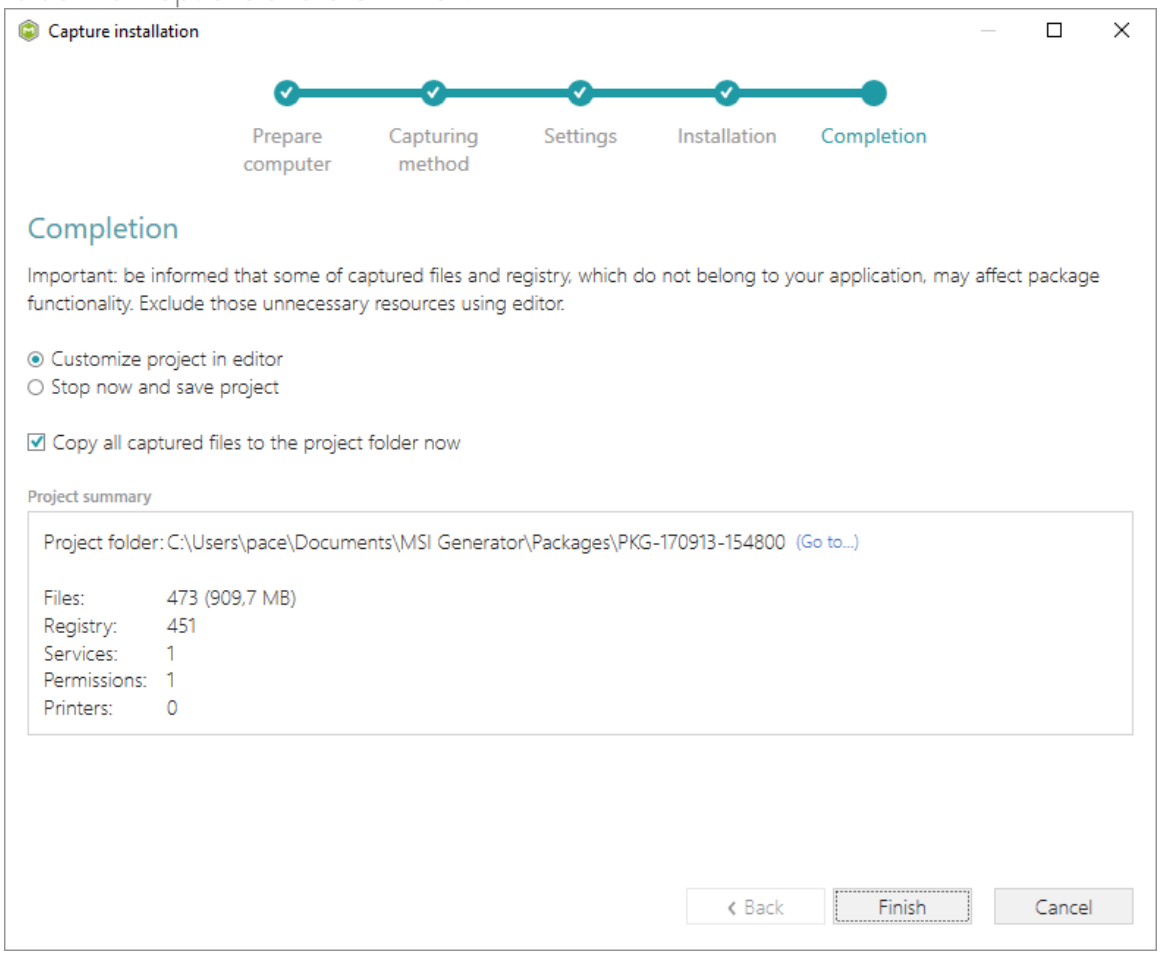
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



- Wait a little, while the capturing process is finishing, filtering captured data and creating the project.

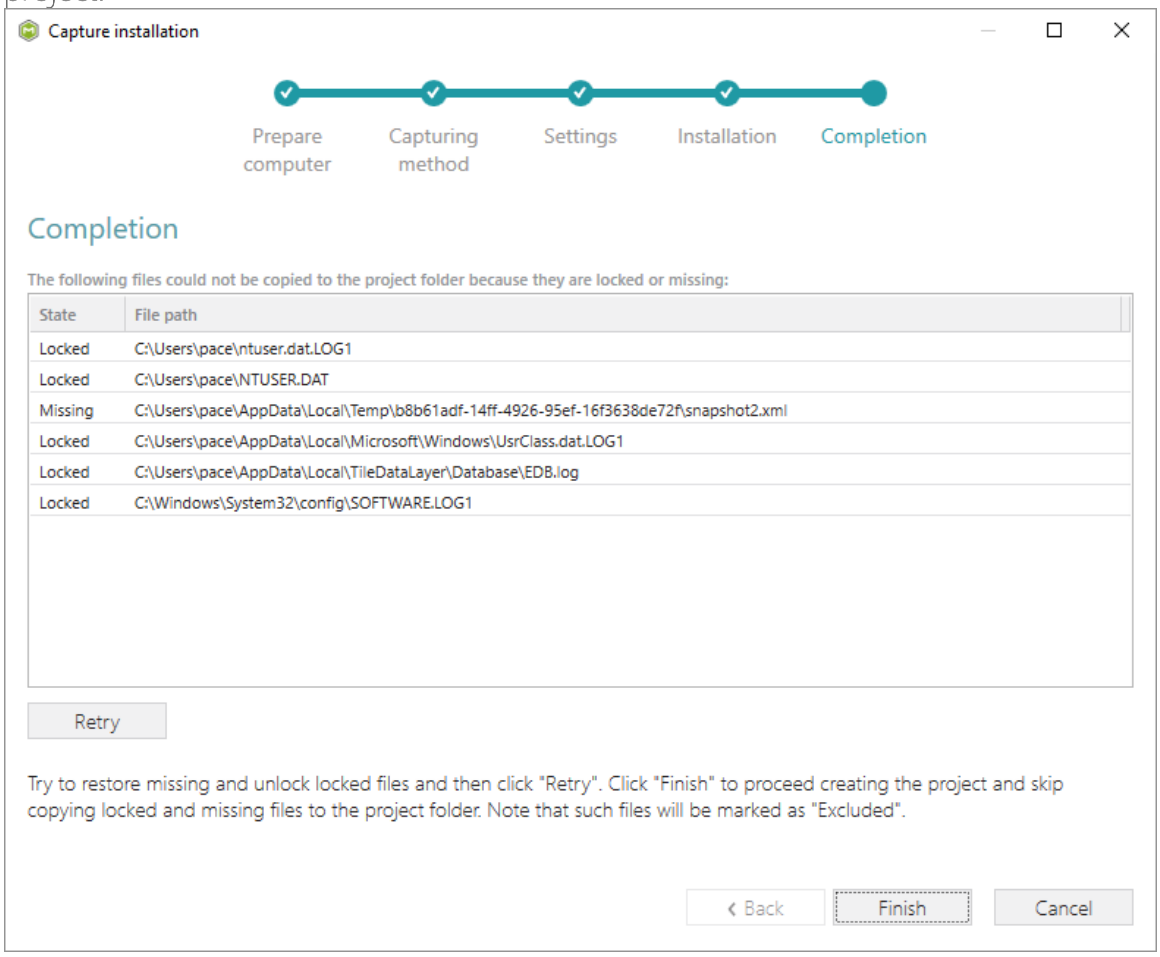


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



- If some of captured files are locked or missing and as result they could not be copied to the project folder, they will be displayed as illustrated below. Usually, the displayed files are system noise and do not refer to the captured application. Click Finish to continue with skiping locked and missing files. Note that such files will be automatically marked as 'Excluded' in your

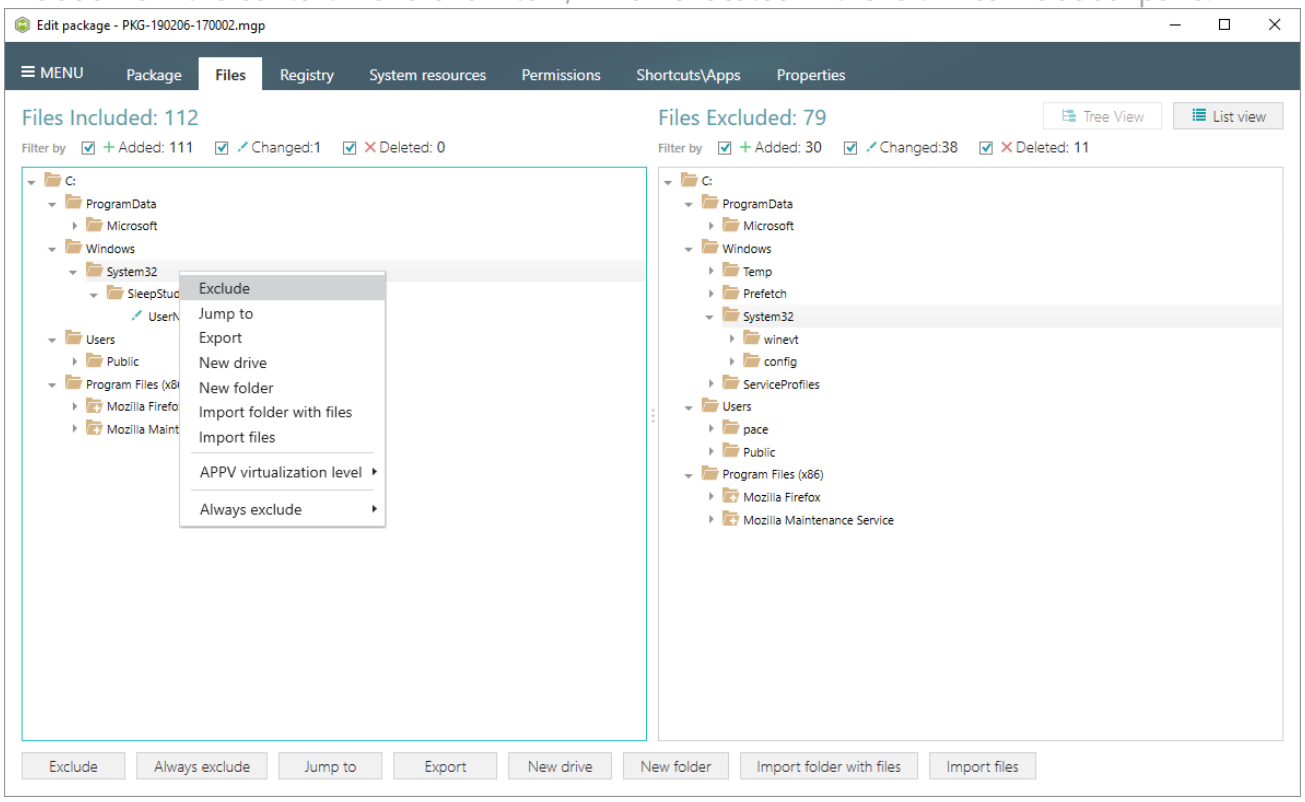
project.



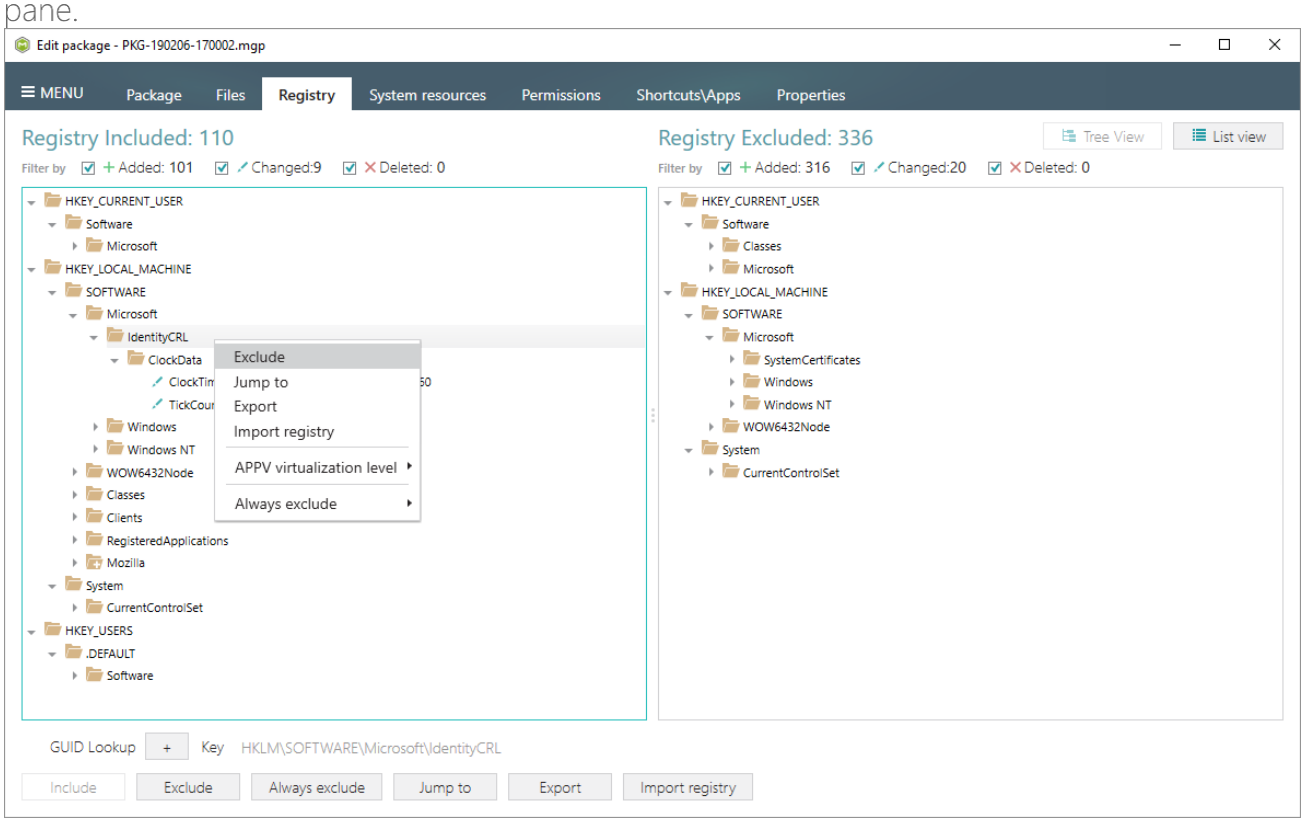
- Once the project is opened in the project editor, it is strongly recommended to review the Files, Registry, System resources, Permissions, and Shortcuts\Apps tabs and exclude the accidentally captured resources, which do not belong to the application. Such accidentally captured resources are also called system noise and it could be Windows Updates, Search indexing, Antivirus software and so on.

  An MSI package with such resources may fail during the installation. The common reason for such fails is trying to write a registry value to the system-protected location, strongly prohibited from writing by third-party installations.

- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.

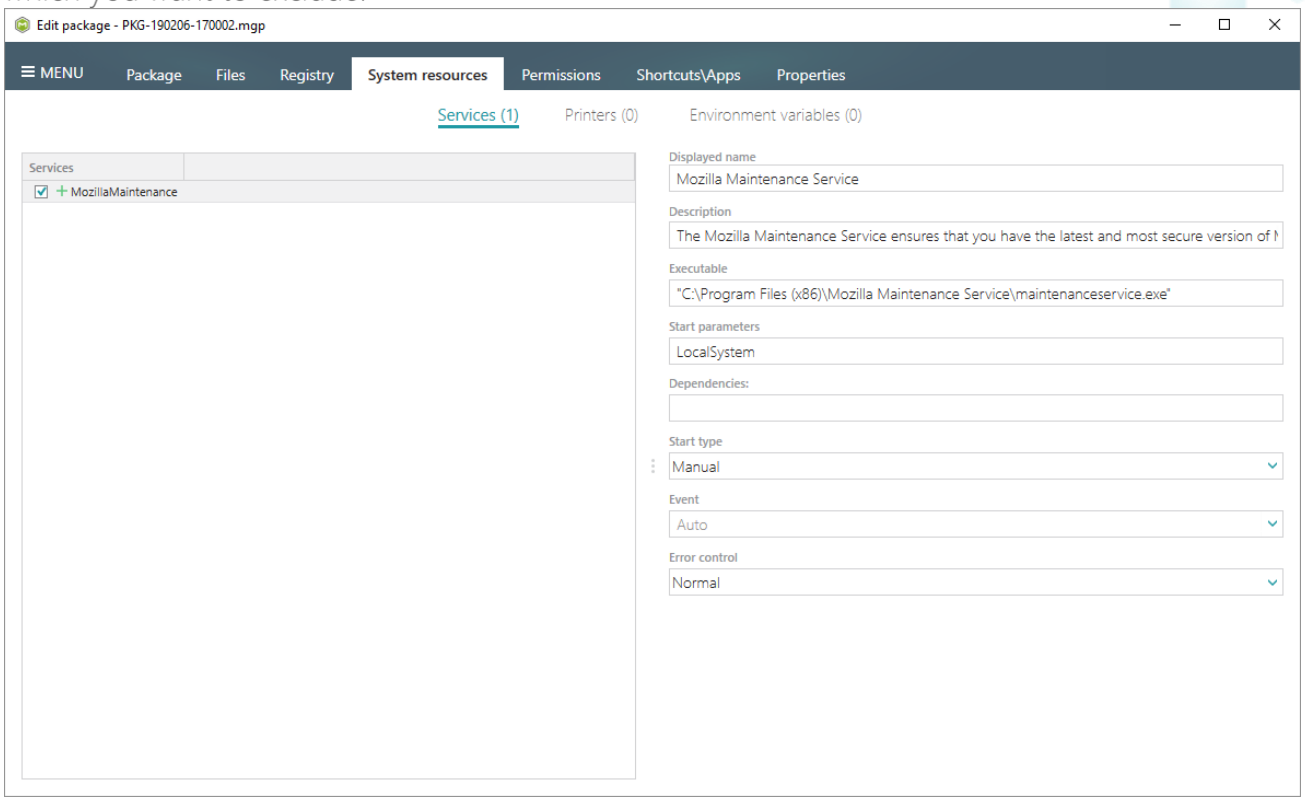


- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.

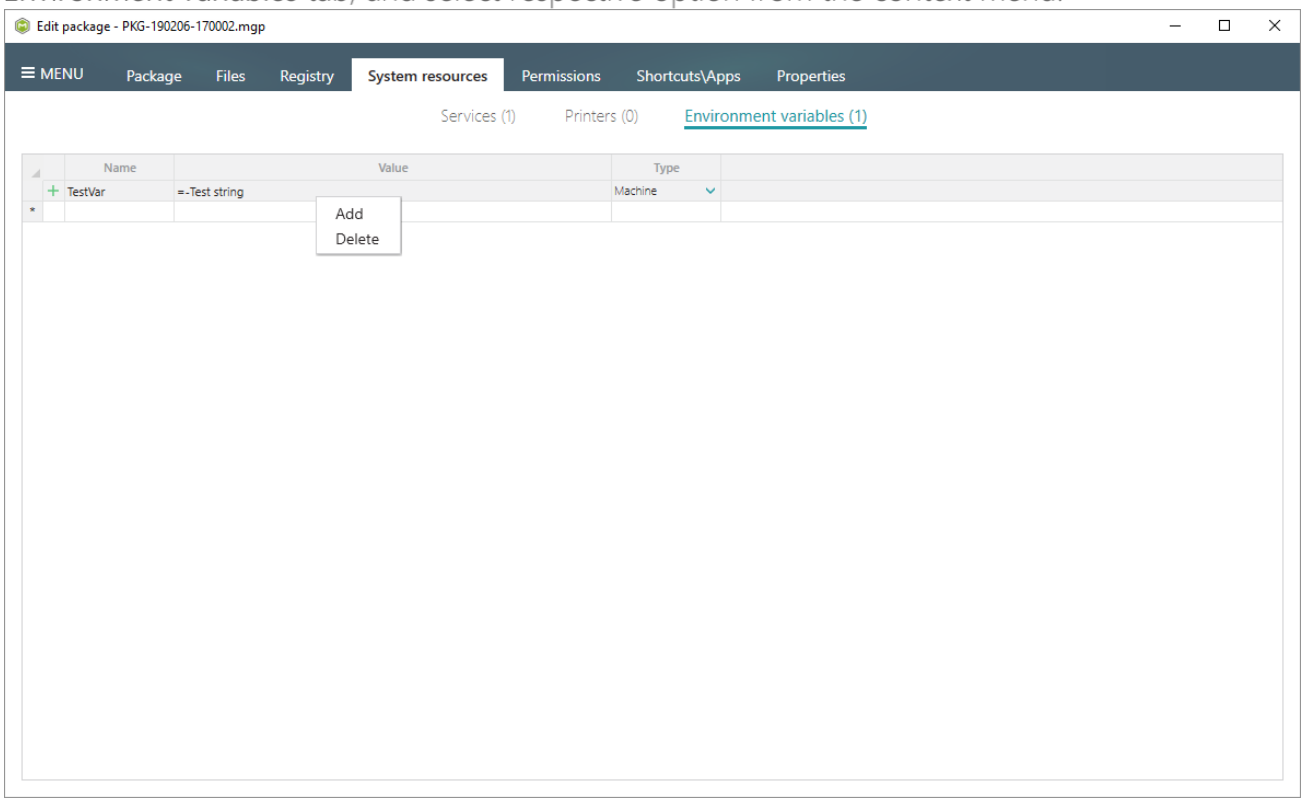


- In order to review and exclude unnecessary services, go to the **System resources** -> **Services** tab, and uncheck the checkbox, located before the service name in the list, for those services,

which you want to exclude.

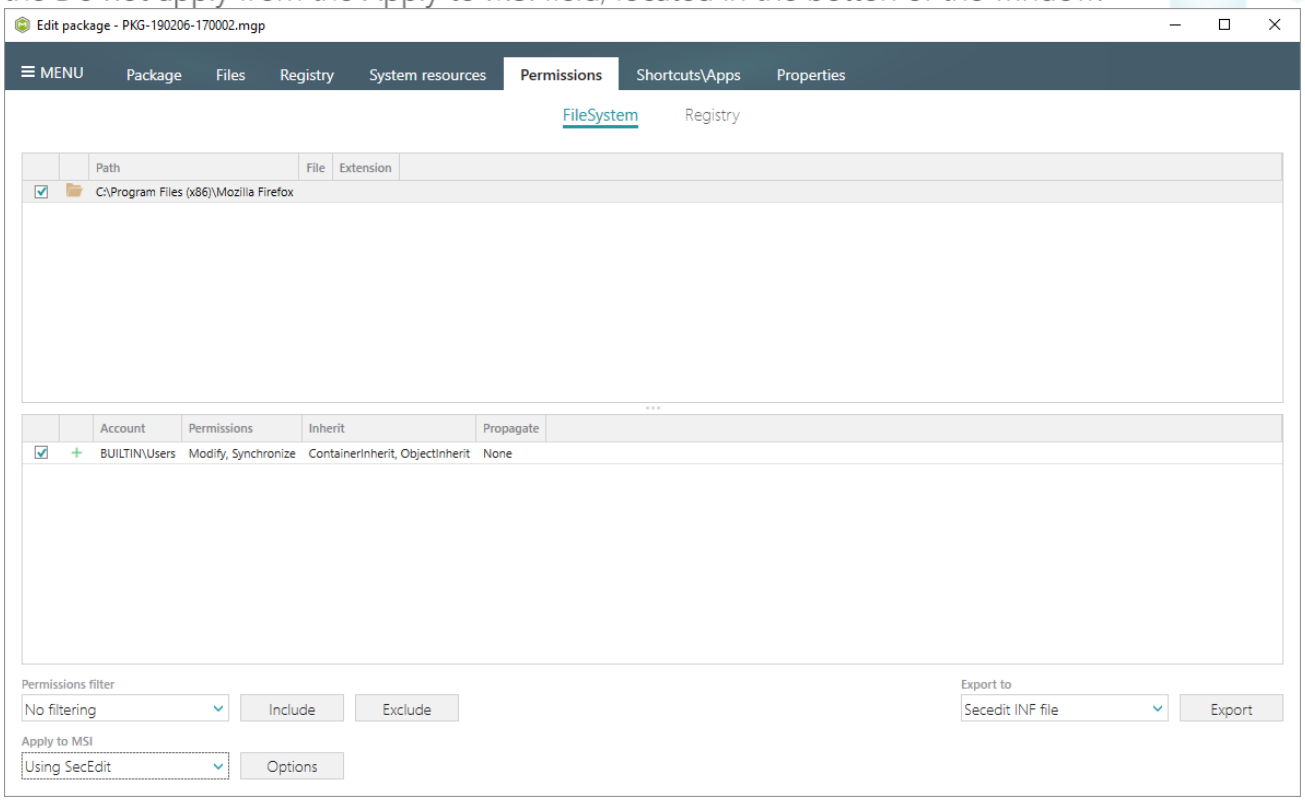


- In order to add and remove Environment variables, go to the System resources -> Environment variables tab, and select respective option from the context menu.

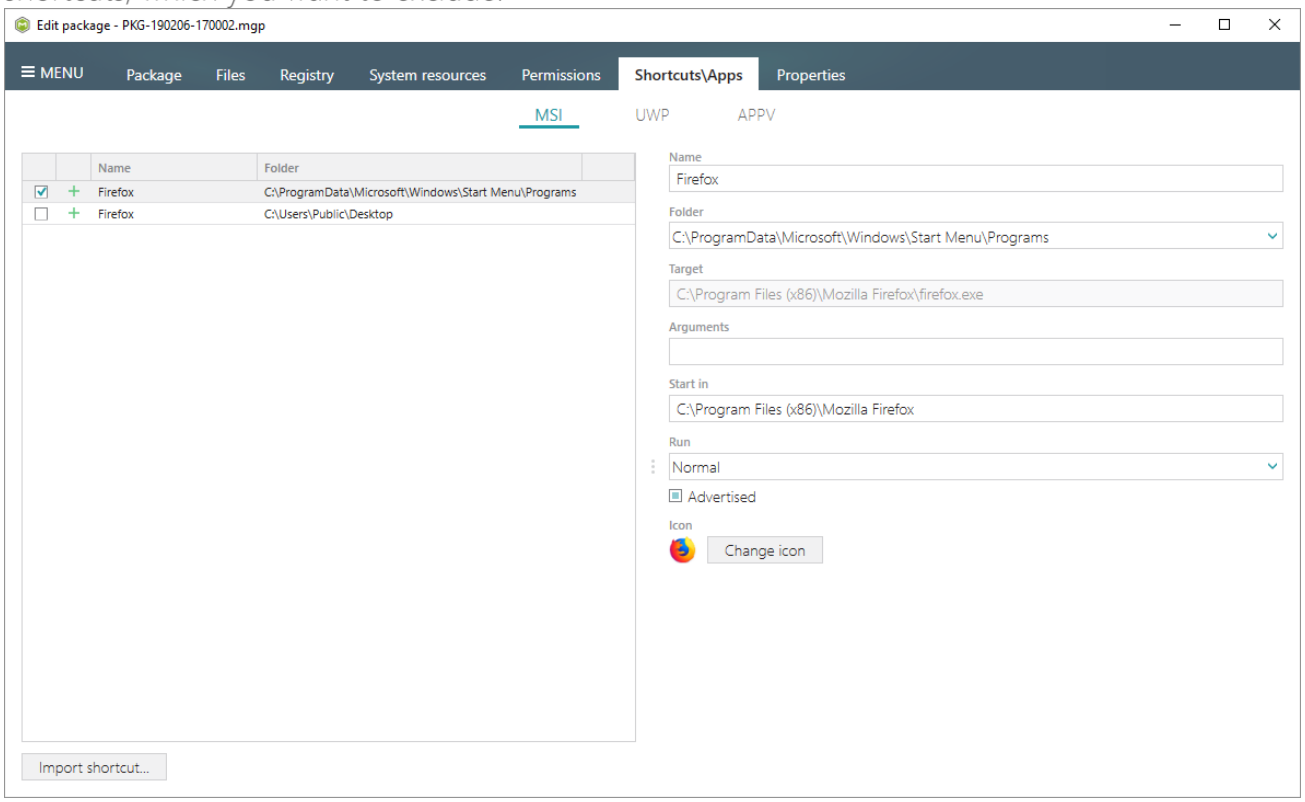


- On the Permissions tab you will find the captured permissions for the file system and registry. For excluding the unnecessary permission changes, either untick the each checkbox, located before the path, for which permission changes were detected, or to skip all permissions select

the Do not apply from the Apply to MSI field, located in the botton of the window.

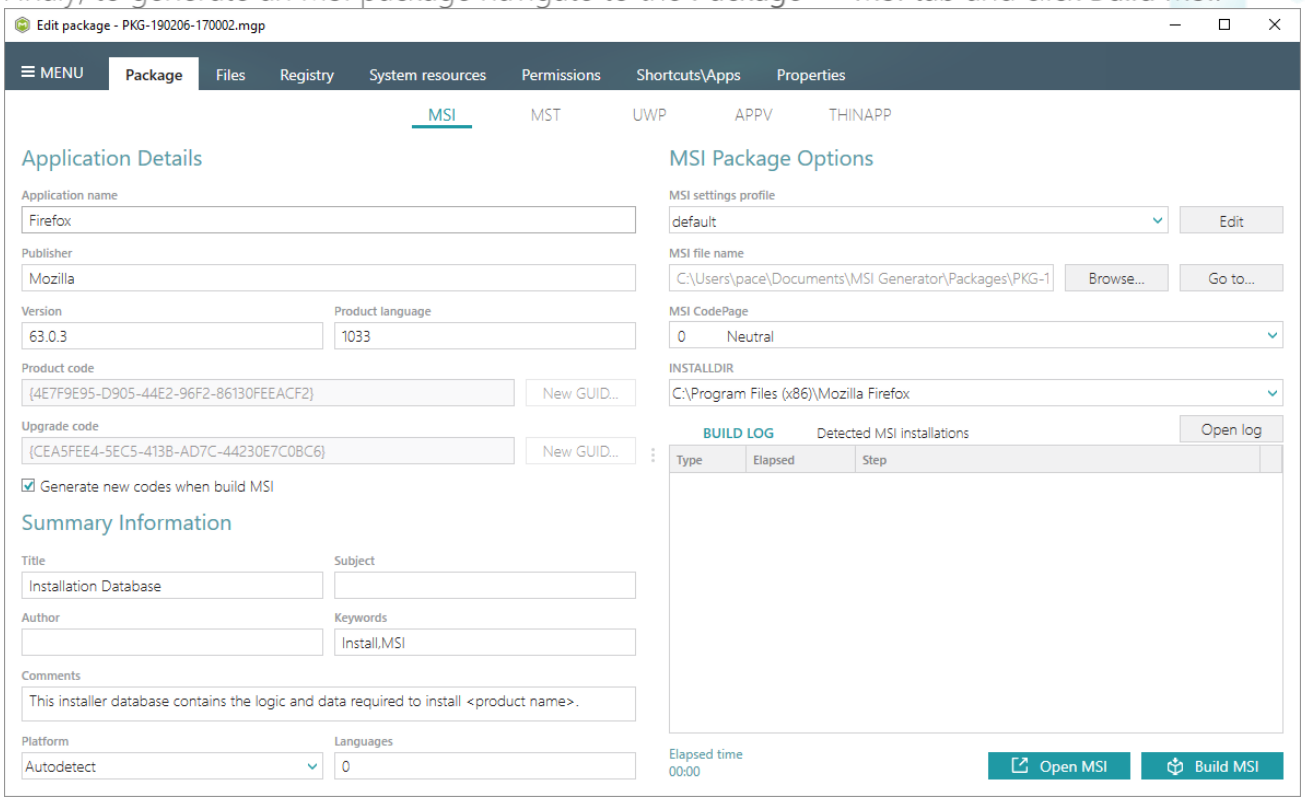


- In order to review and exclude the unnecessary MSI shortcuts, go to the Shortcuts\Apps -> MSI tab, and untick the checkbox, located before the shortcut name in the list, for those shortcuts, which you want to exclude.

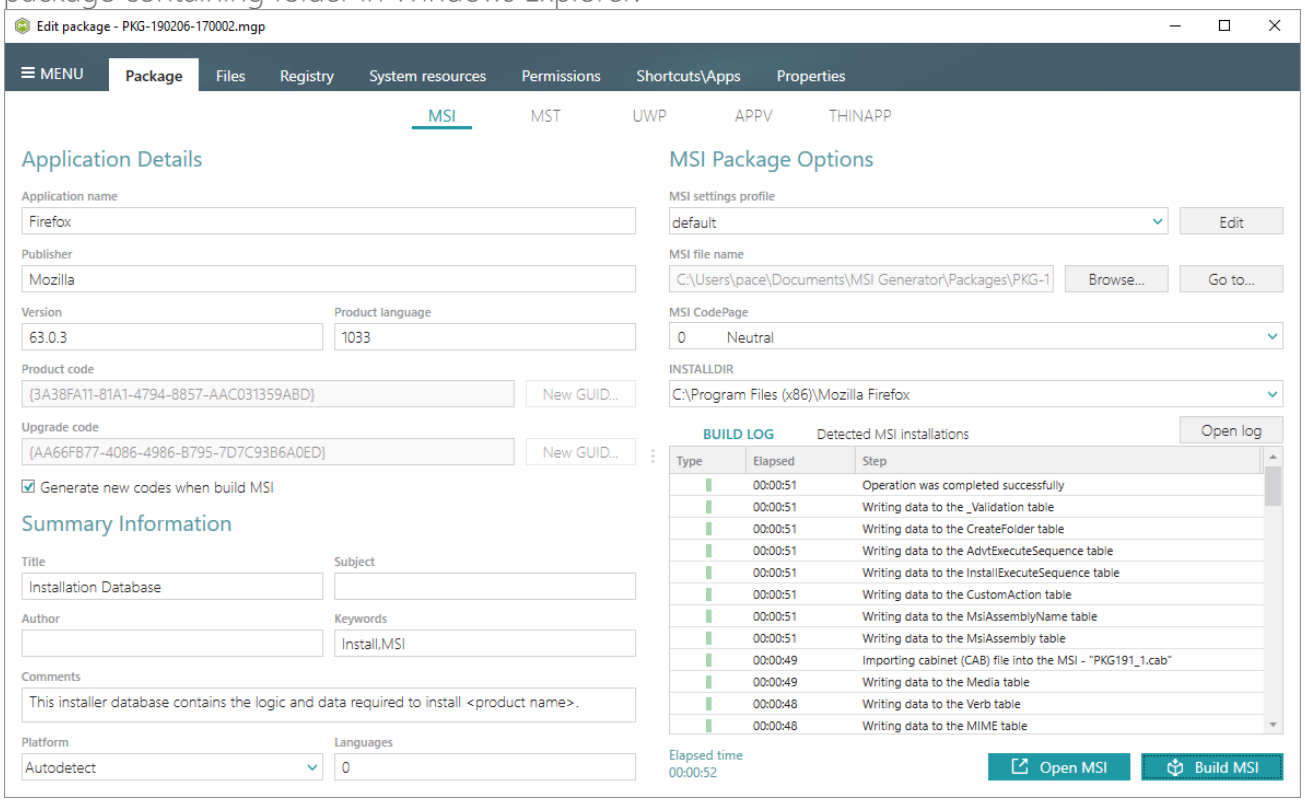


- For managing MSI Settings such as application details, compression type, use of advertised tables, adding, active setup and the like, see section 3.8.6

- Finaly, to generate an MSI package navigate to the **Package** -> **MSI** tab and click **Build MSI**.



- Once the package is built, click **Go to...**, located next to the **MSI file name** field, to open the package containing folder in Windows Explorer.



3.2 Create MST

Choose a scenario that better suits your needs:

- Create Blank MST, described in section 3.2.1
  Create a blank transform file (MST) for your MSI package in MSI Editor. All changes, made to the opened MSI database, will be saved to the MST.
- Create Response MST, described in section 3.2.2
  Capture all your inputs and changes, made to the Windows Installer UI dialogs of the MSI installation, and save them to the MST file using MSI Generator. The MSI installation will be simulated, which allows capturing all necessary changes very promptly and without affecting your real system.
- Capture System Changes to MST, described in section 3.2.3
  Capture system changes, which were made to the file system and registry, by script, by application or manually and save them into MST file, generated against the selected MSI package using MSI Generator. Along with files and registry, services and environment variables, you can capture the modified permissions to the file system and registry.

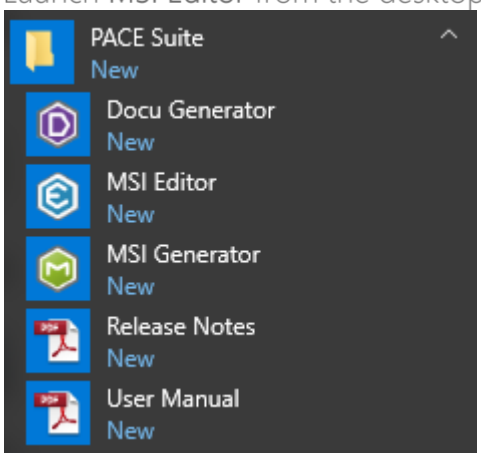
3.2.1 Create Blank MST

Create a blank transform file (MST) for your MSI package in MSI Editor. All changes, made to the opened MSI database, will be saved to the MST.

NOTE PACE Suite supports two scenarious of saving changes to MST:
1. Open MSI -> create and apply MST -> modify the package -> save changes
2. Open MSI -> modify the package -> save changes as MST

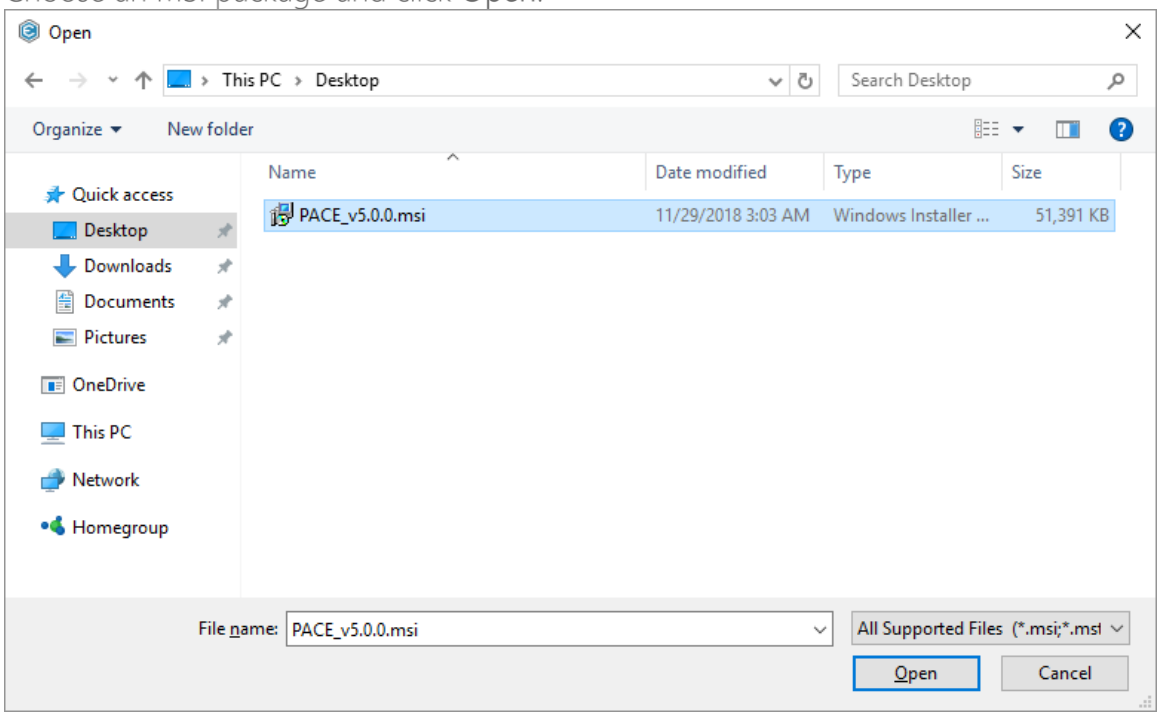
The following describes the 1st scenario "Open MSI -> create and apply MST -> modify the package -> save changes".

- Launch MSI Editor from the desktop or the start menu shortcut.



- Select Open... from the MENU to open the MSI package, which you want to modify via MST.

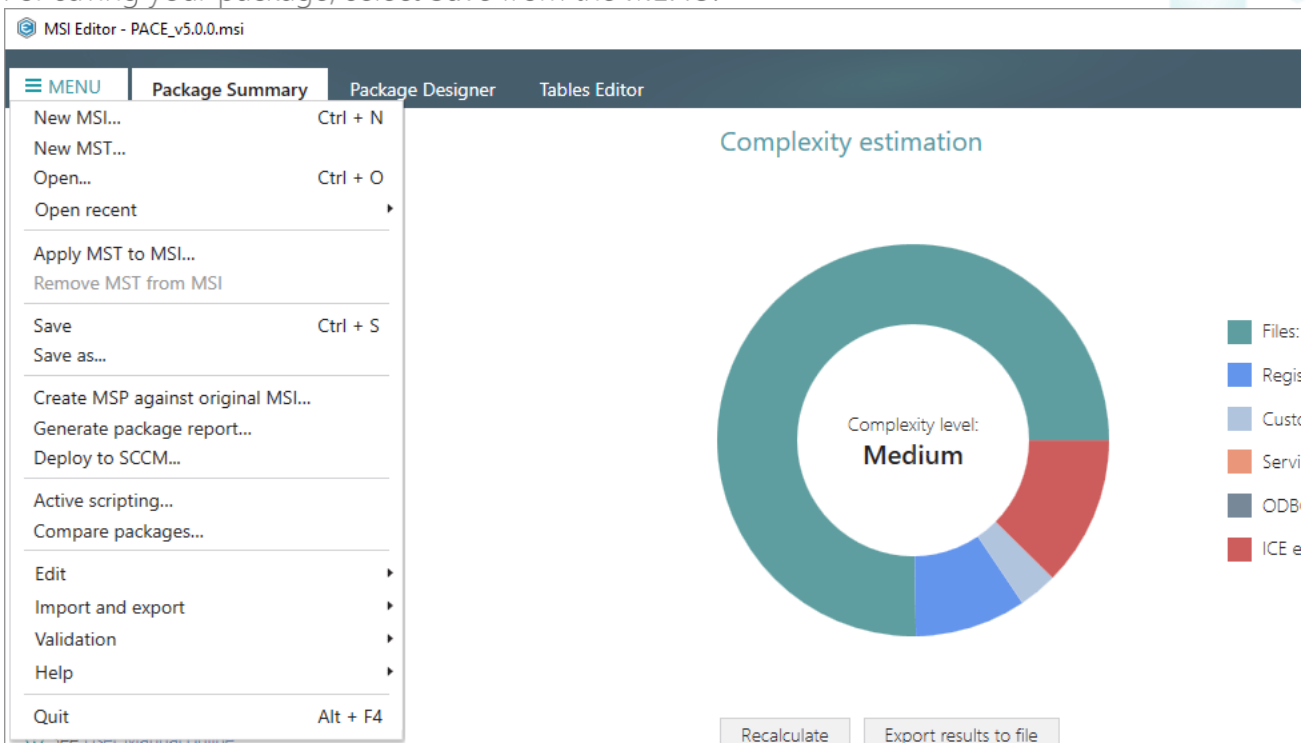
- Choose an MSI package and click Open.



- For managing package resources and settings, refer to the respective section of this document:
    - For managing Application Details see section 3.7.2
      Manage product name, version, language and manufacturer of the application.
    - For managing Package IDs see section 3.7.3
      Manage product, upgrade and package codes.

- For managing Files and Folders see section 3.7.4
  Review package files and folders, add new folders, import files, set permissions, register COM objects and .NET assemblies, set the INSTALLDIR folder and the like.
- For managing Shortcuts see section 3.7.5
  Review package shortcuts, add new or import existing shortcuts, manage their properties.
- For managing Registry see section 3.7.6
  Review package registry, add new or import from REG file registry keys and values, integrate ActiveSetup, set permissions and the like.
- For managing Custom Actions see section 3.7.7
  Manage custom and standard actions of the package, add new one to run scripts, EXE, etc.
- For managing Drivers see section 3.7.8
  Review the package device drivers, modify and add new ones.
- For managing Features and Components see section 3.7.9
  Review the feature-component structure and resources, linked with components. Add new features, components and manage their properties.
- For managing ARP Appearance see section 3.7.10
  Control how the package will be displayed in the App list (Add & Remove Programs).
- For managing Upgrades see section 3.7.11
  Manage the package upgrade settings to automatically upgrade previous MSI packages.
- For managing MSI Properties see section 3.7.12
  Manage package properties, defined in the Property table and those, which will be updated by the CustomAction, AppSearch, Control, ComboBox, CheckBox, RadioButton tables.
- For managing Summary Information see section 3.7.13
  Manage Summary Information of the opened MSI/MST.
- For managing MSI Tables see section 3.7.14
  Add, import, drop MSI tables.
- For managing Merge Modules, described in section 3.7.15
  Create new MSM or importing existing MSM to the package.

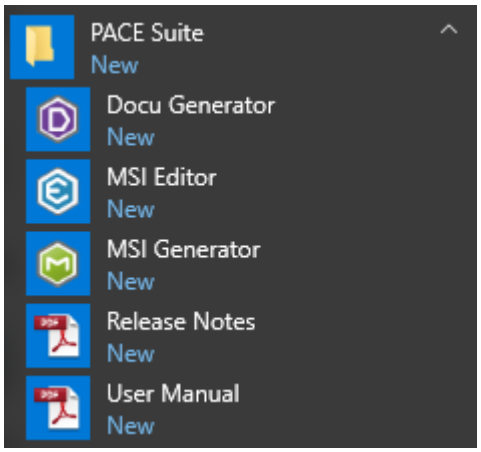
- For saving your package, select Save from the MENU.



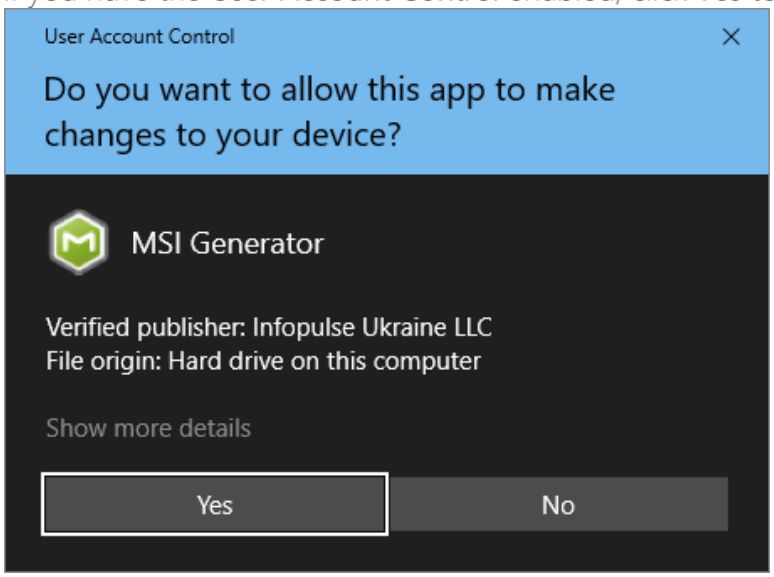
3.2.2 Create Response MST

Capture all your inputs and changes, made to the Windows Installer UI dialogs of the MSI installation, and save them to the MST file using MSI Generator. The MSI installation will be simulated, which allows capturing all necessary changes very promptly and without affecting your real system.

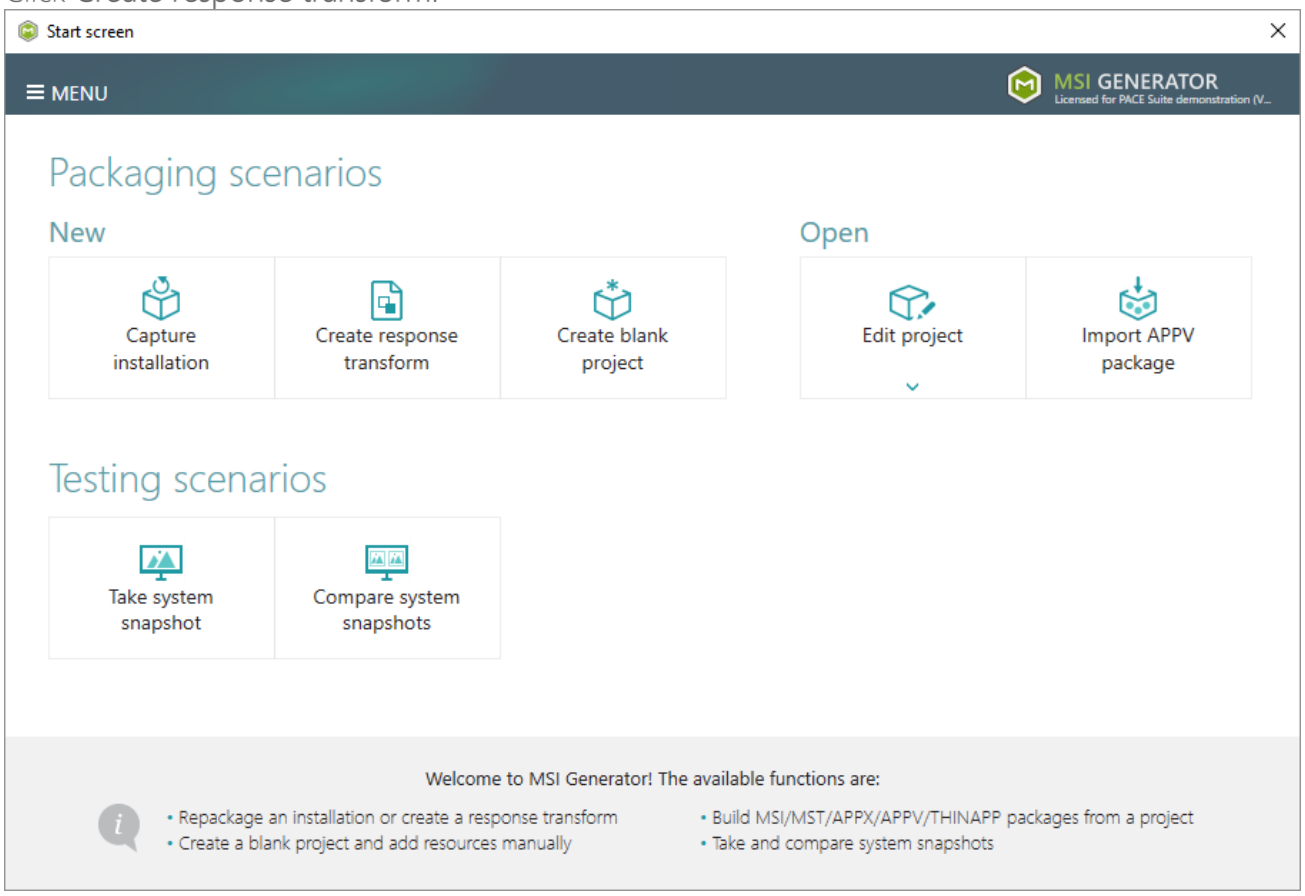
- Launch MSI Generator from the desktop of the start menu shortcut.



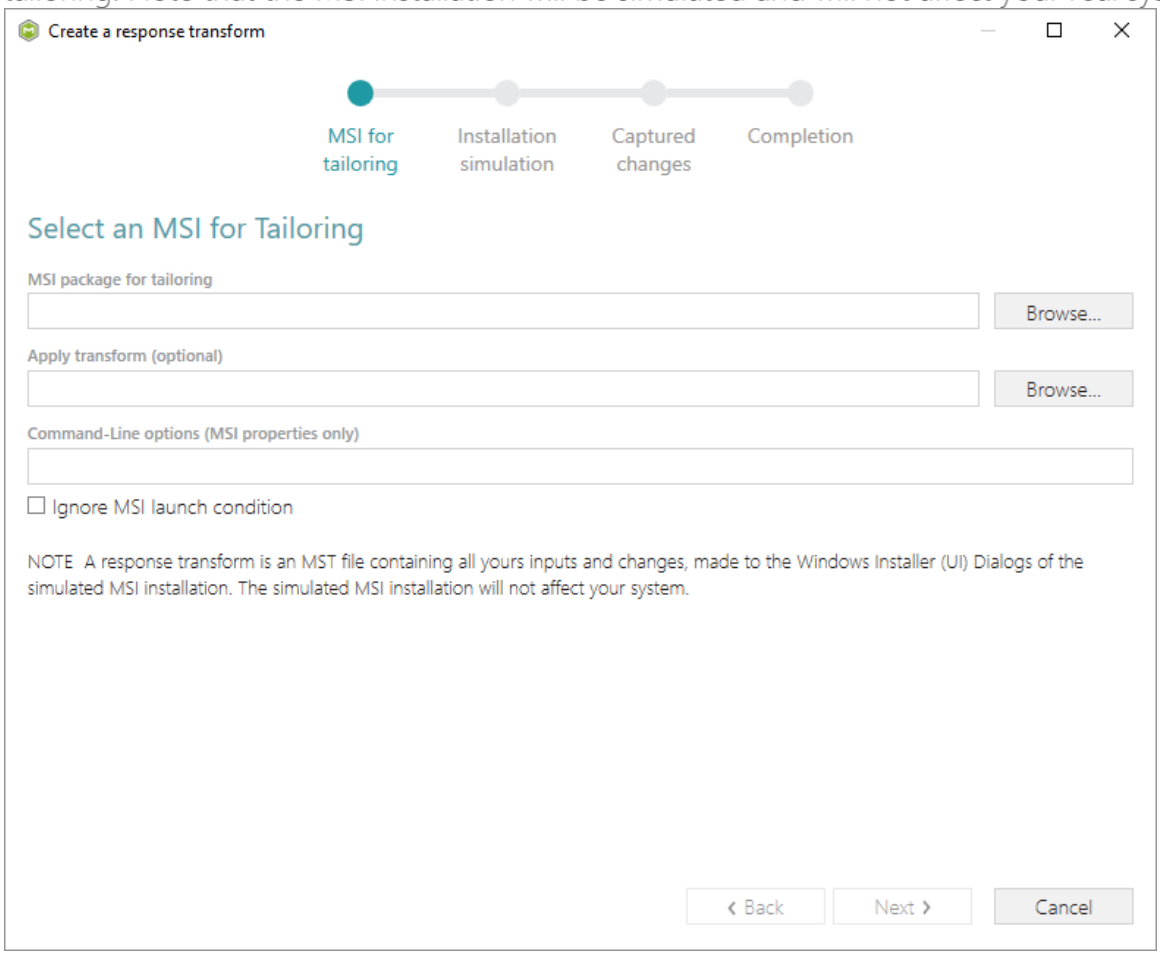
- If you have the User Account Control enabled, click Yes to launch the application.



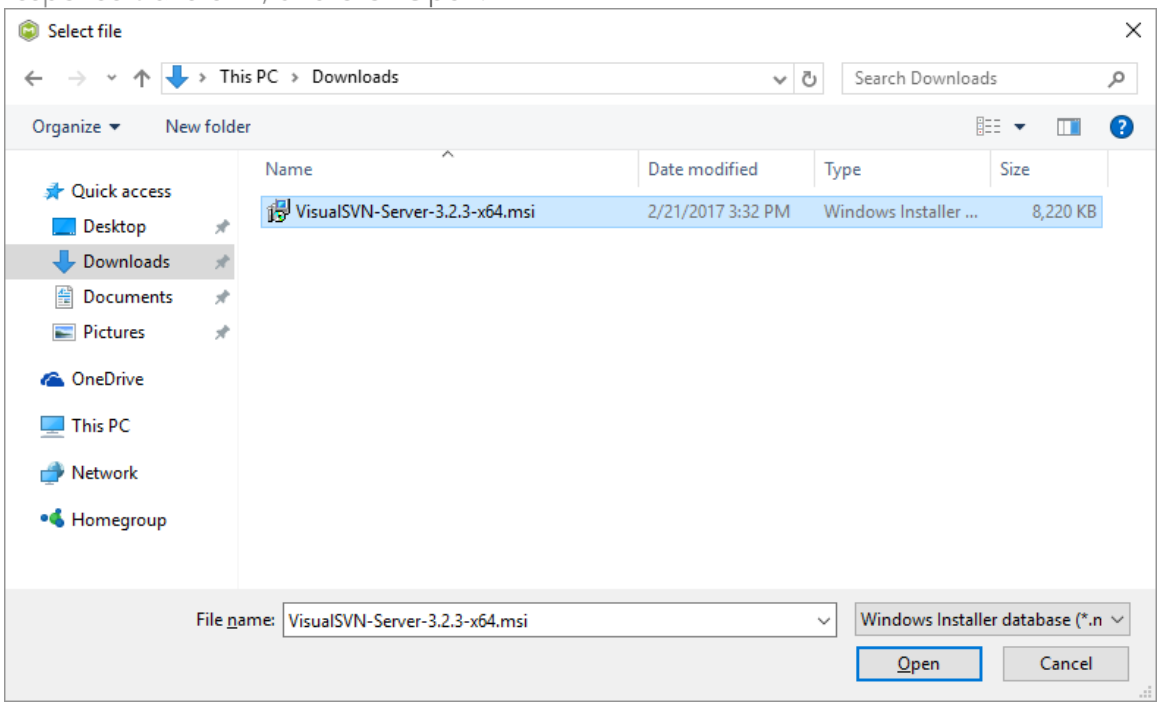
- Click Create response transform.



- Click Browse..., located next to the MSI package for tailoring field, to select an MSI package for tailoring. Note that the MSI installation will be simulated and will not affect your real system.

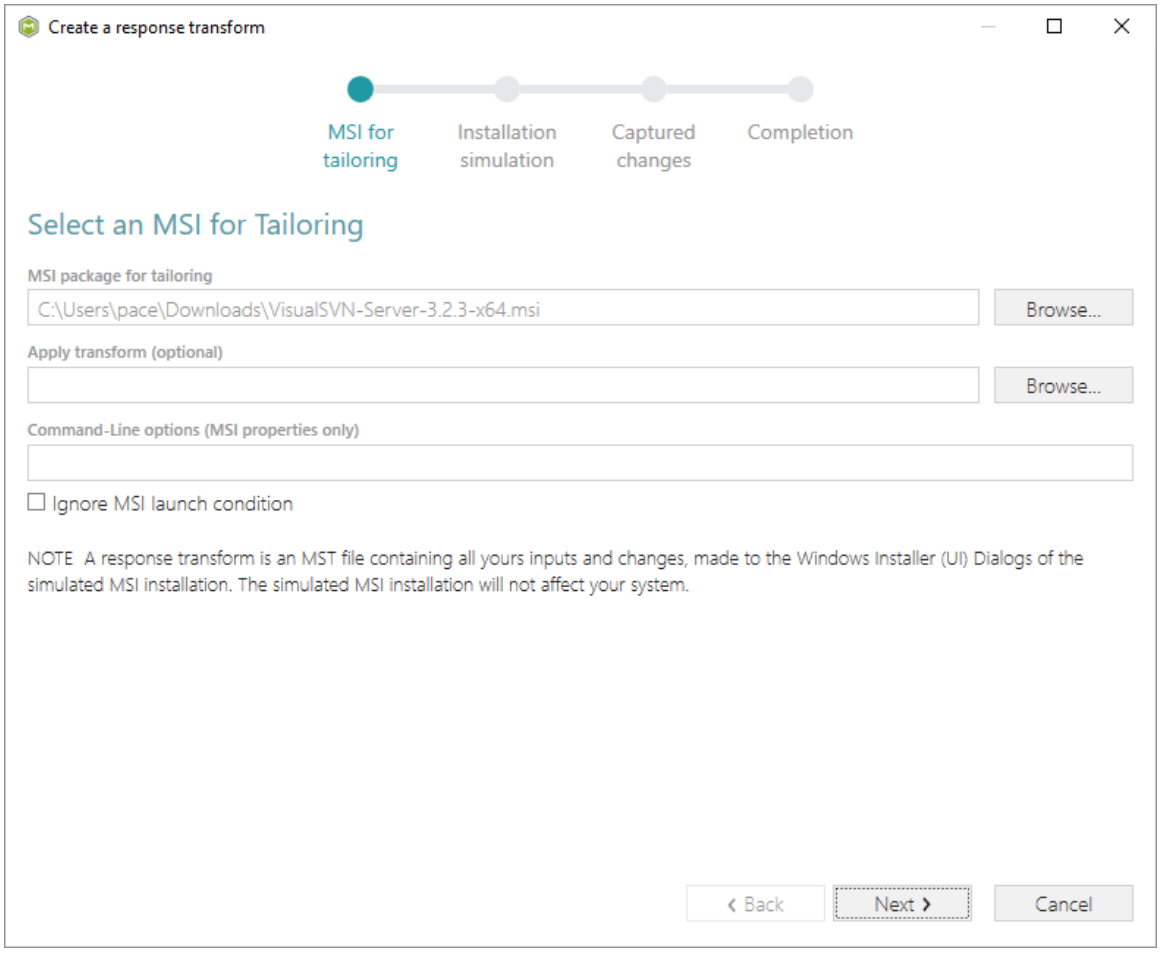


- Choose an MSI package (e.g. VisualSVN-Server-3.2.3-x64.msi), for which you want to create a response transform, and click Open.

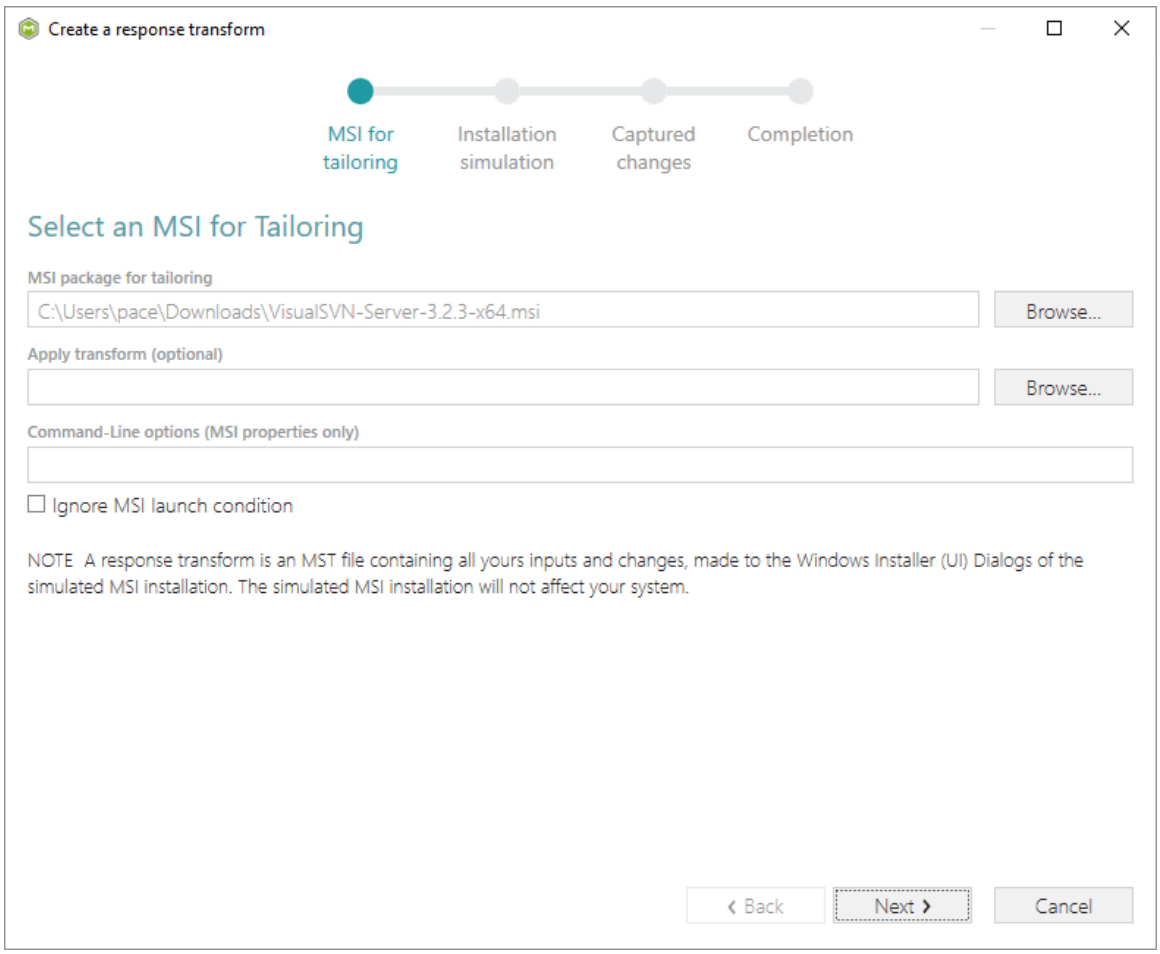


- The Apply transform (optional) and Command-Line options (MSI properties only) fields are optional and they allow applying additional MST and setting MSI properties before the MSI tailoring. Select the Ignore MSI launch condition option to skip all condition from the

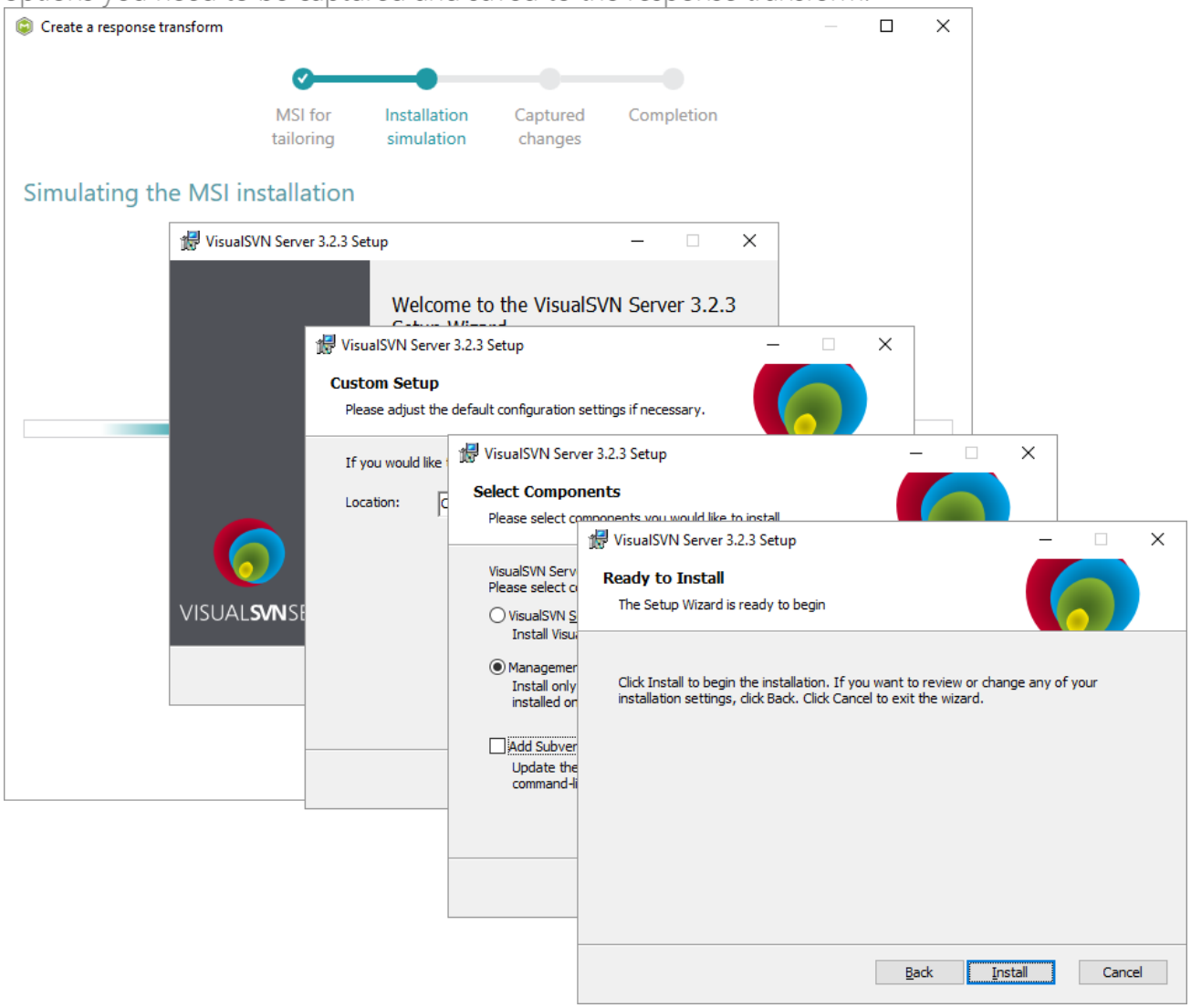
LaunchCondition MSI table.



- Click Next to start the MSI installation simulation.

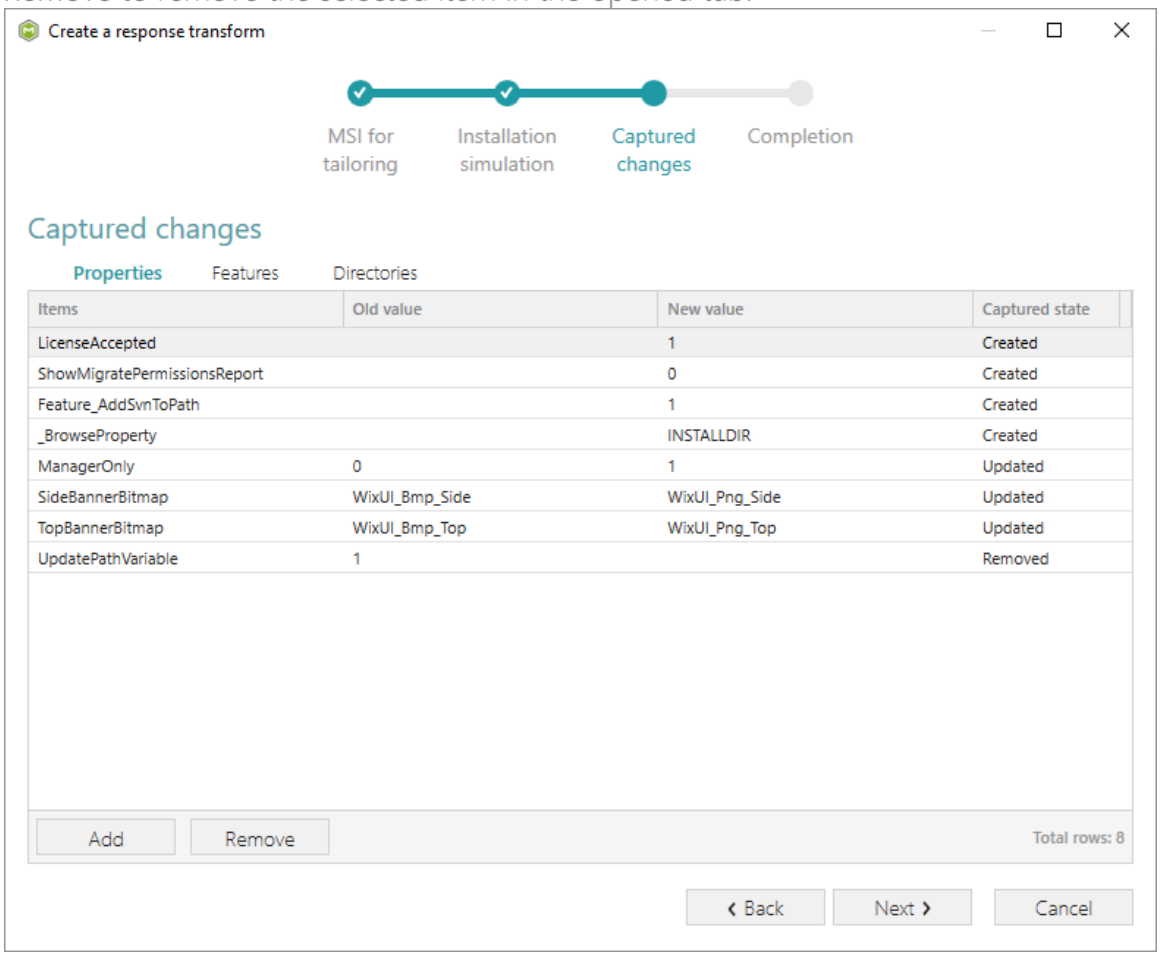


- Complete the simulated MSI installation following the Windows Installer dialogs and choosing options you need to be captured and saved to the response transform.

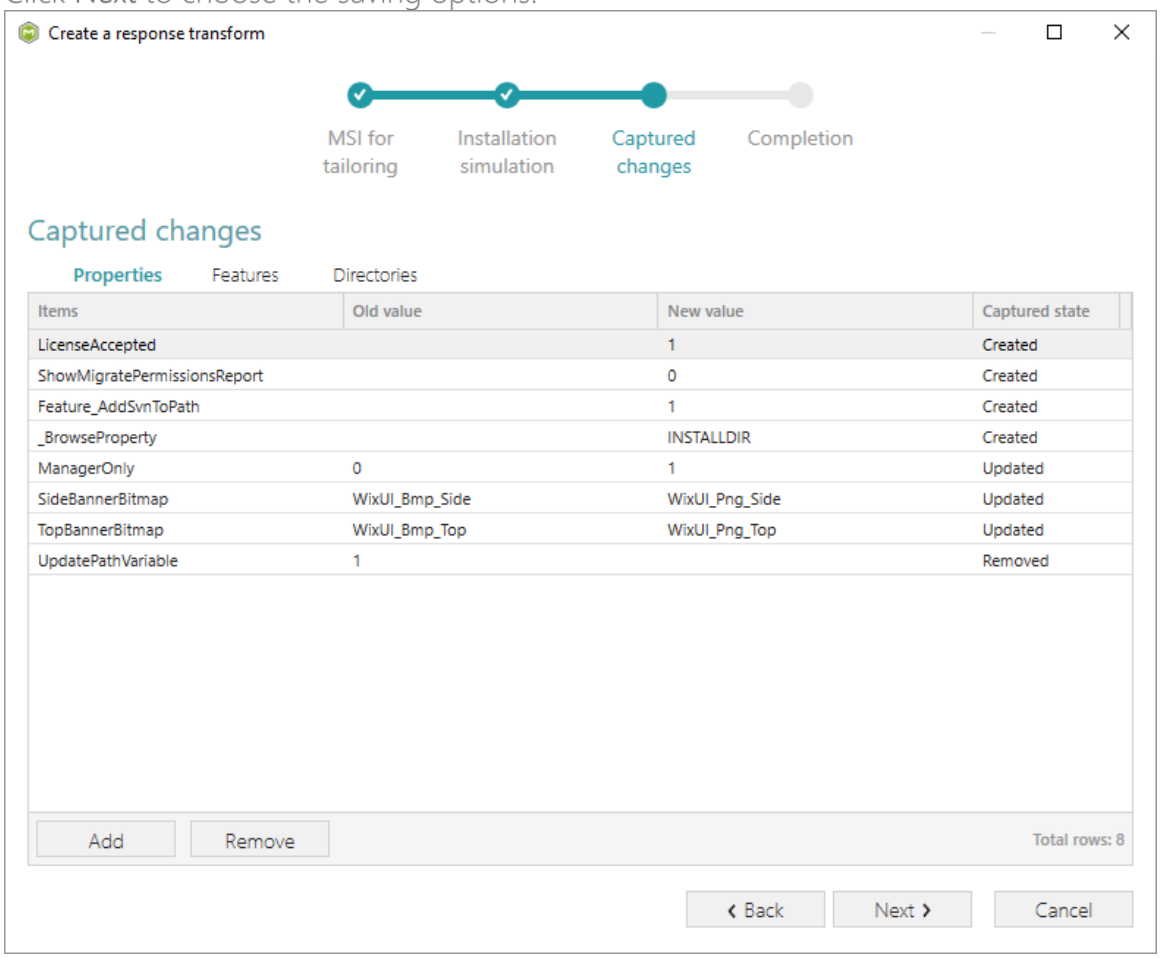


- Review captured information, which will be saved to the response transform (MST) file, by opening the Properties, Features and Dicrectories tabs. Click Add to create new or click

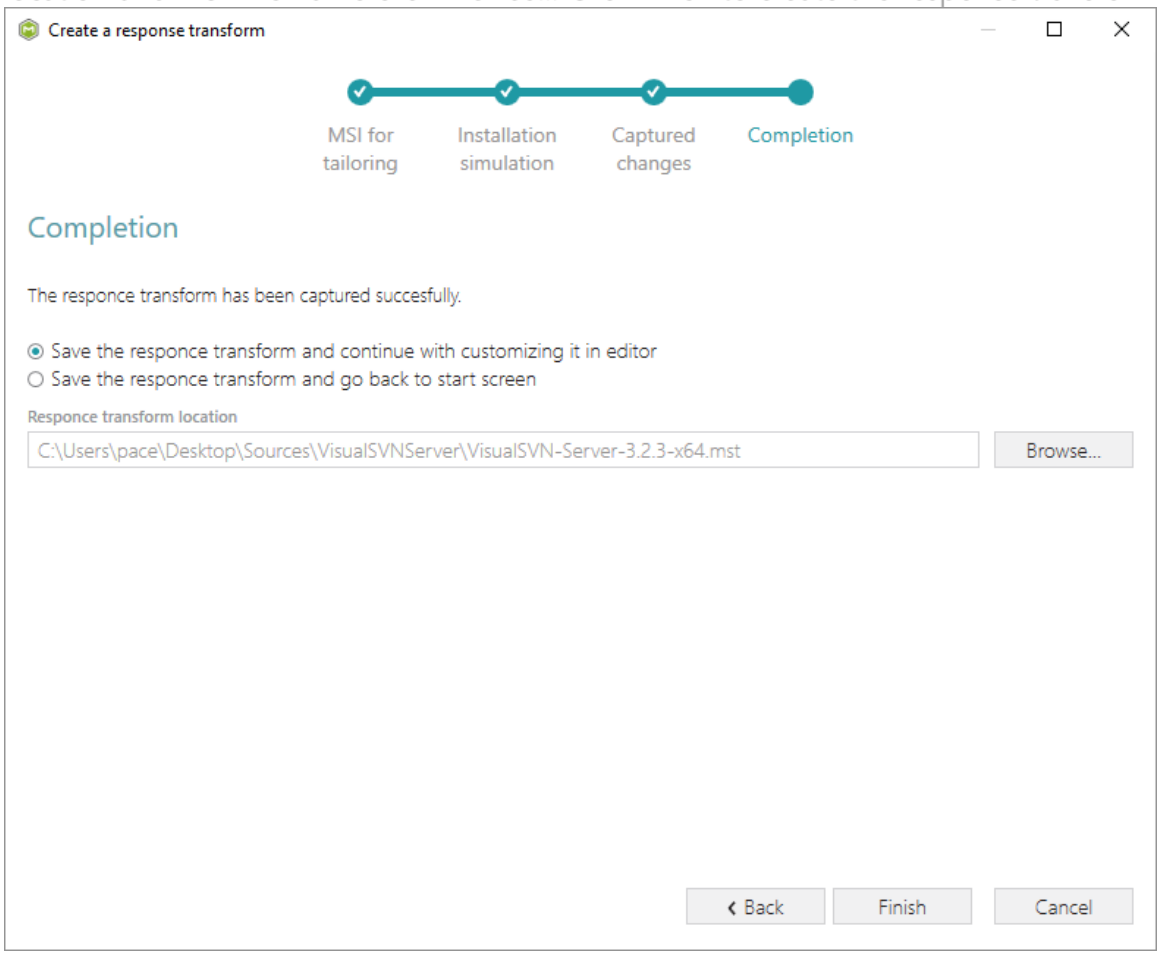
Remove to remove the selected item in the opened tab.



- Click Next to choose the saving options.



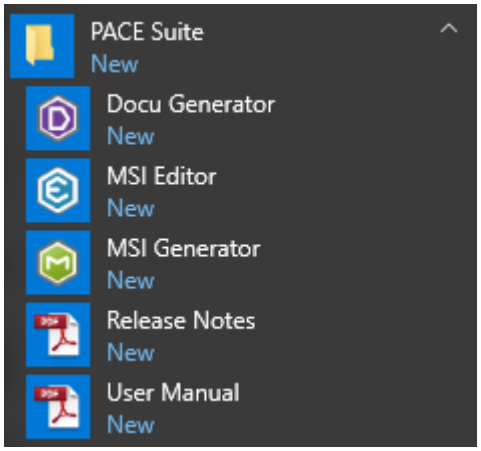
- We recommend to leave the Save the response transform and continue with customizing it in editor option selected to open the generated MST in MSI Editor. For changing the destination location and MST file name click Browse... Click Finish to create the response transform.



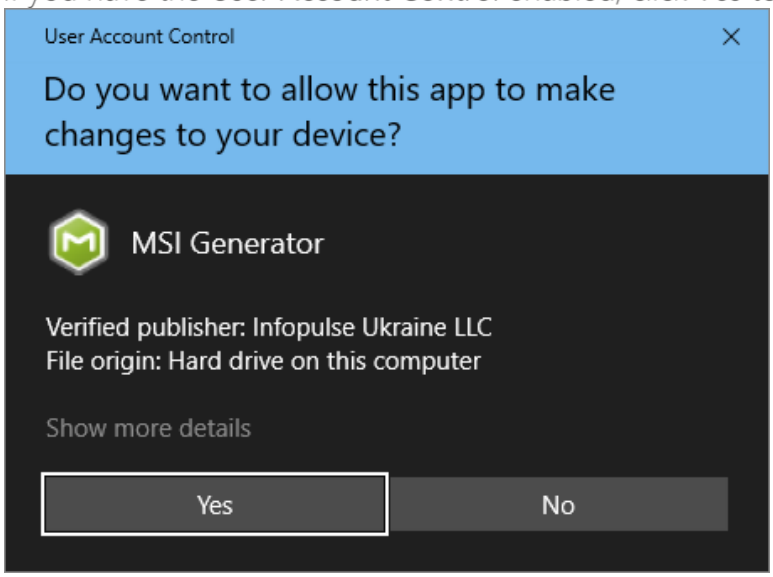
3.2.3 Capture System Changes to MST

Capture system changes, which were made to the file system and registry, by script, by application or manually and save them into MST file, generated against the selected MSI package using MSI Generator. Along with files and registry, services and environment variables, you can capture the modified permissions to the file system and registry.

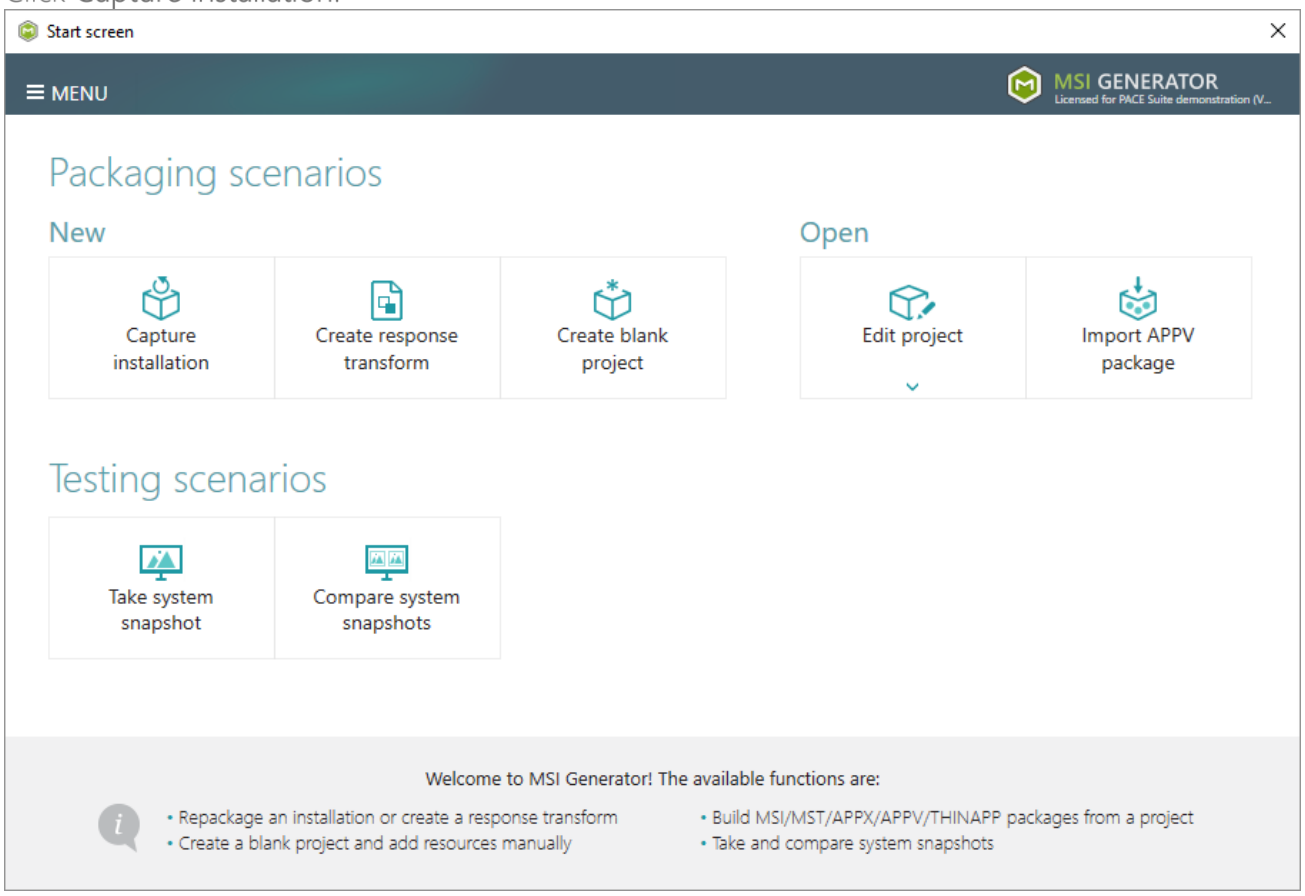
- Launch MSI Generator from the desktop of the start menu shortcut.



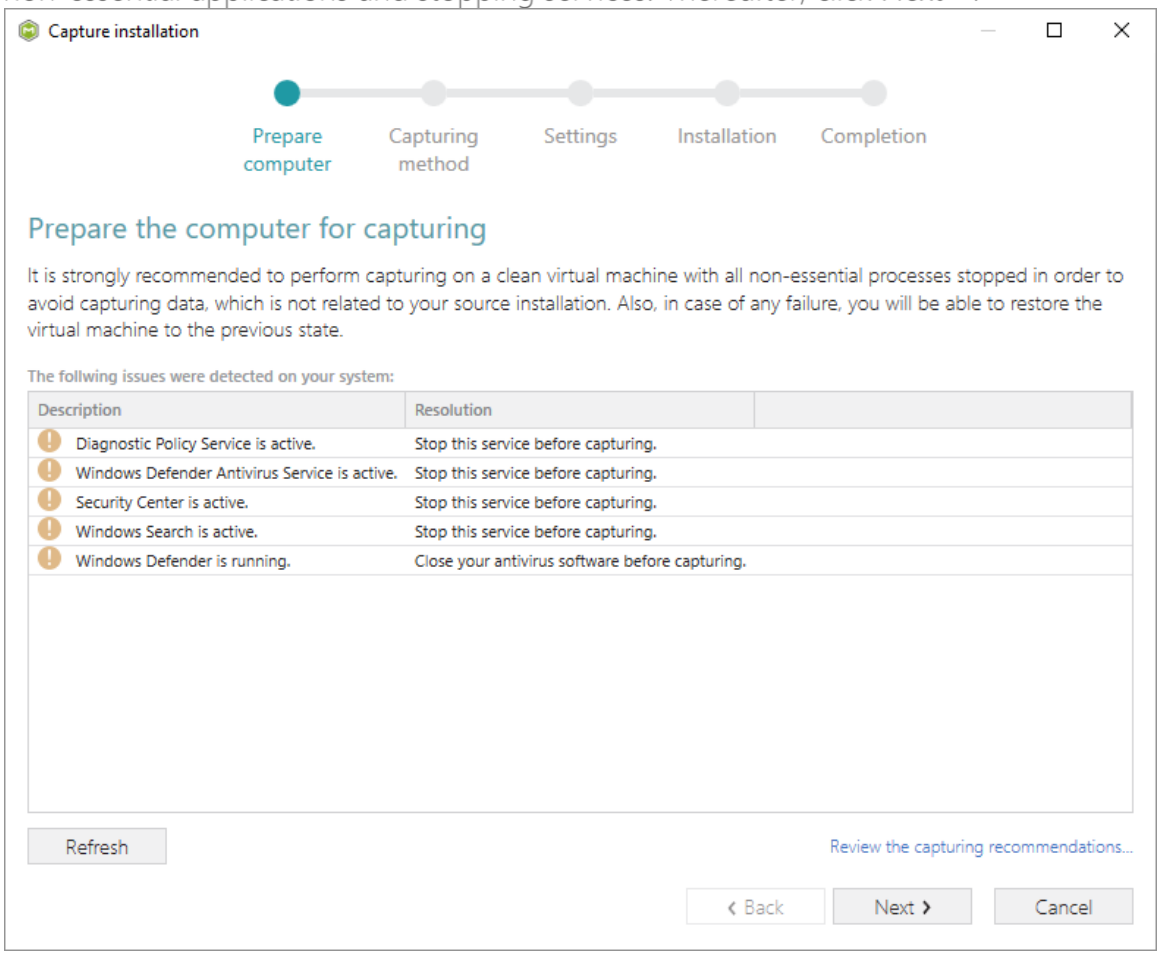
- If you have the User Account Control enabled, click Yes to launch the application.



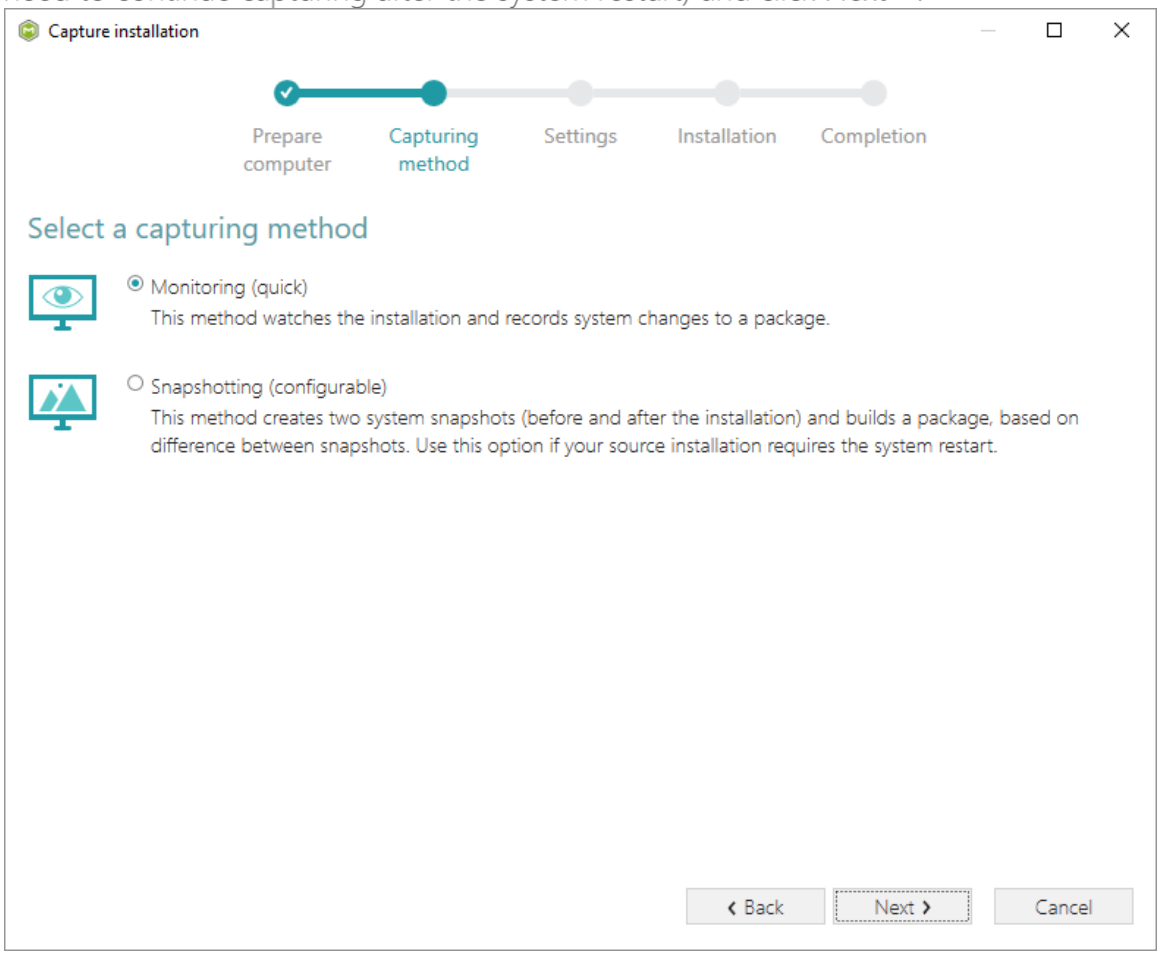
- Click Capture installation.



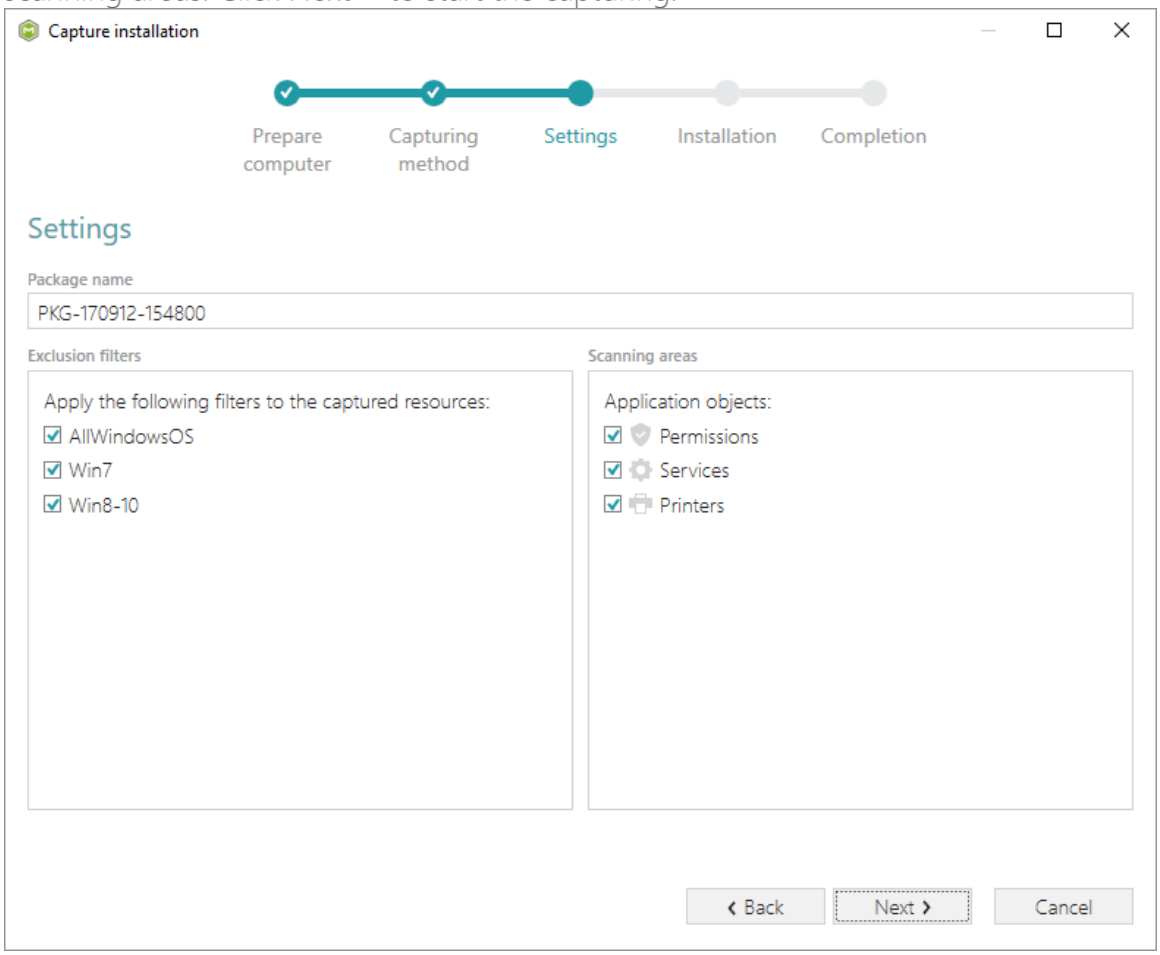
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



- Select the **Monitoring** method for the quicker capturing (or use the **Snapshotting** one if you need to continue capturing after the system restart) and click **Next >**.

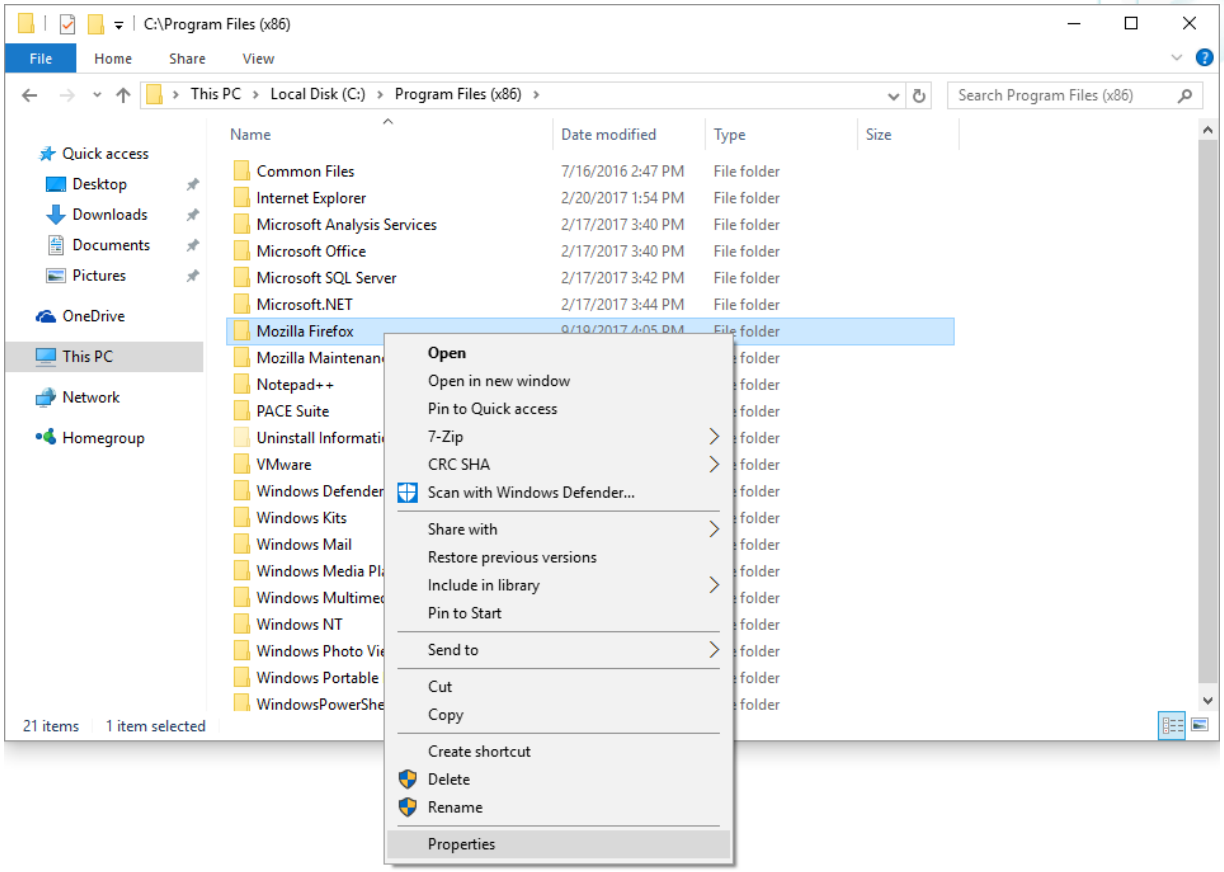


- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.

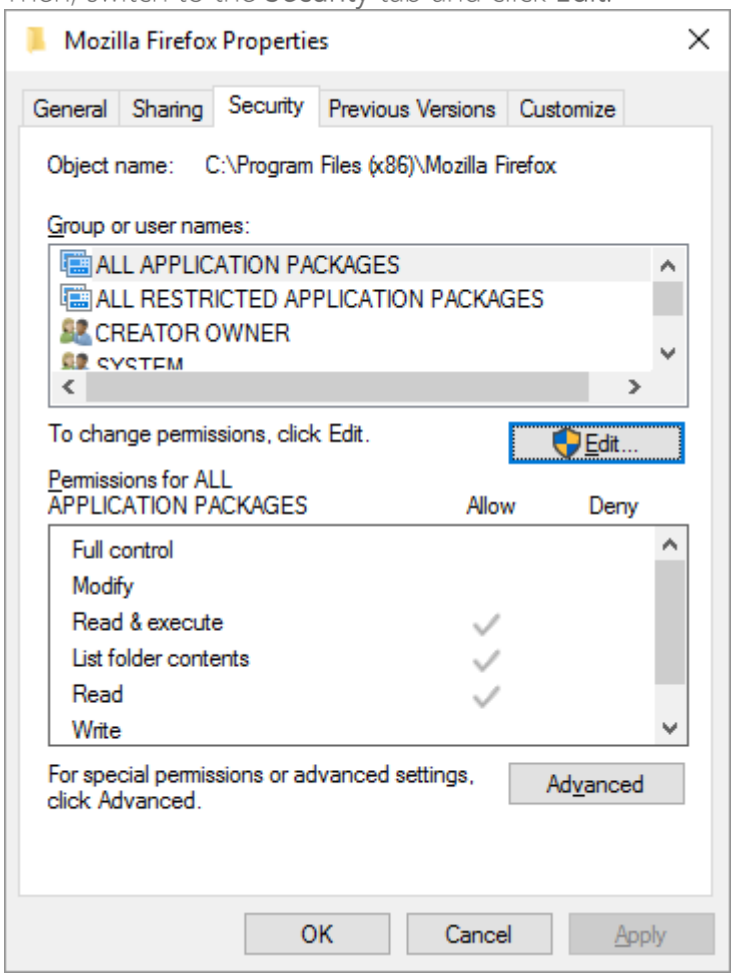


- Now you can make any changes to the file system and registry, which you want to capture and include to the package. For instance, you can copy files, import REG file to the system registry, change file, folder permissions, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.

  In case of necessity, to change the default permission settings for a folder or a file, select Properties from the Windows Explorer context menu of a folder, for which you want to set new permissions.

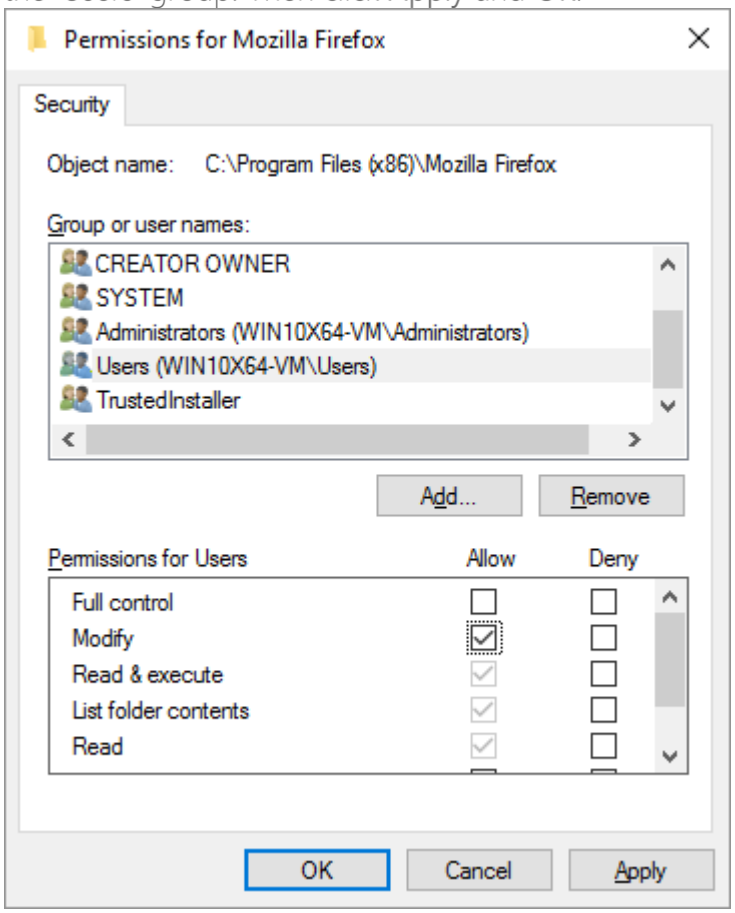


Then, switch to the Security tab and click Edit.

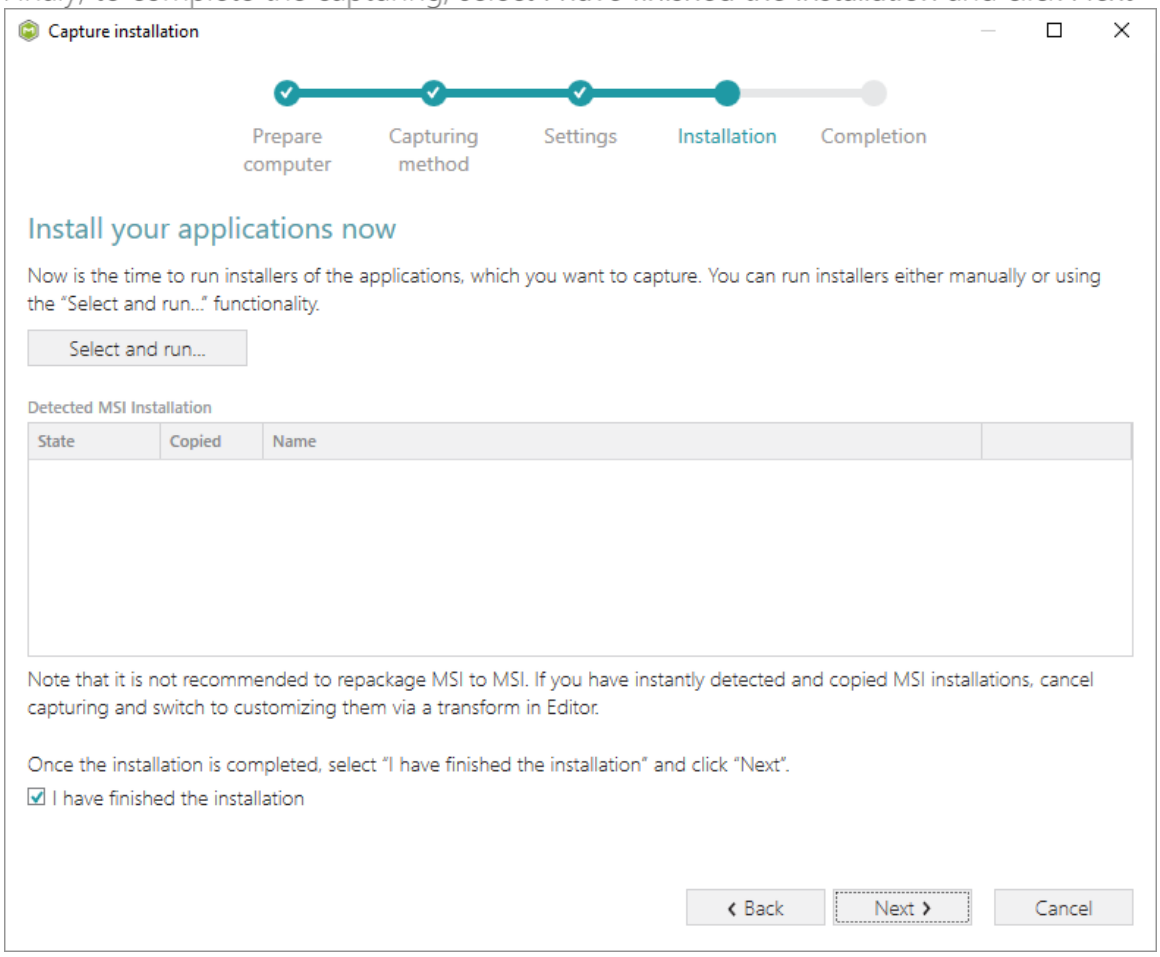


And select a group or a user, for which you want to change the permissions and then select the necessary access rights. For example, lets set the 'Write' and the 'Modify' access rights for

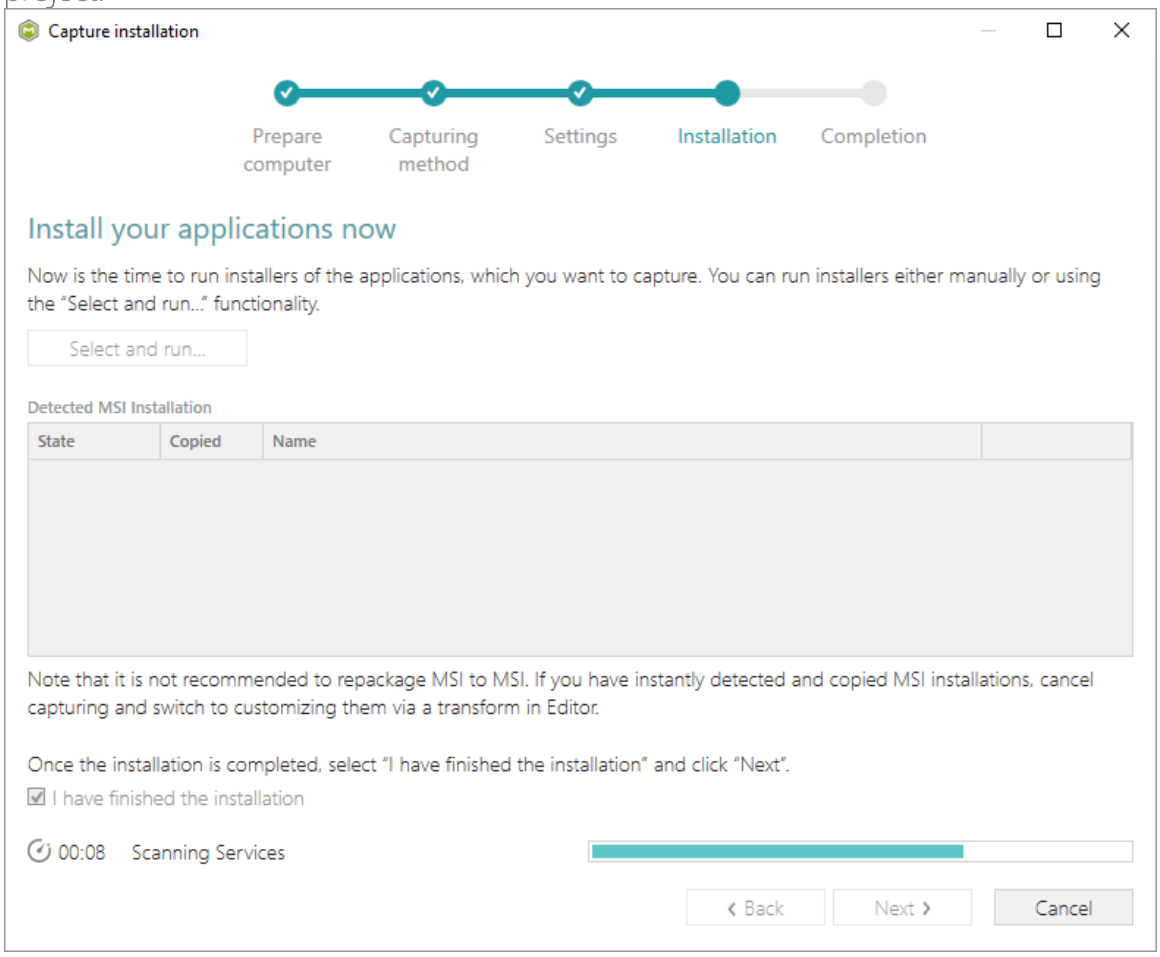
the 'Users' group. Then click Apply and OK.



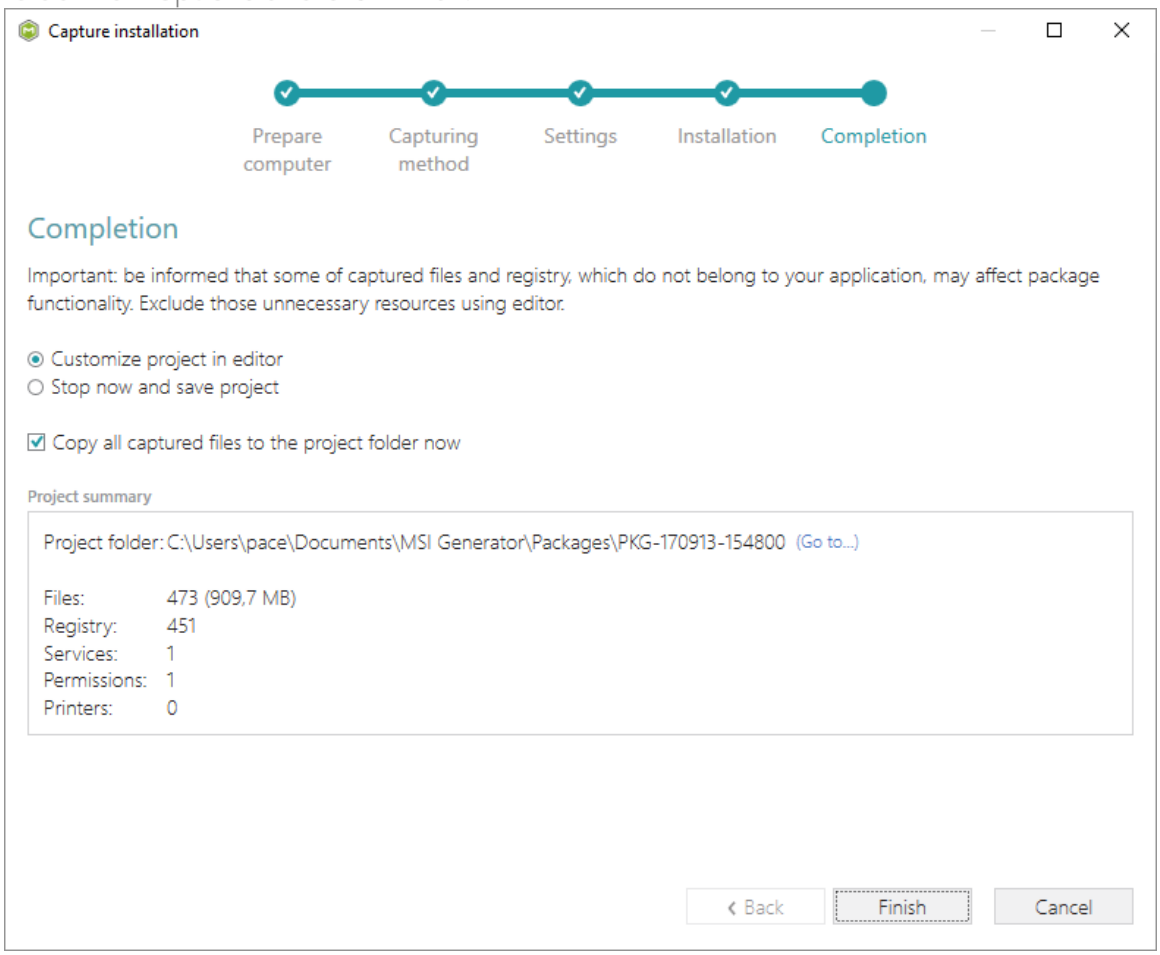
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



- Wait a little, while the capturing process is finishing, filtering captured data and creating the project.

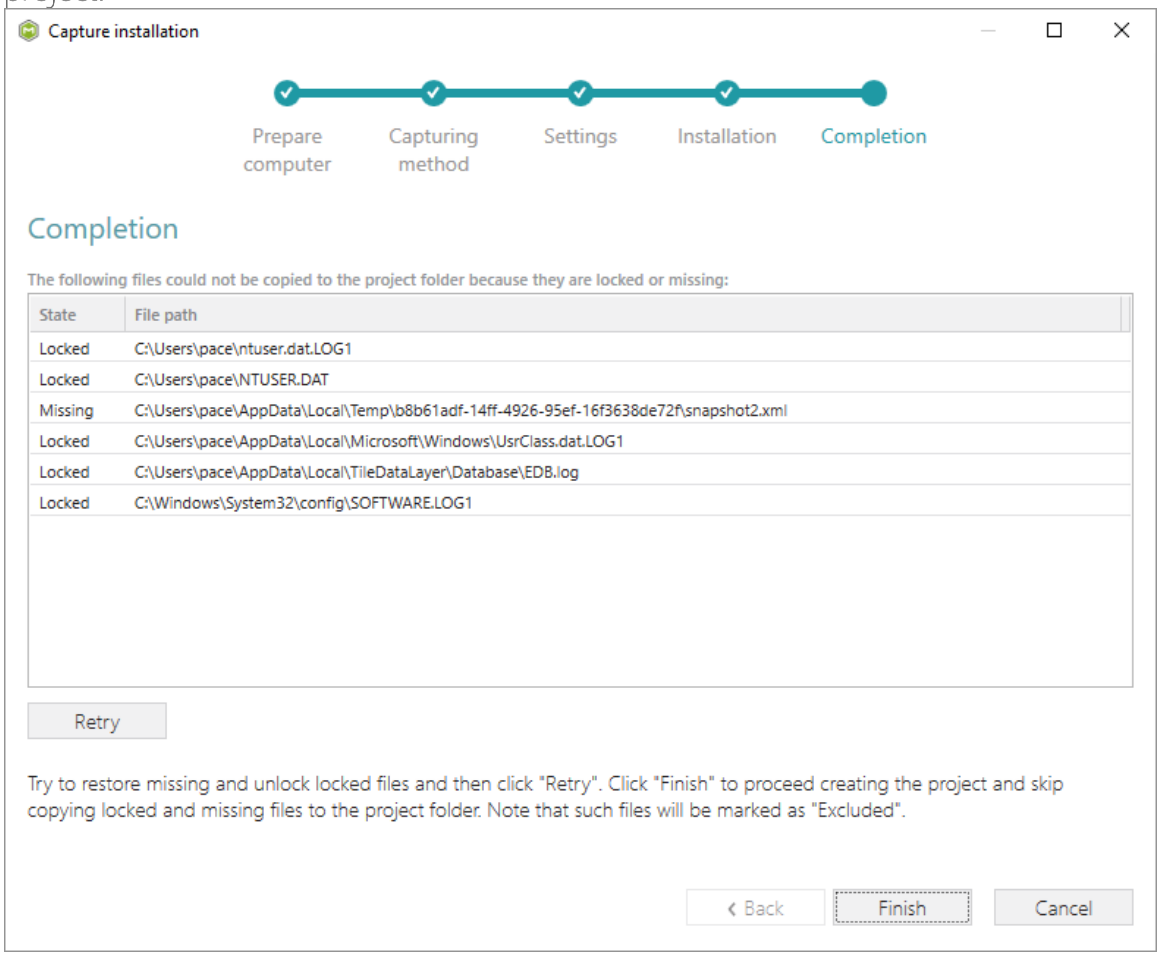


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



- If some of captured files are locked or missing and as result they could not be copied to the project folder, they will be displayed as illustrated below. Usually, the displayed files are system noise and do not refer to the captured application. Click Finish to continue with skiping locked and missing files. Note that such files will be automatically marked as 'Excluded' in your

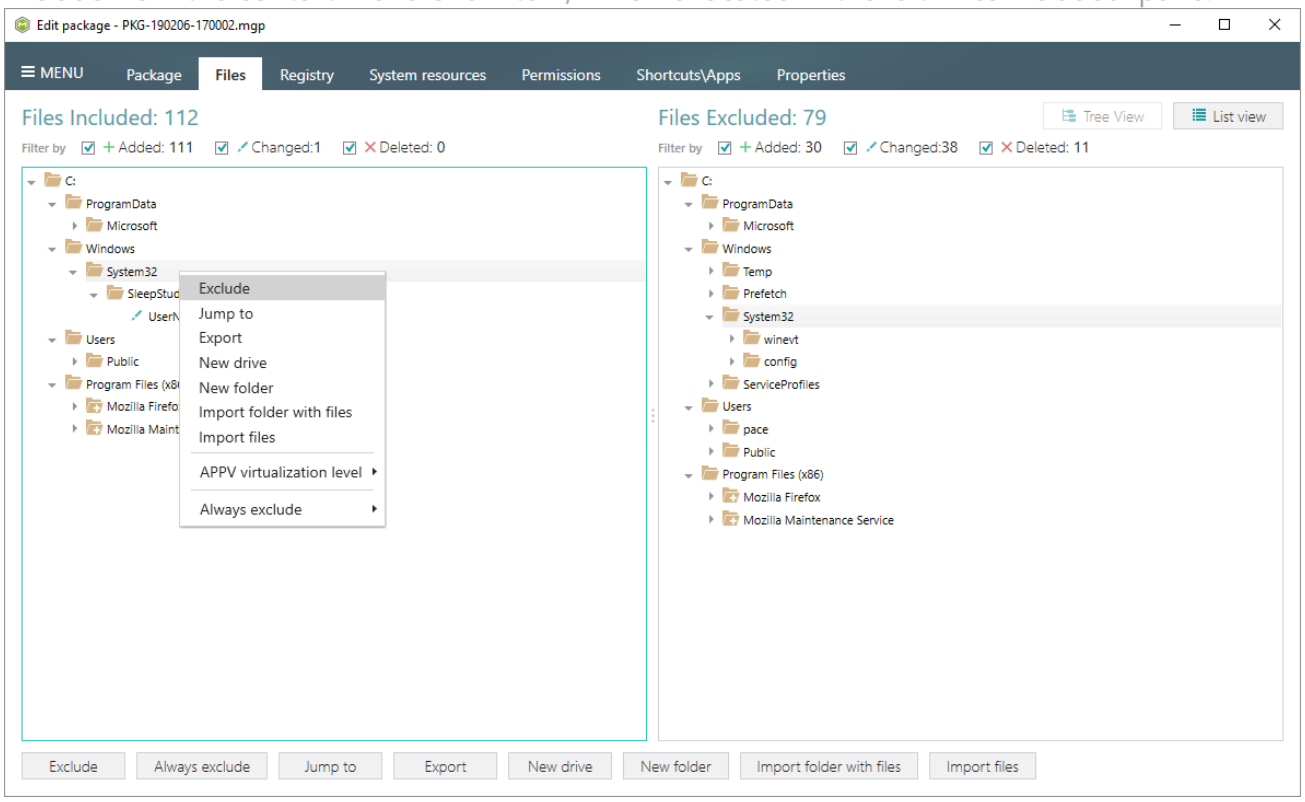
project.



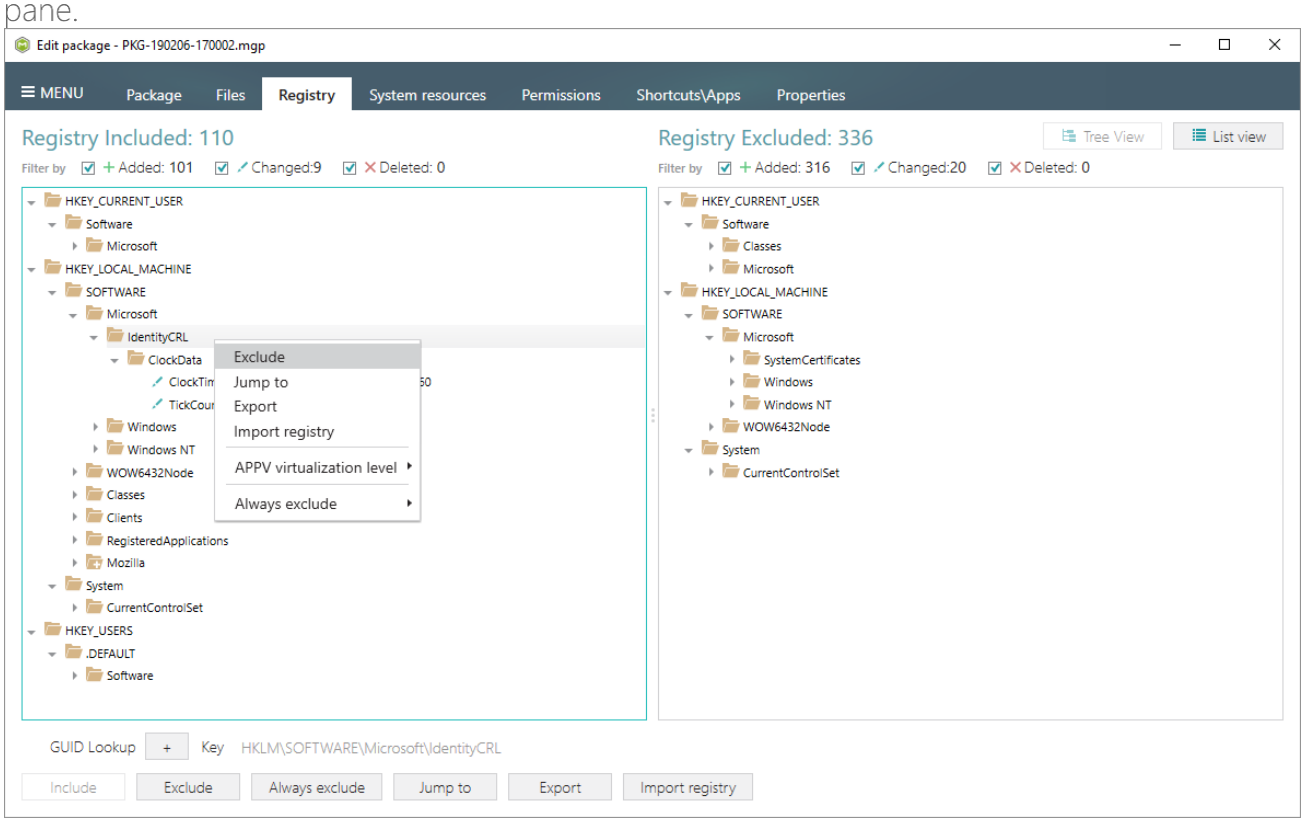
- Once the project is opened in the project editor, it is strongly recommended to review the Files, Registry, System resources, Permissions, and Shortcuts\Apps tabs and exclude the accidentally captured resources, which do not belong to the application. Such accidentally captured resources are also called system noise and it could be Windows Updates, Search indexing, Antivirus software and so on.

  An MSI package with such resources may fail during the installation. The common reason for such fails is trying to write a registry value to the system-protected location, strongly prohibited from writing by third-party installations.

- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.

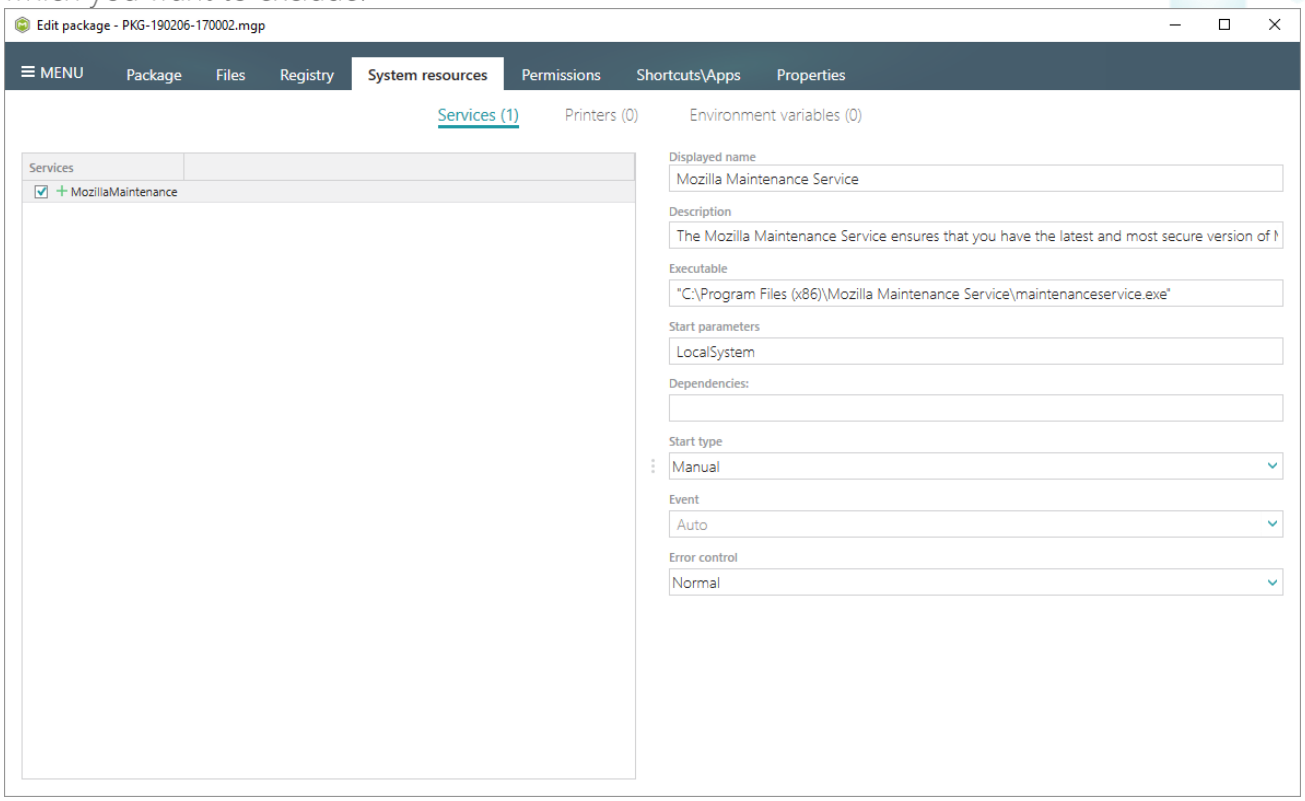


- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.

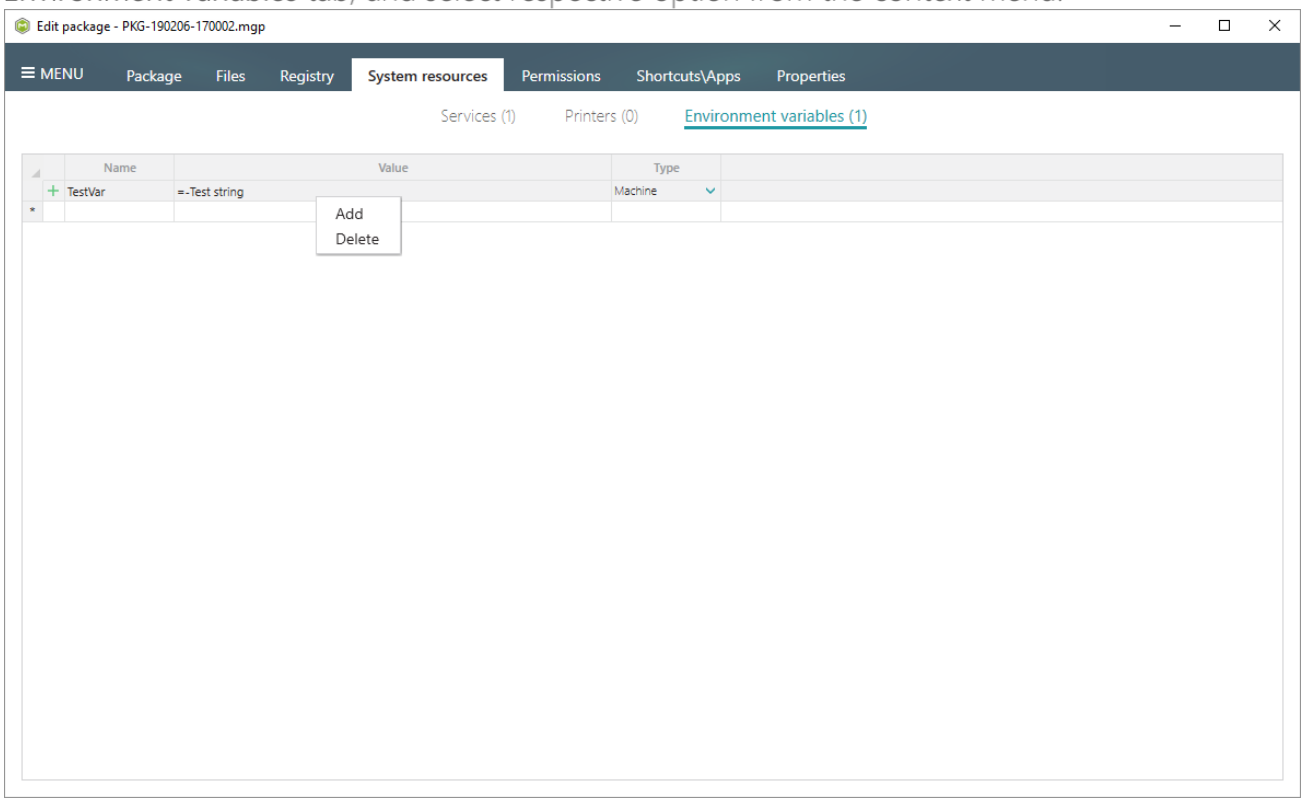


- In order to review and exclude unnecessary services, go to the **System resources** -> **Services** tab, and uncheck the checkbox, located before the service name in the list, for those services,

which you want to exclude.

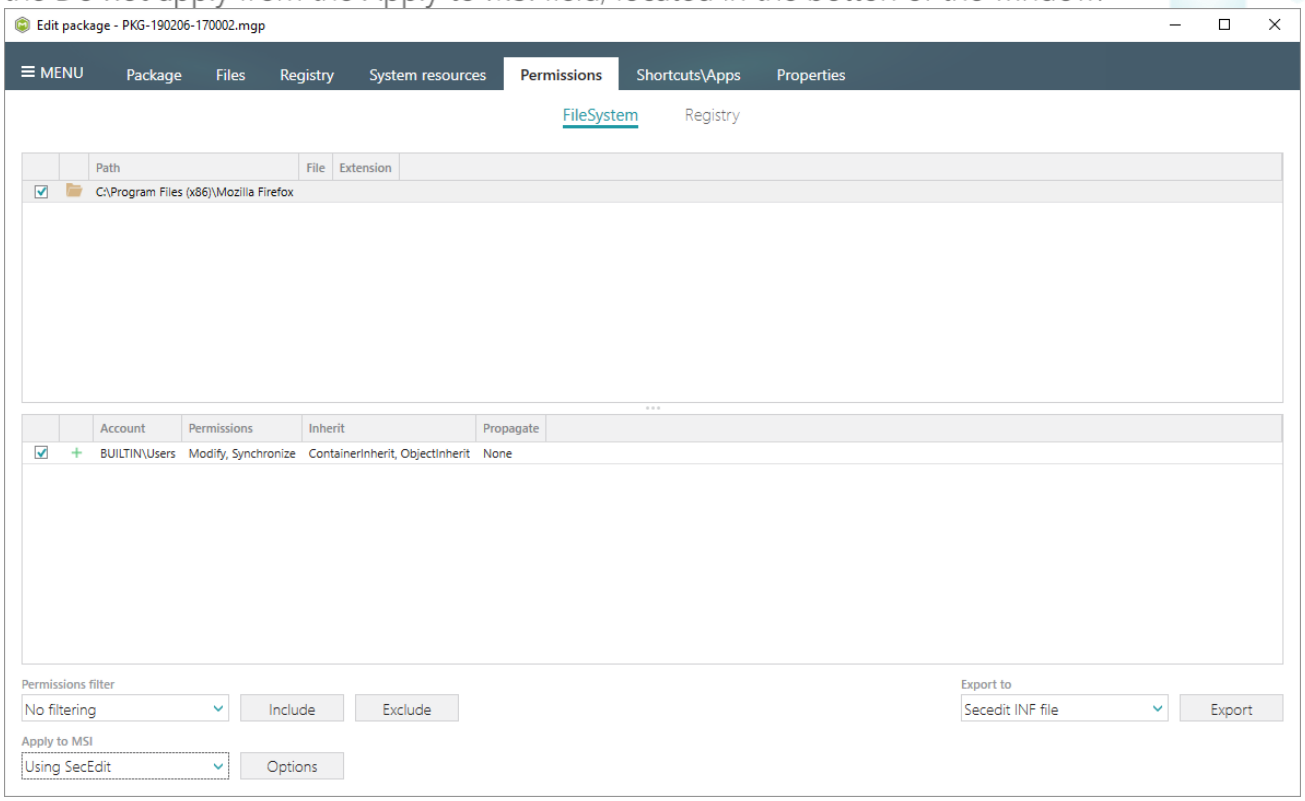


- In order to add and remove Environment variables, go to the System resources -> Environment variables tab, and select respective option from the context menu.

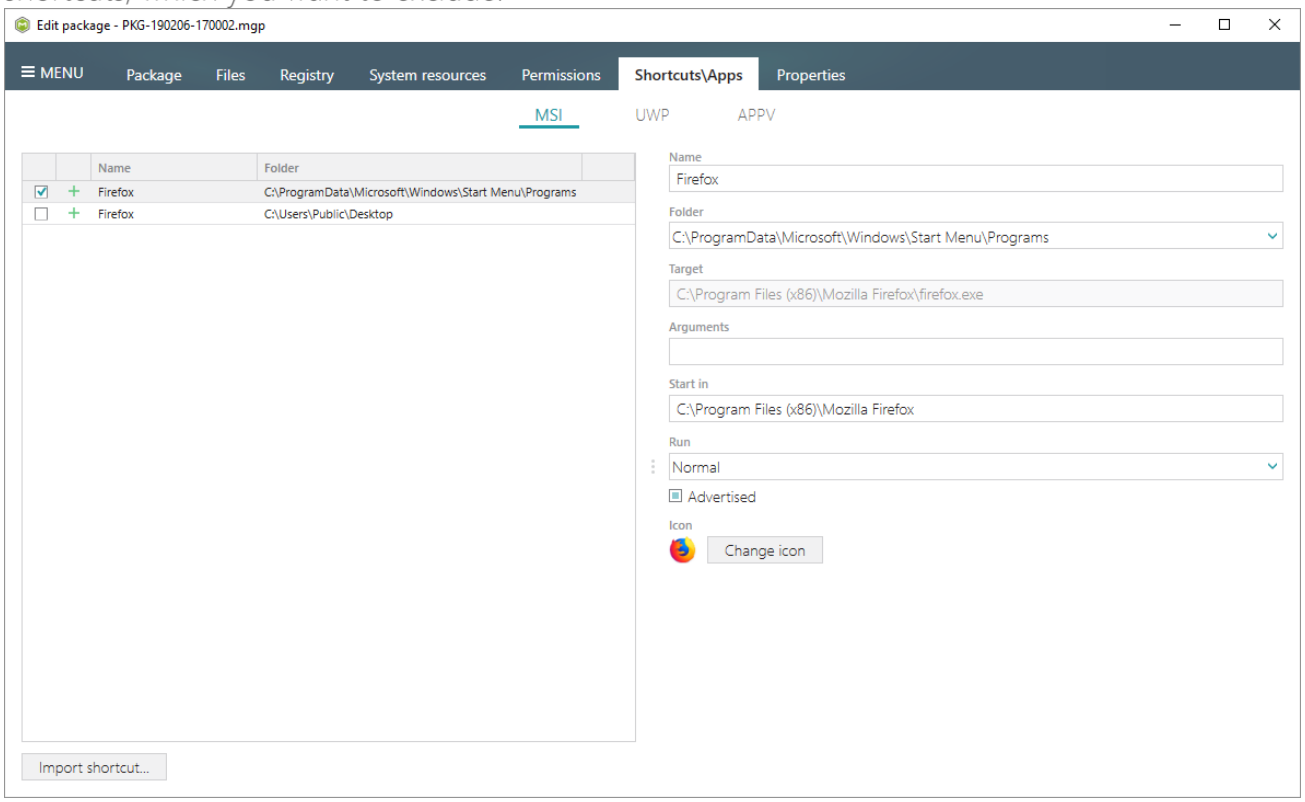


- On the Permissions tab you will find the captured permissions for the file system and registry. For excluding the unnecessary permission changes, either untick the each checkbox, located before the path, for which permission changes were detected, or to skip all permissions select

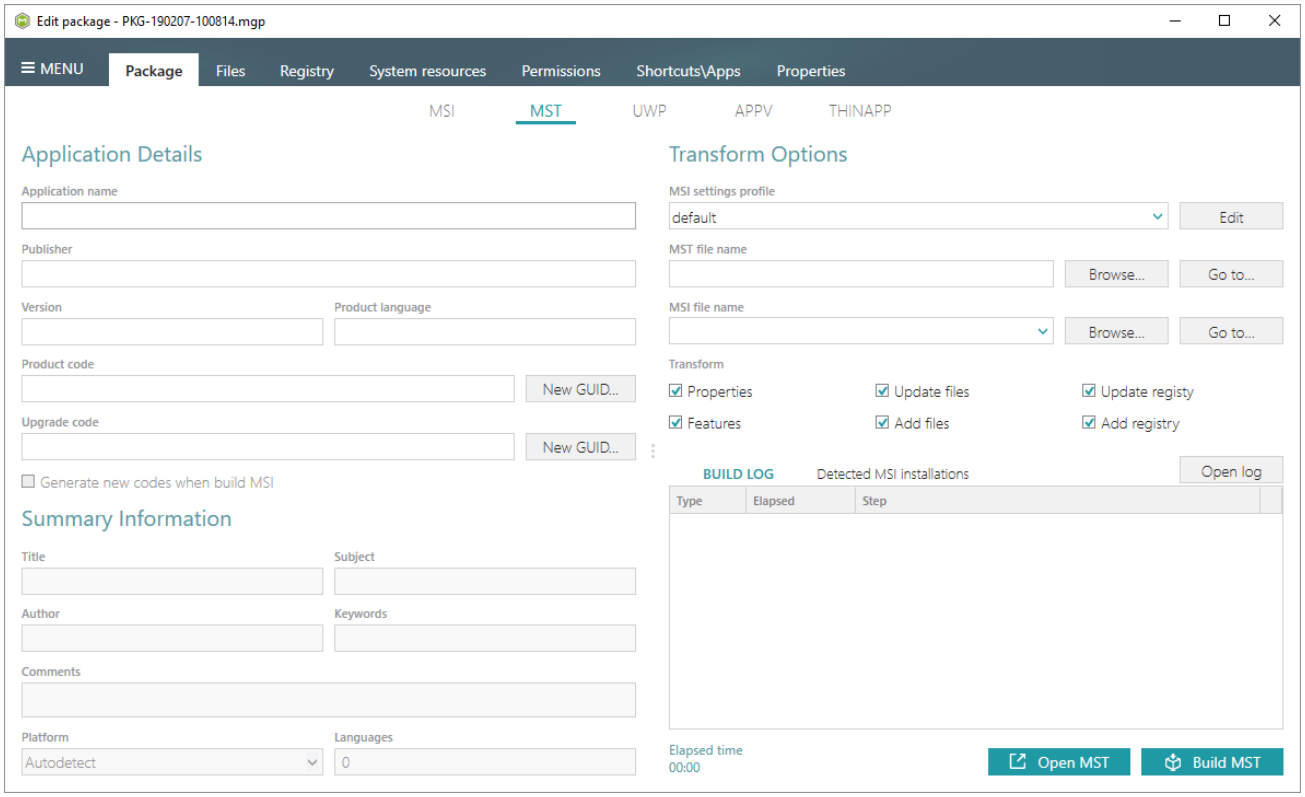
the Do not apply from the Apply to MSI field, located in the botton of the window.



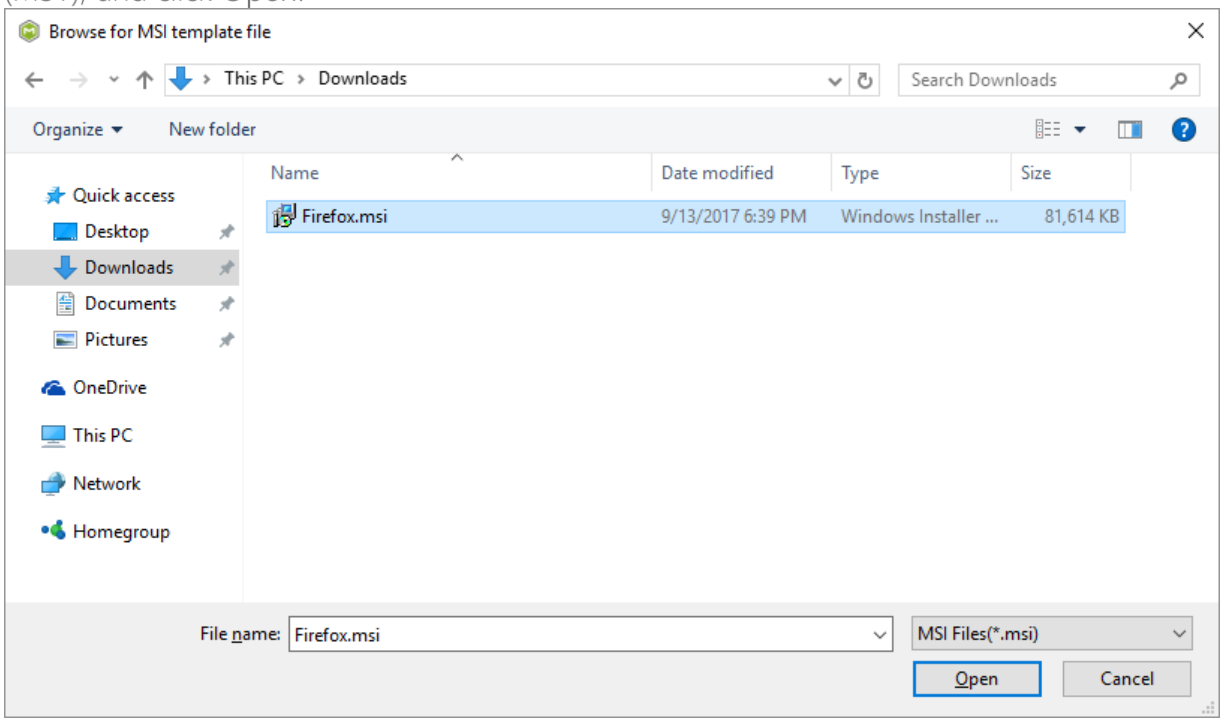
- In order to review and exclude the unnecessary MSI shortcuts, go to the Shortcuts\Apps -> MSI tab, and untick the checkbox, located before the shortcut name in the list, for those shortcuts, which you want to exclude.



- Finaly, to build an MST from the project, navigate to the Package -> MST tab, click Browse..., located next to the MSI file name field.

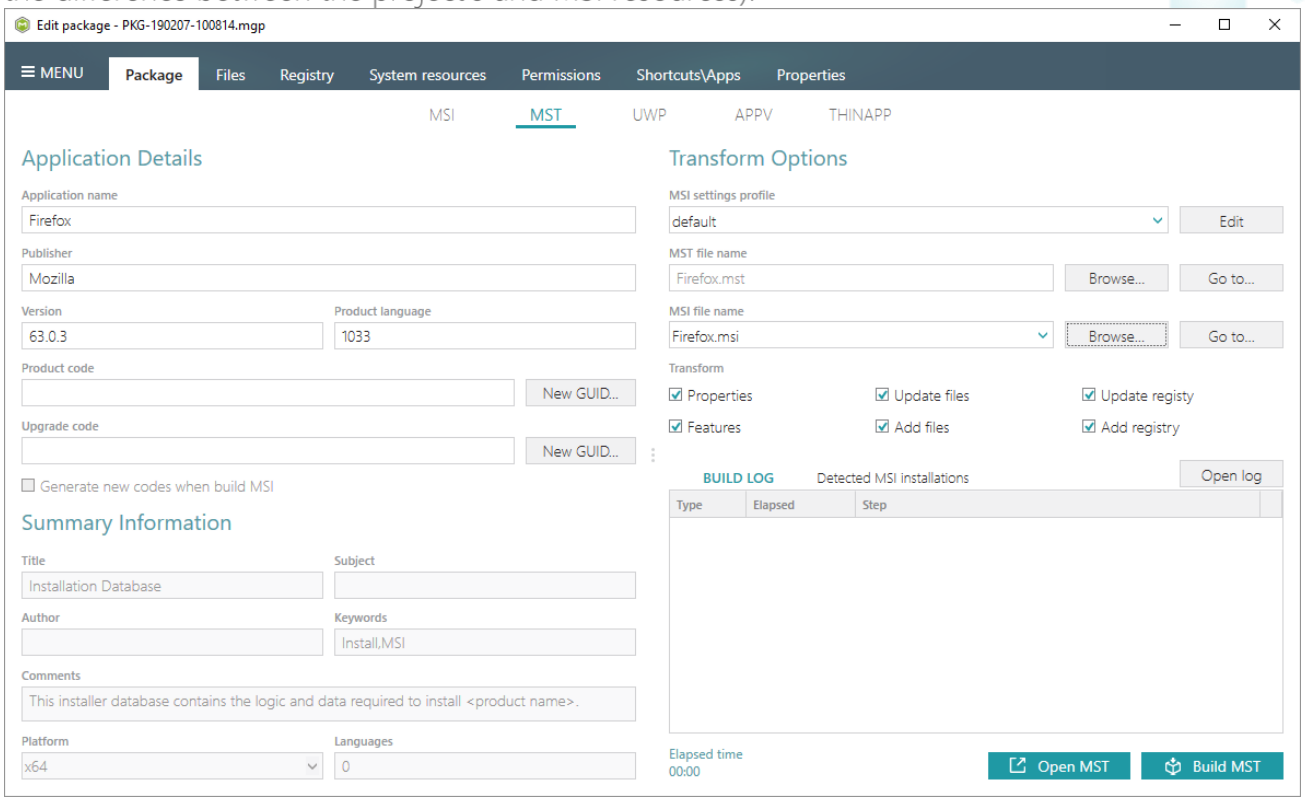


- Choose an MSI package (e.g. Firefox.msi), against to which you want to create a transform (MST), and click Open.

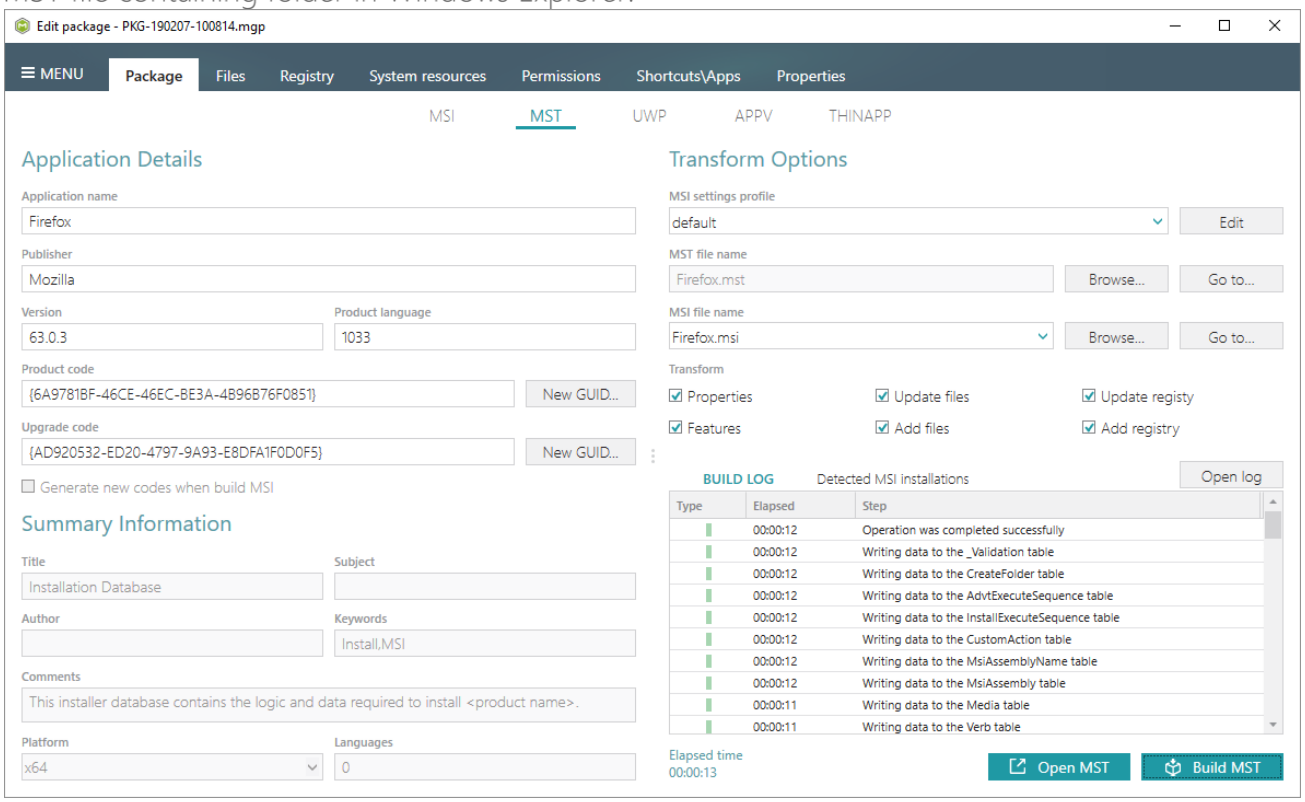


- Update the Application details such as name, publisher, version, language and click Build MST to generate a transform (MST) file. Note that the generated MST will contain only those resources that do not exist in the selected MSI package (in other words, MST will contain only

the difference between the project's and MSI resources).



- Once the MST file is built, click **Go to...**, located next to the **MST file name** field, to open the MST file containing folder in Windows Explorer.



3.3 Create MSP

Choose a scenario that better suits your needs:

- Generate MSP on Difference, described in section 3.3.1
  Create a Patch (MSP) package, based on differences between two MSI packages: recent and old ones.
- Save Changes as MSP, described in section 3.3.2
  Save changes, made to the opened MSI database, as a Patch (MSP) package.

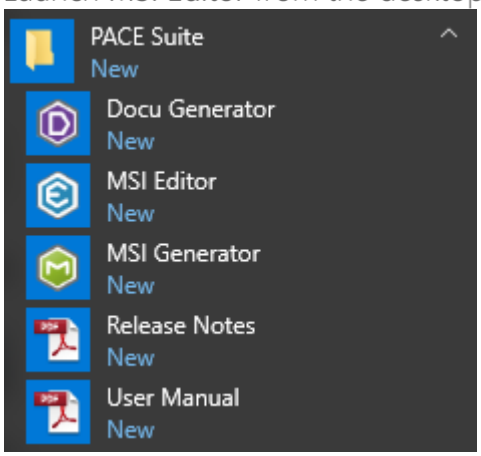
NOTE Adhere to the following restrictions during creating a Patch package:

- Do not move files from one folder to another.
- Do not move files from one CAB to another.
- Do not change the order of files in a CAB.
- Do not change the sequence number of existing files. The sequence number is the value specified in the Sequence column of the File Table.
- Any new files that are added by the patch must be placed at the end of the existing file sequence. The sequence number of any new file in the upgraded image must be greater than the largest sequence number of existing files in the target image.
- Do not change the primary keys in the File Table between the original and new .msi file versions.

3.3.1 Generate MSP on Difference

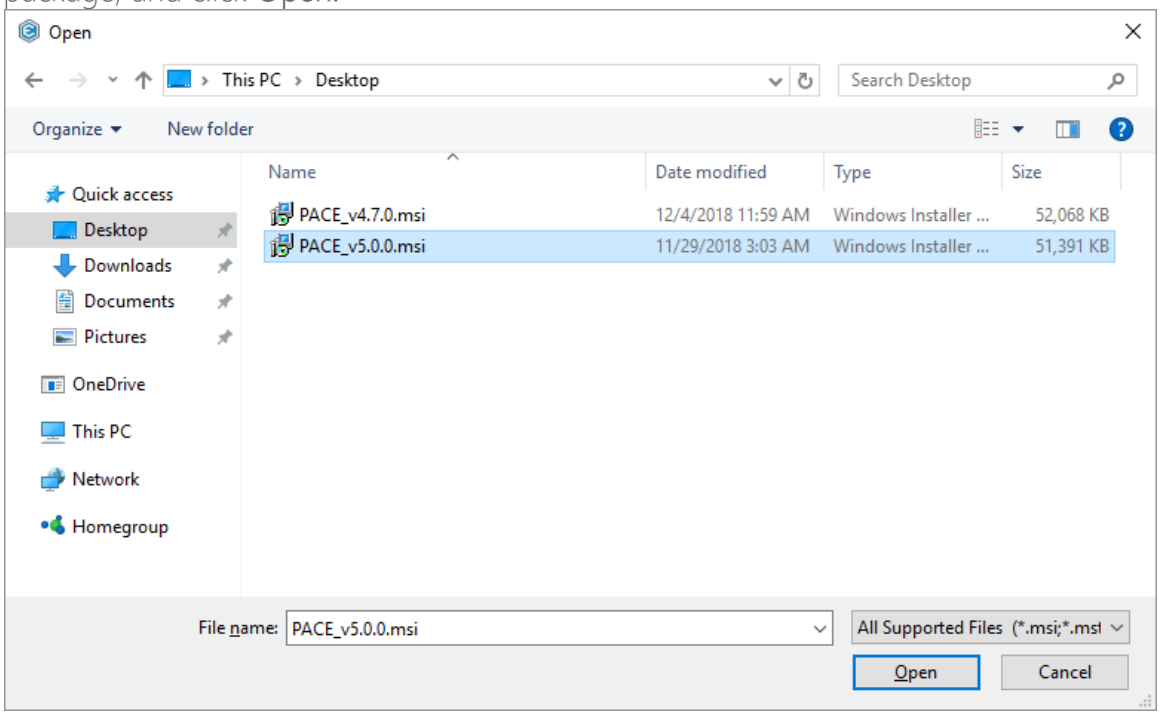
Create a Patch (MSP) package, based on differences between two MSI packages: recent and old ones.

- Launch MSI Editor from the desktop or the start menu shortcut.

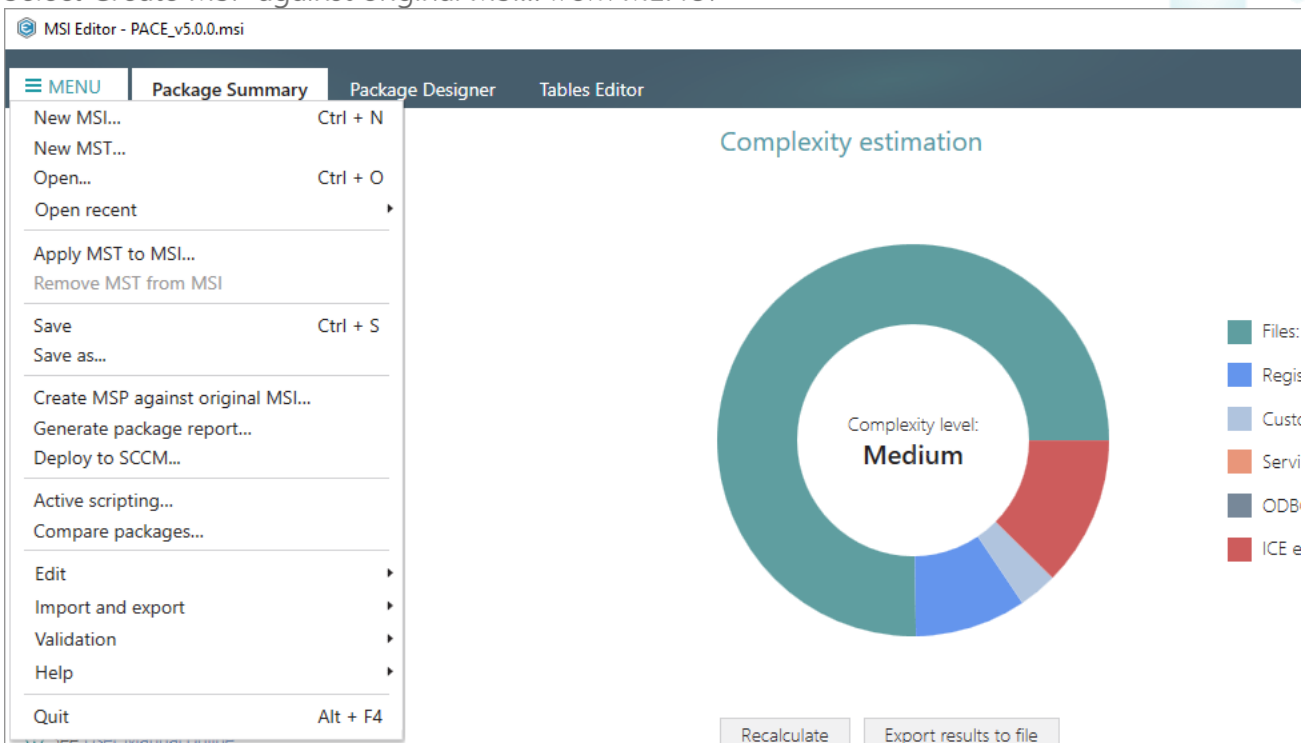


- Select Open... from MENU.

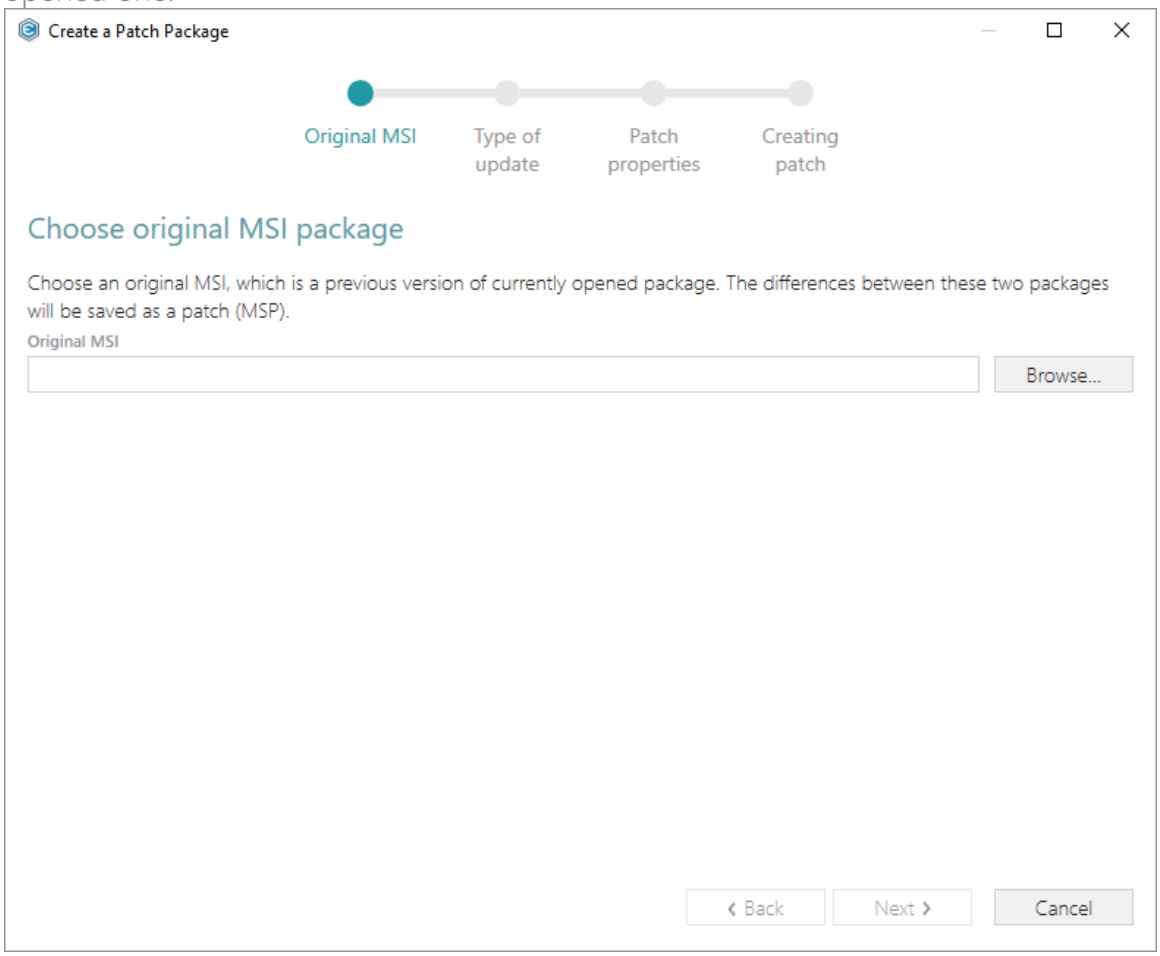
- Choose a recent MSI package, containing new resources in comparison to the previous MSI package, and click Open.



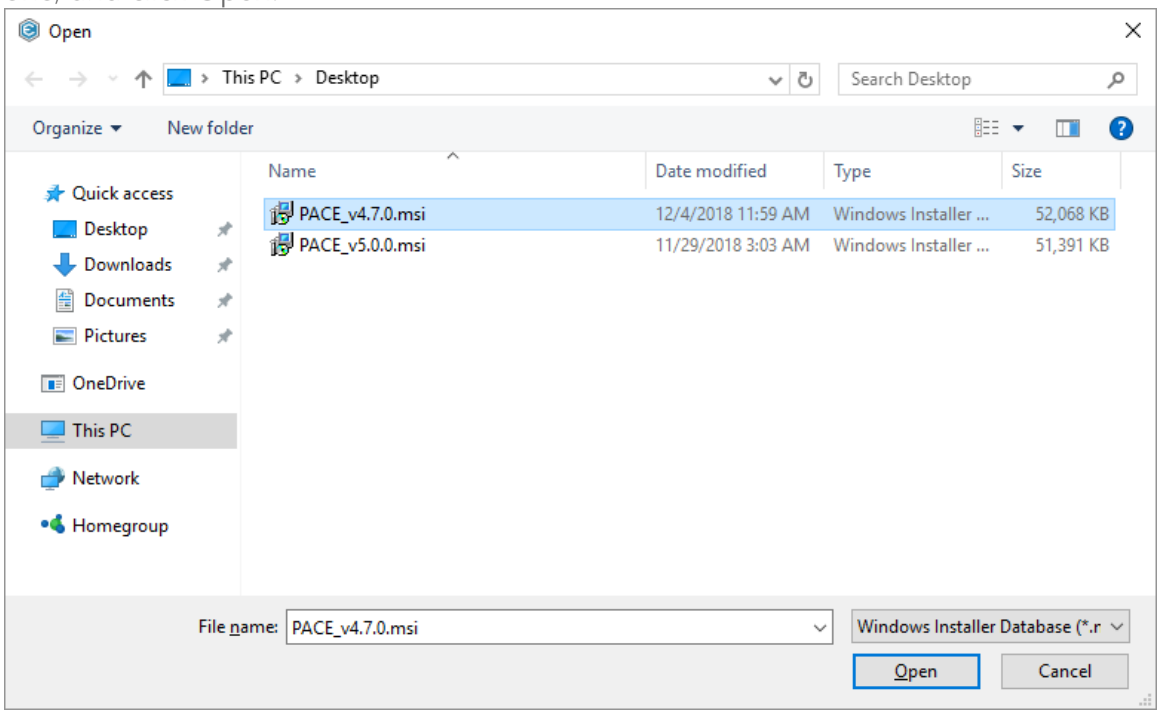
- Select Create MSP against original MSI... from MENU.



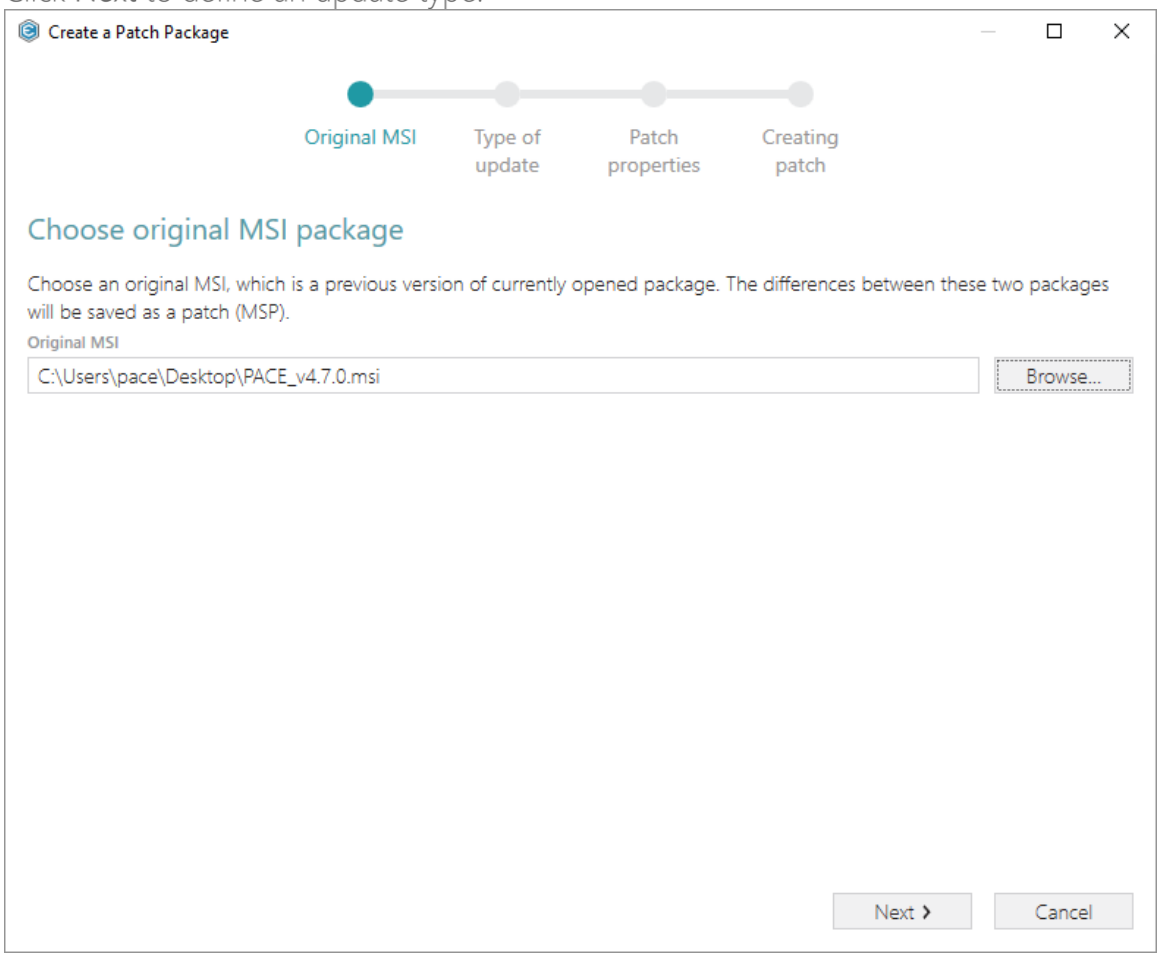
- Click Browse... to choose an original MSI package, which is a previous version of the currently opened one.



- Choose an original (old) MSI package, which is a previous version of the currently opened one, and click Open.

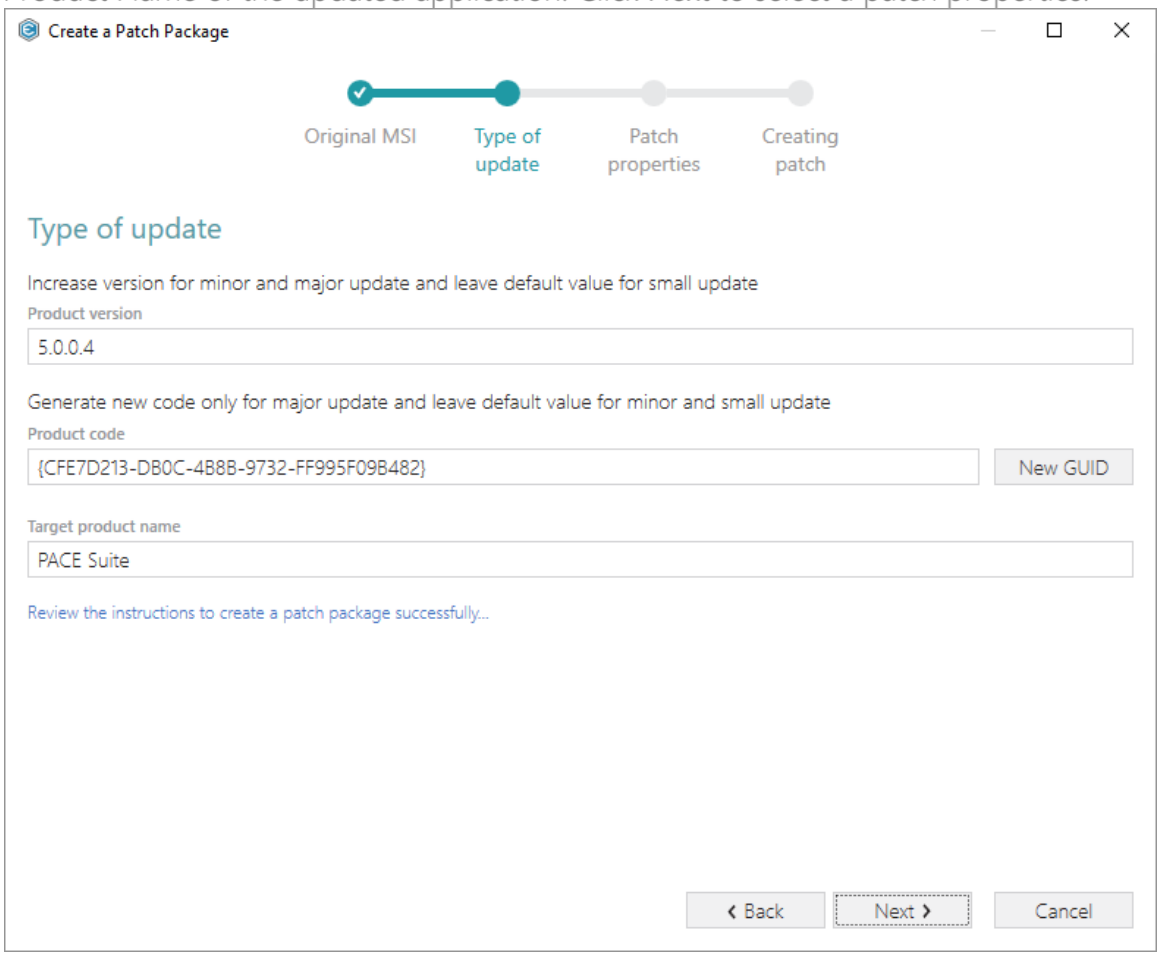


- Click Next to define an update type.

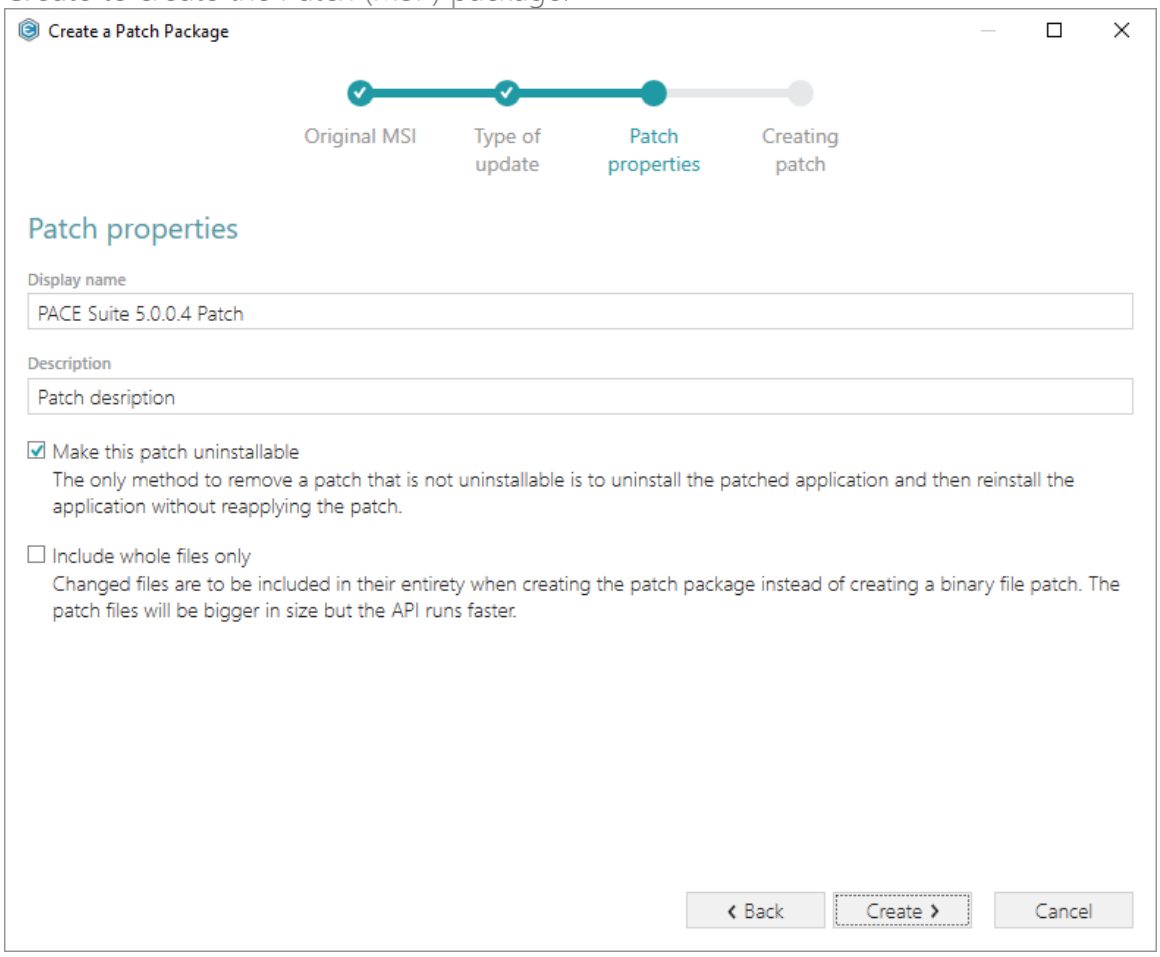


- Increase a Product version value for the minor and major update and leave default one for the small update. Generate new Product code only for the major update and leave default one for the minor and small update. If needed, update a Target product name in order to change the

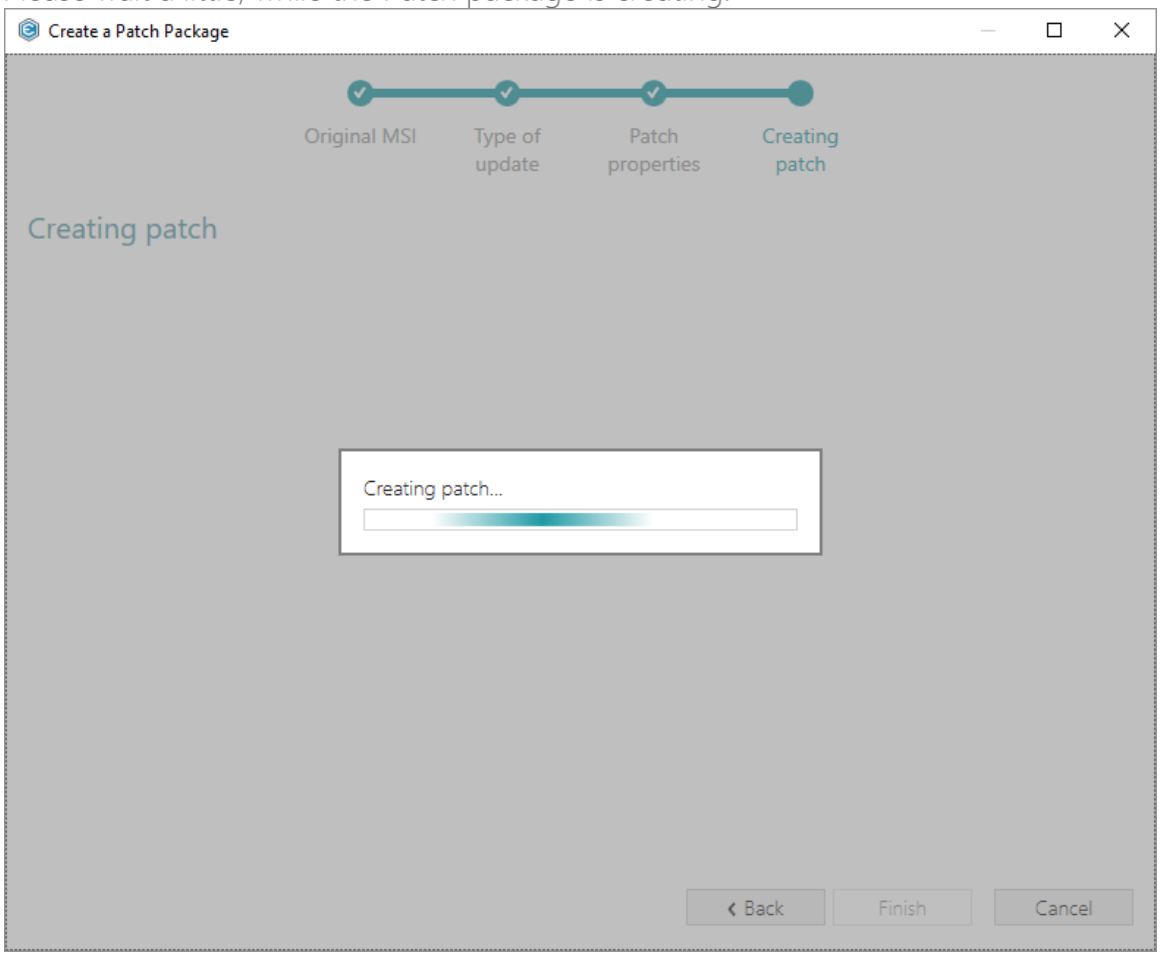
Product Name of the updated application. Click **Next** to select a patch properties.



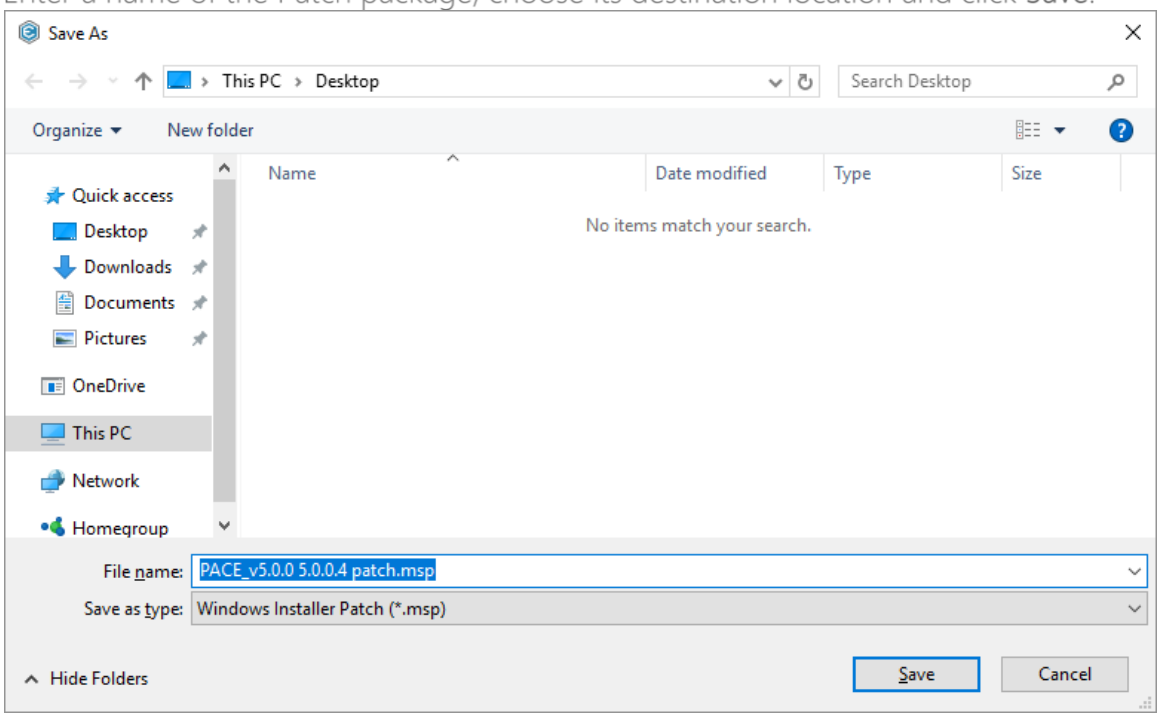
- Update Patch **Display name**, **Description** and select the necessary Patch properties. Click **Create** to create the Patch (MSP) package.



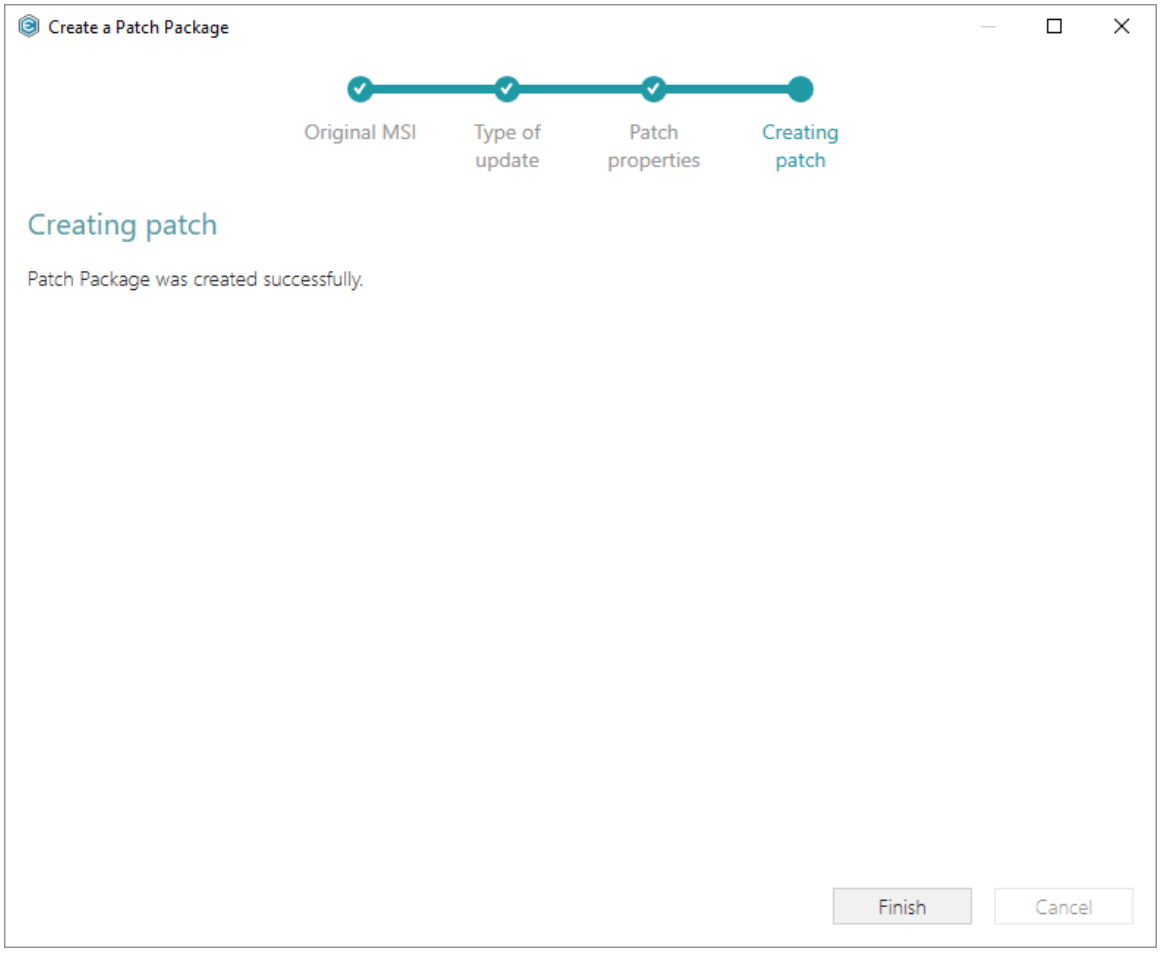
- Please wait a little, while the Patch package is creating.



- Enter a name of the Patch package, choose its destination location and click Save.



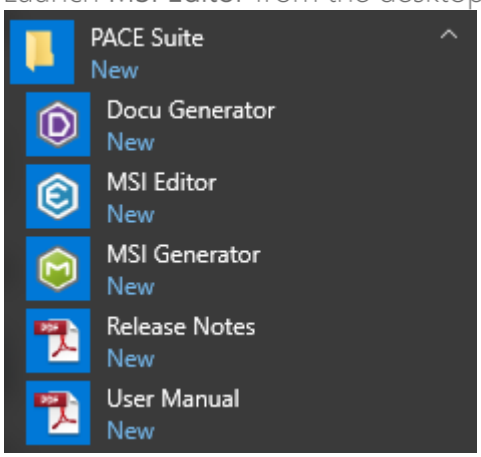
- Click **Finish** to close the wizard.



3.3.2 Save Changes as MSP

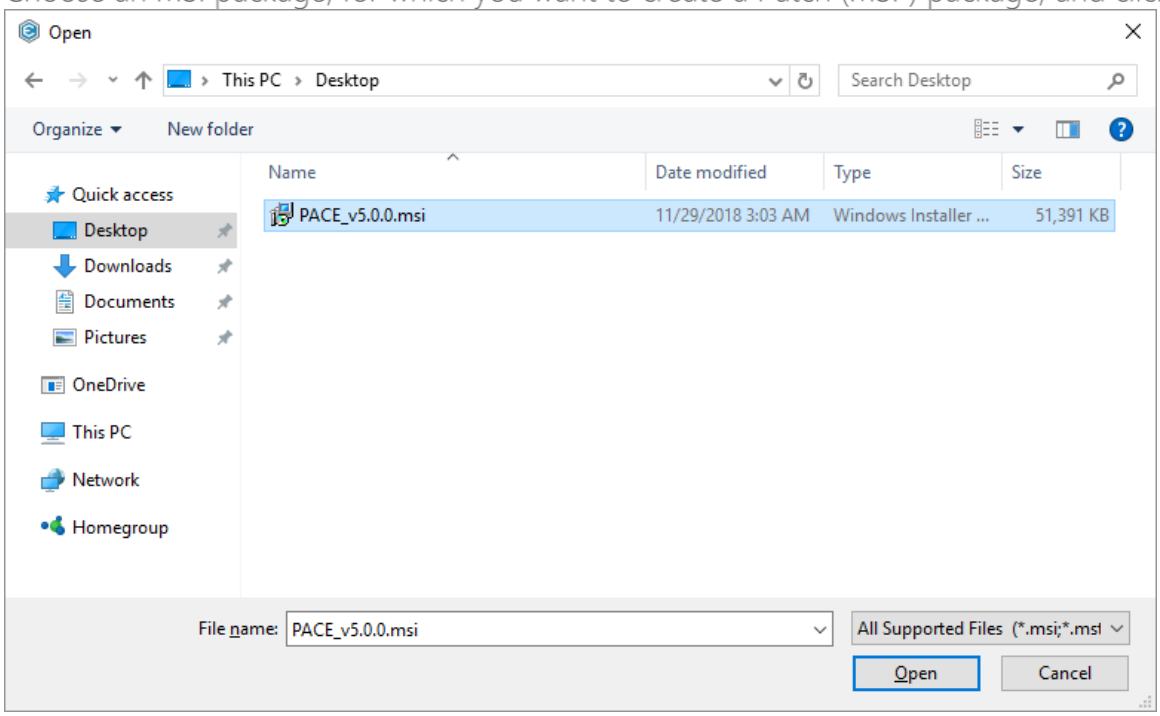
Save changes, made to the opened MSI database, as a Patch (MSP) package.

- Launch **MSI Editor** from the desktop or the start menu shortcut.



- Select Open... from MENU.

- Choose an MSI package, for which you want to create a Patch (MSP) package, and click Open.



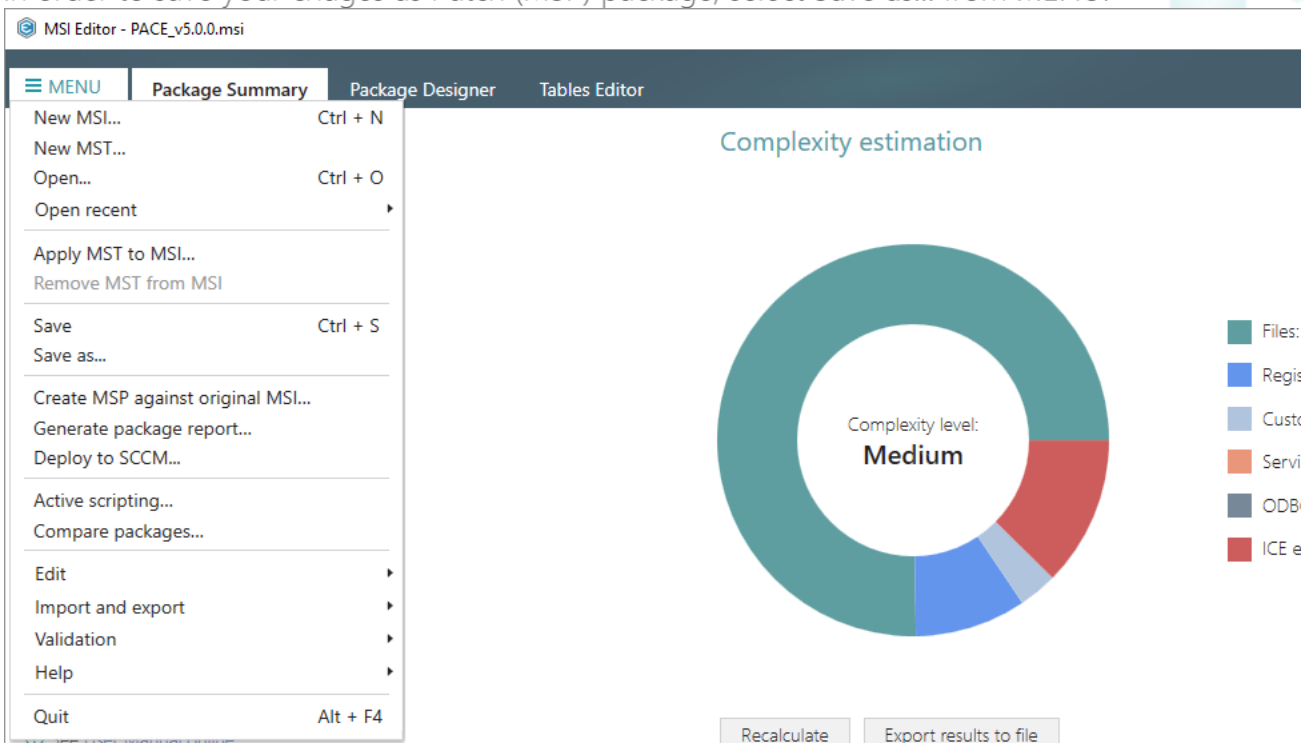
- Now you can start adding the necessary resources, which you want to include to the Patch (MSP) package. For managing package resources and settings, refer to the respective section of this document:

  - For managing Application Details see section 3.7.2
    Manage product name, version, language and manufacturer of the application.

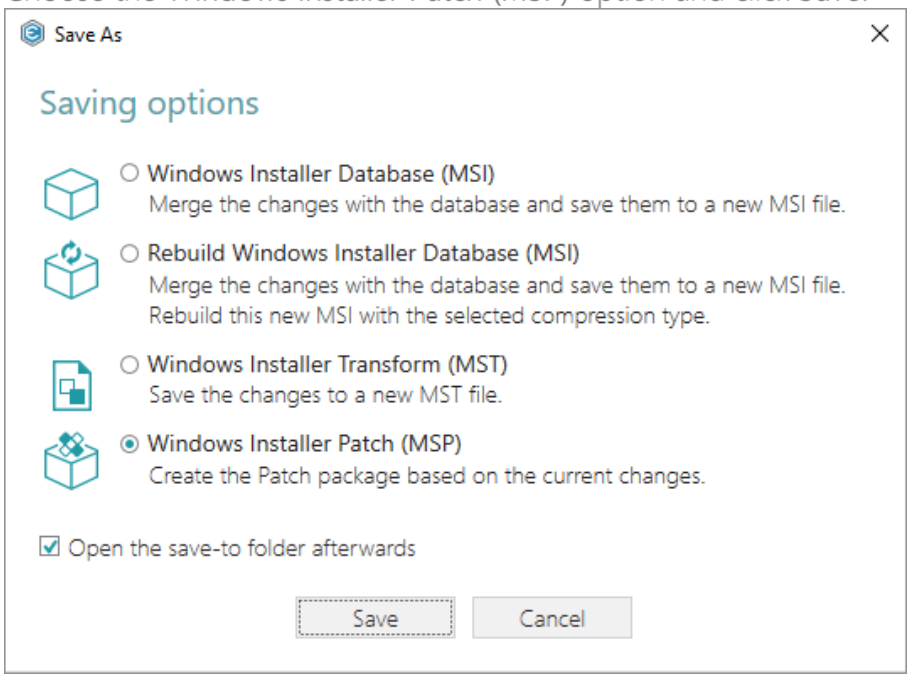
  - For managing Package IDs see section 3.7.3
    Manage product, upgrade and package codes.

- For managing Files and Folders see section 3.7.4
  Review package files and folders, add new folders, import files, set permissions, register COM objects and .NET assemblies, set the INSTALLDIR folder and the like.
- For managing Shortcuts see section 3.7.5
  Review package shortcuts, add new or import existing shortcuts, manage their properties.
- For managing Registry see section 3.7.6
  Review package registry, add new or import from REG file registry keys and values, integrate ActiveSetup, set permissions and the like.
- For managing Custom Actions see section 3.7.7
  Manage custom and standard actions of the package, add new one to run scripts, EXE, etc.
- For managing Drivers see section 3.7.8
  Review the package device drivers, modify and add new ones.
- For managing Features and Components see section 3.7.9
  Review the feature-component structure and resources, linked with components. Add new features, components and manage their properties.
- For managing ARP Appearance see section 3.7.10
  Control how the package will be displayed in the App list (Add & Remove Programs).
- For managing Upgrades see section 3.7.11
  Manage the package upgrade settings to automatically upgrade previous MSI packages.
- For managing MSI Properties see section 3.7.12
  Manage package properties, defined in the Property table and those, which will be updated by the CustomAction, AppSearch, Control, ComboBox, CheckBox, RadioButton tables.
- For managing Summary Information see section 3.7.13
  Manage Summary Information of the opened MSI/MST.
- For managing MSI Tables see section 3.7.14
  Add, import, drop MSI tables.
- For managing Merge Modules, described in section 3.7.15
  Create new MSM or importing existing MSM to the package.

- In order to save your chages as Patch (MSP) package, select Save as... from MENU.

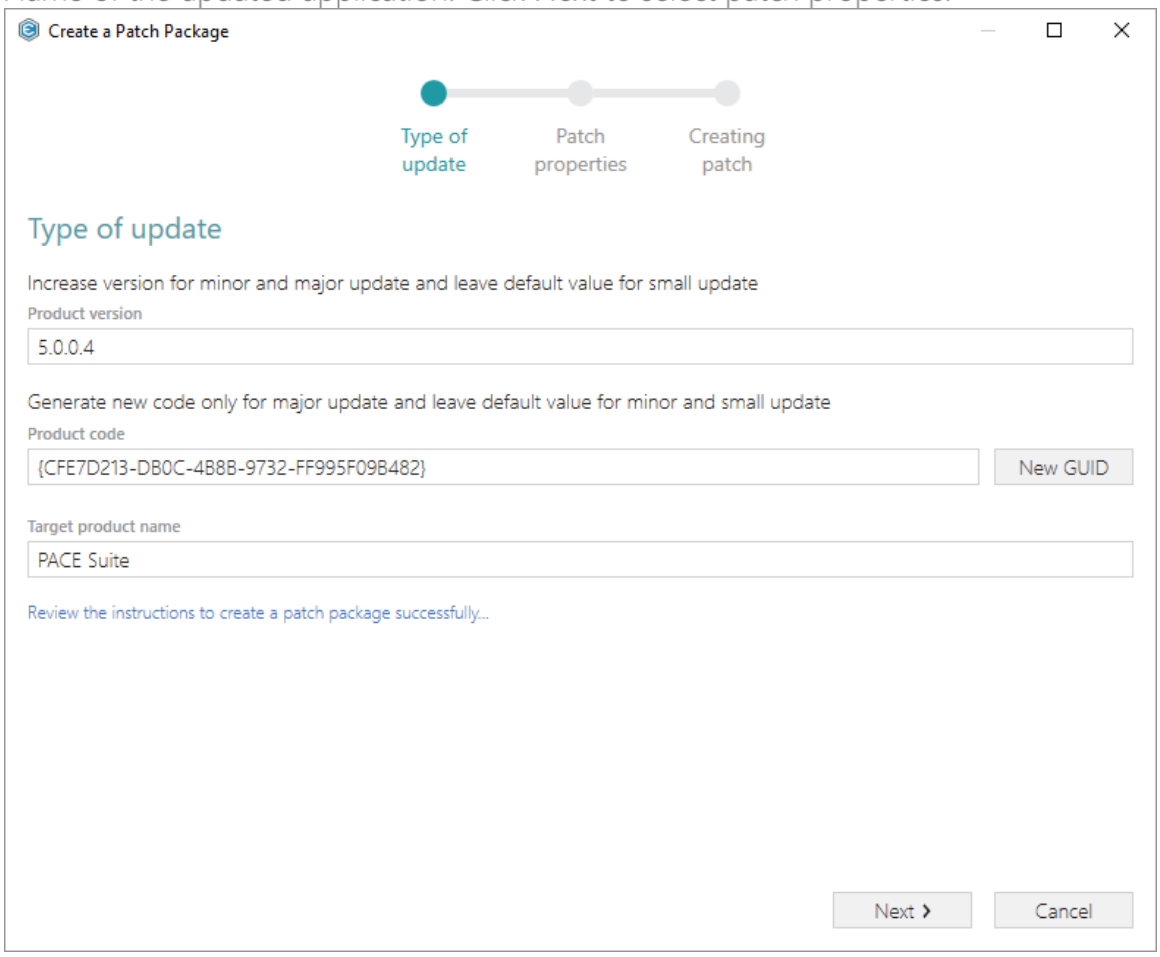


- Choose the Windows Installer Patch (MSP) option and click Save.

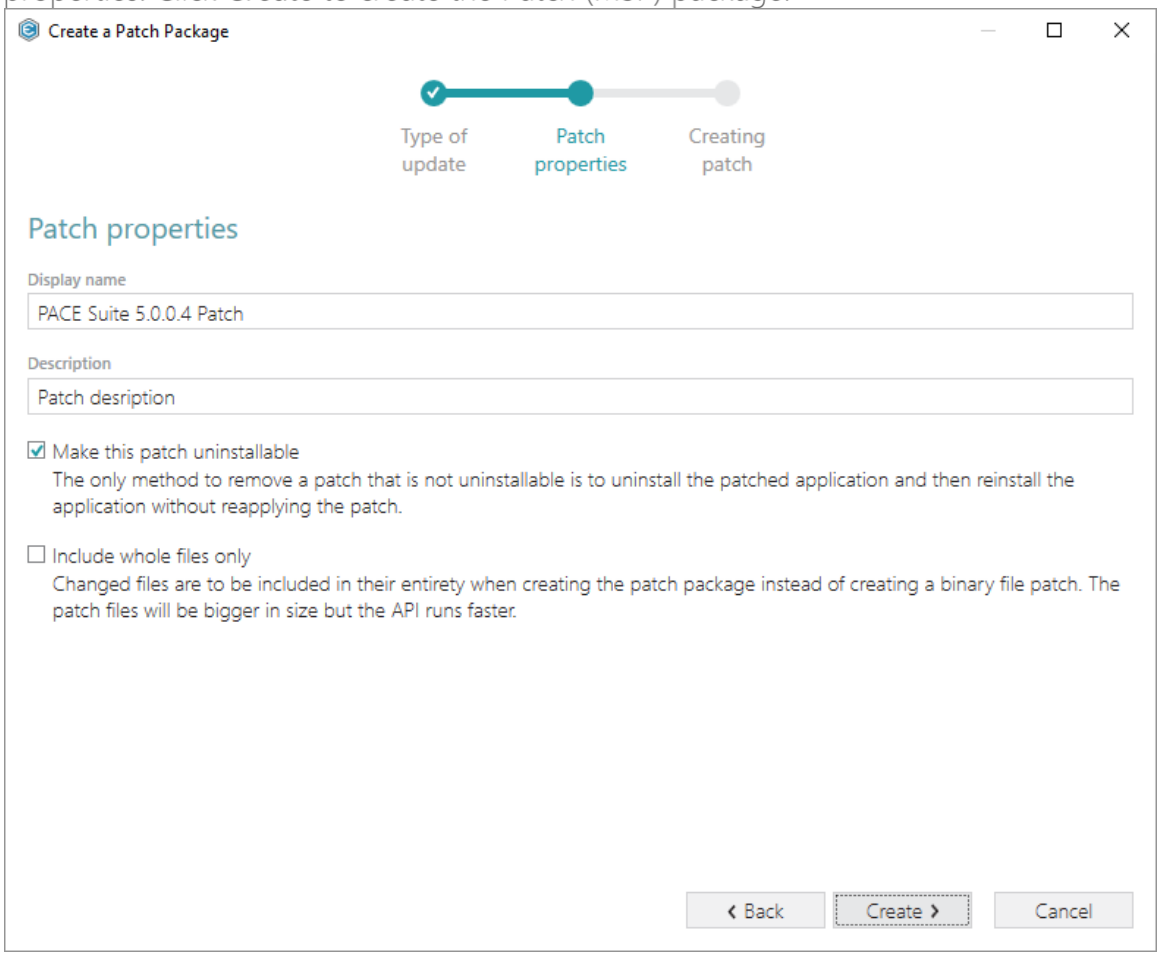


- Increase a Product version for the minor and major update and leave default for the small update. Generate new Product code only for the major update and leave default for the minor and small update. If needed, update a Target product name in order to change the Product

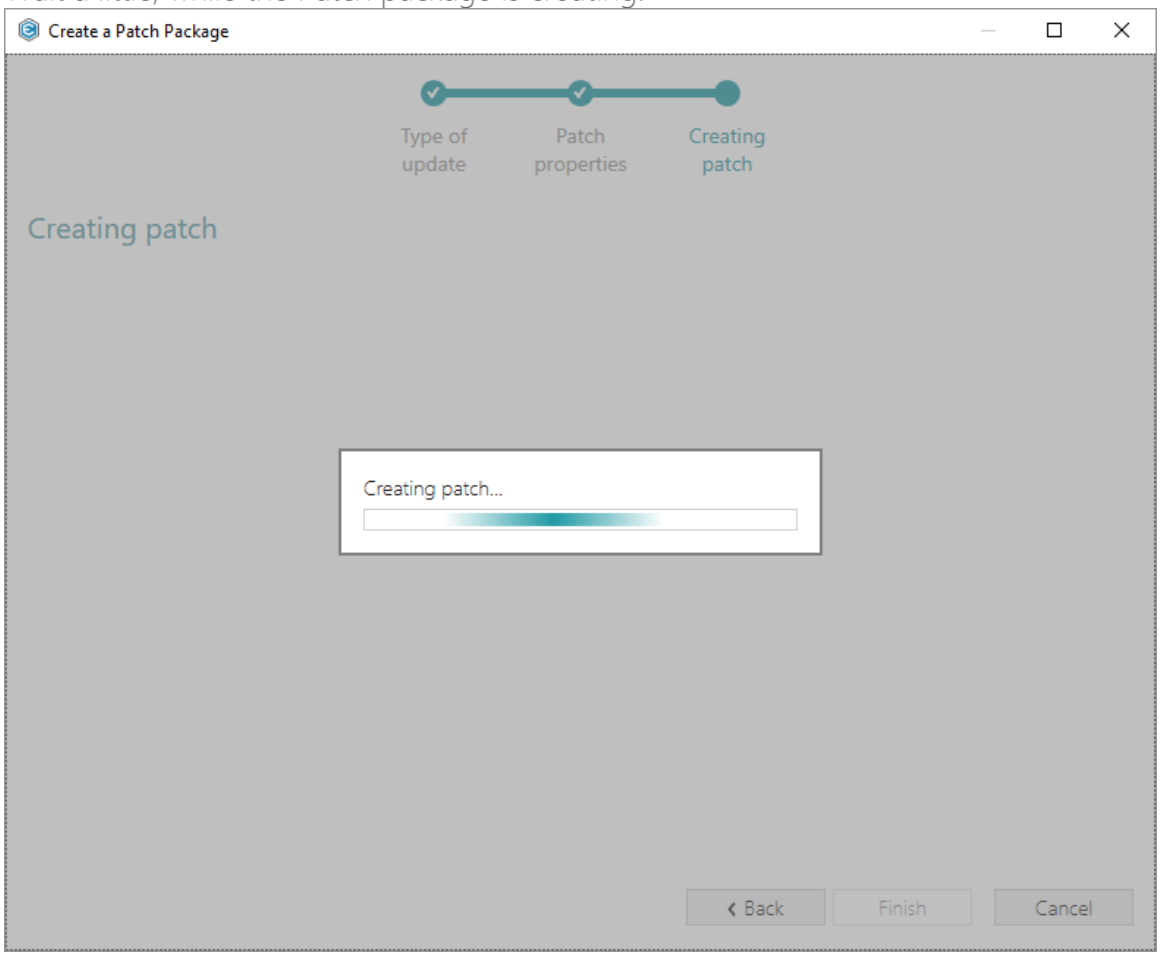
Name of the updated application. Click **Next** to select patch properties.



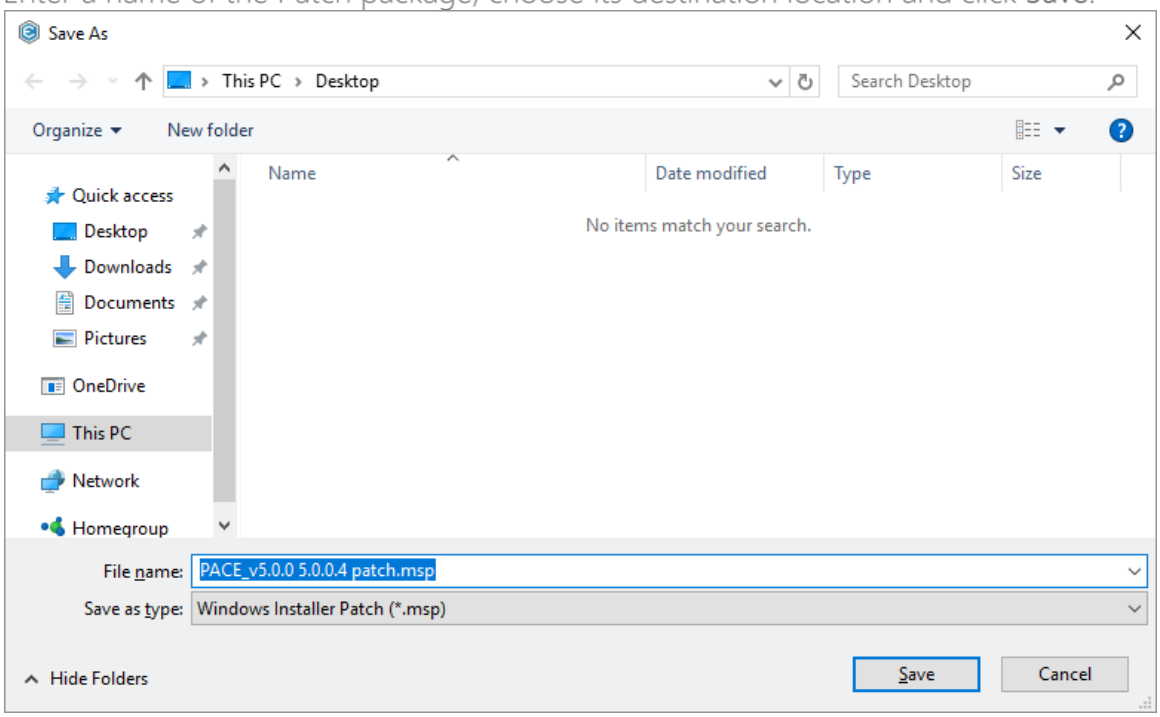
- Update **Display name** and **Description** of the Patch package and select the necessary properties. Click **Create** to create the Patch (MSP) package.



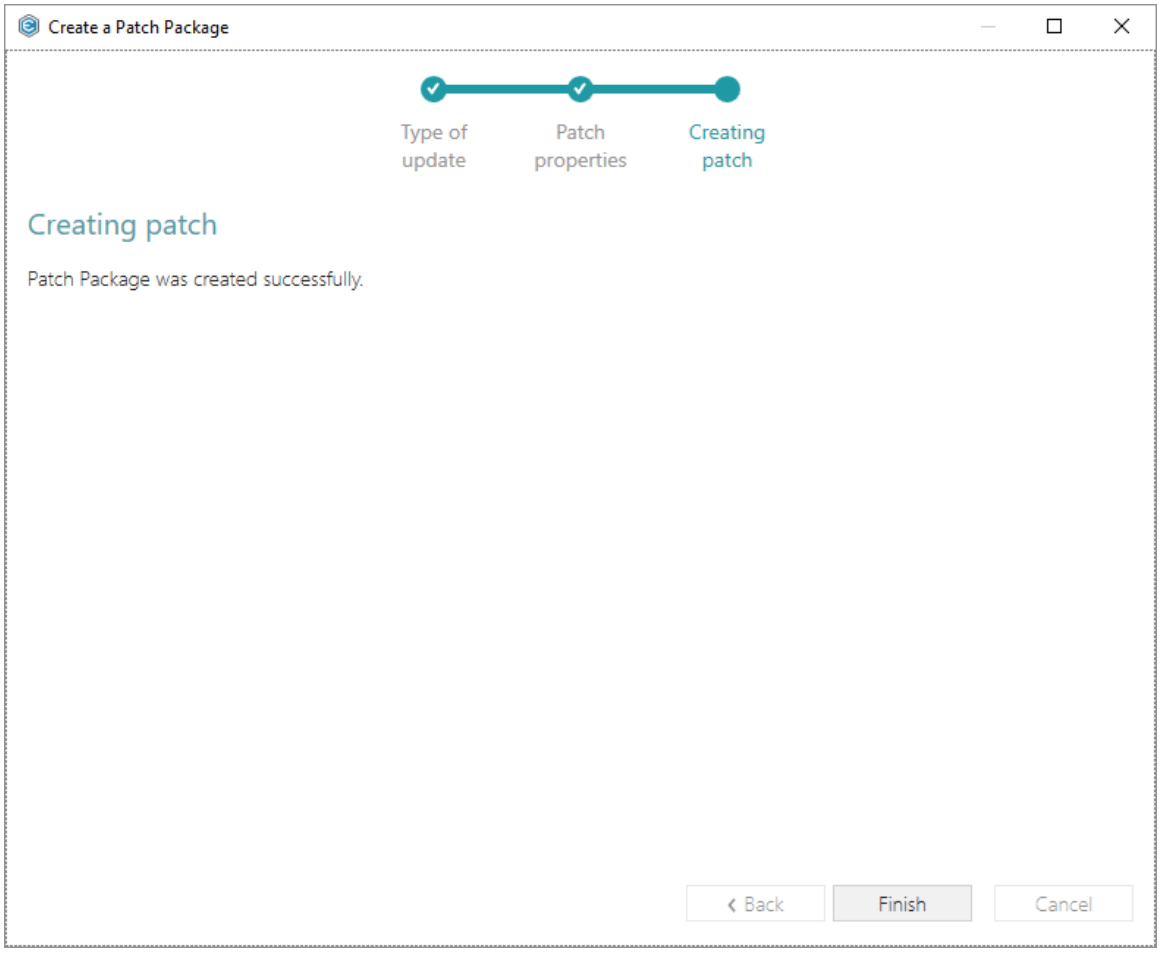
- Wait a little, while the Patch package is creating.



- Enter a name of the Patch package, choose its destination location and click Save.



- Click **Finish** to close the wizard.



3.4 Create MSIX/APPX

NOTE Building Universal Windows Platform app packages (MSIX, APPX) is available on Windows 10 or Windows Server 2016 or newer.

NOTE Before packaging you app to MSIX/APPX, review UWP application limitations.

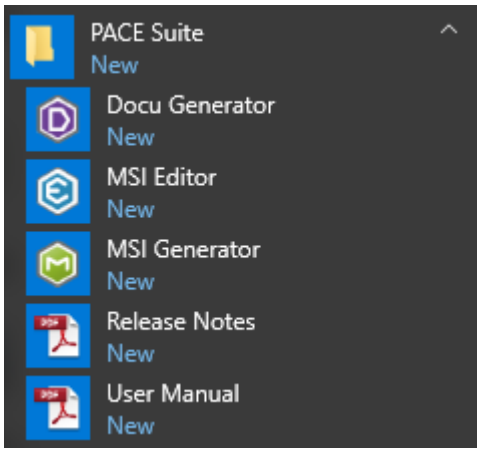
Choose a scenario that better suits your needs:

- Create MSIX/APPX from Scratch, described in section 3.4.1
  Create a blank project in MSI Generator, add files, registry, select package settings and build MSIX/APPX package.
- Repackage Installation to MSIX/APPX, described in section 3.4.2
  Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into MSIX/APPX package using MSI Generator.

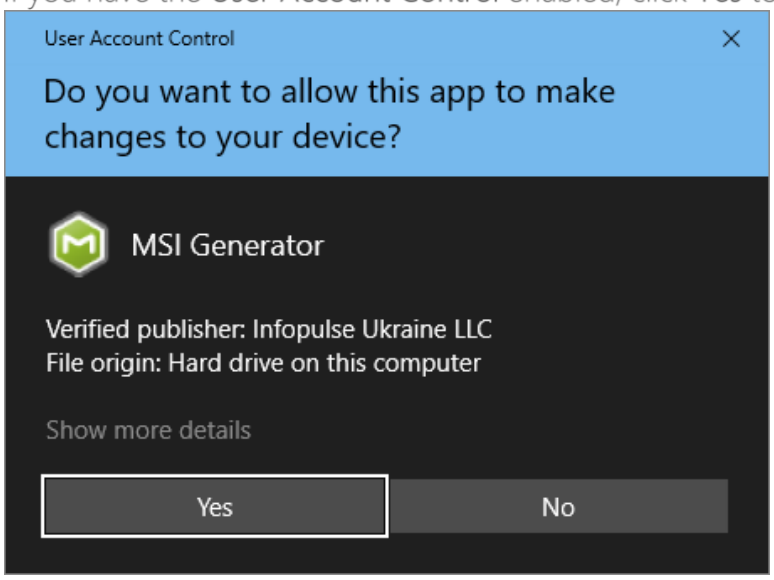
3.4.1 Create MSIX/APPX from Scratch

Create a blank project in MSI Generator, add files, registry, select package settings and build MSIX/APPX package.

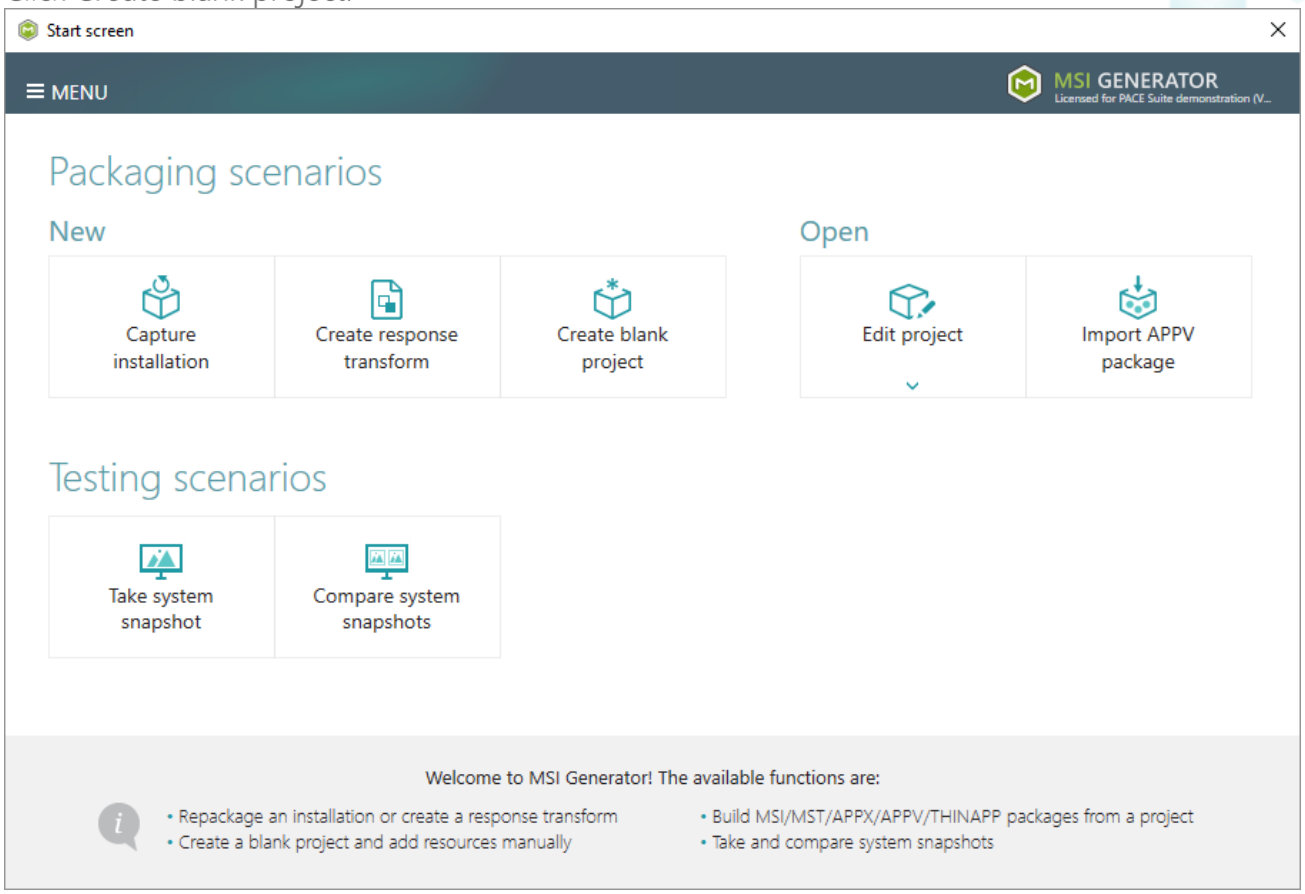
- Launch MSI Generator from the desktop of the start menu shortcut.



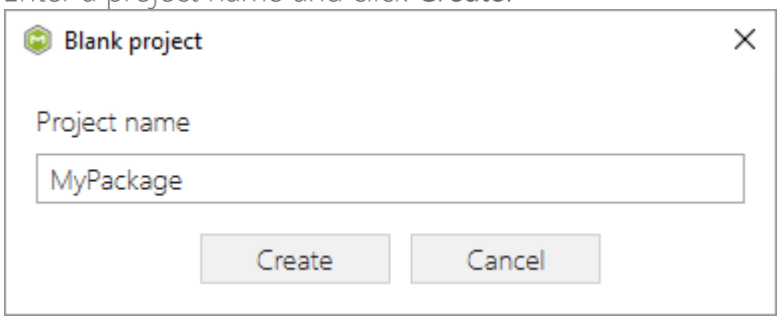
- If you have the User Account Control enabled, click Yes to launch the application.



- Click Create blank project.

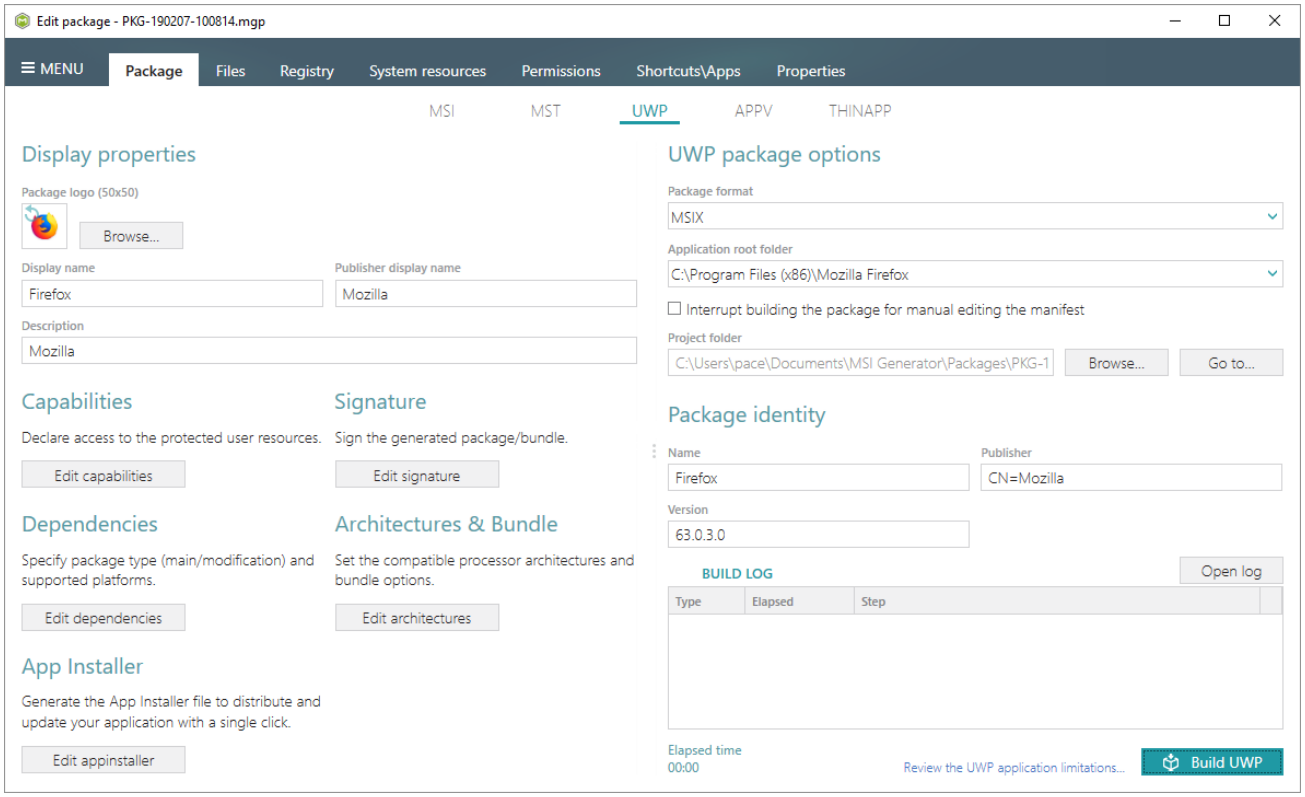


- Enter a project name and click Create.

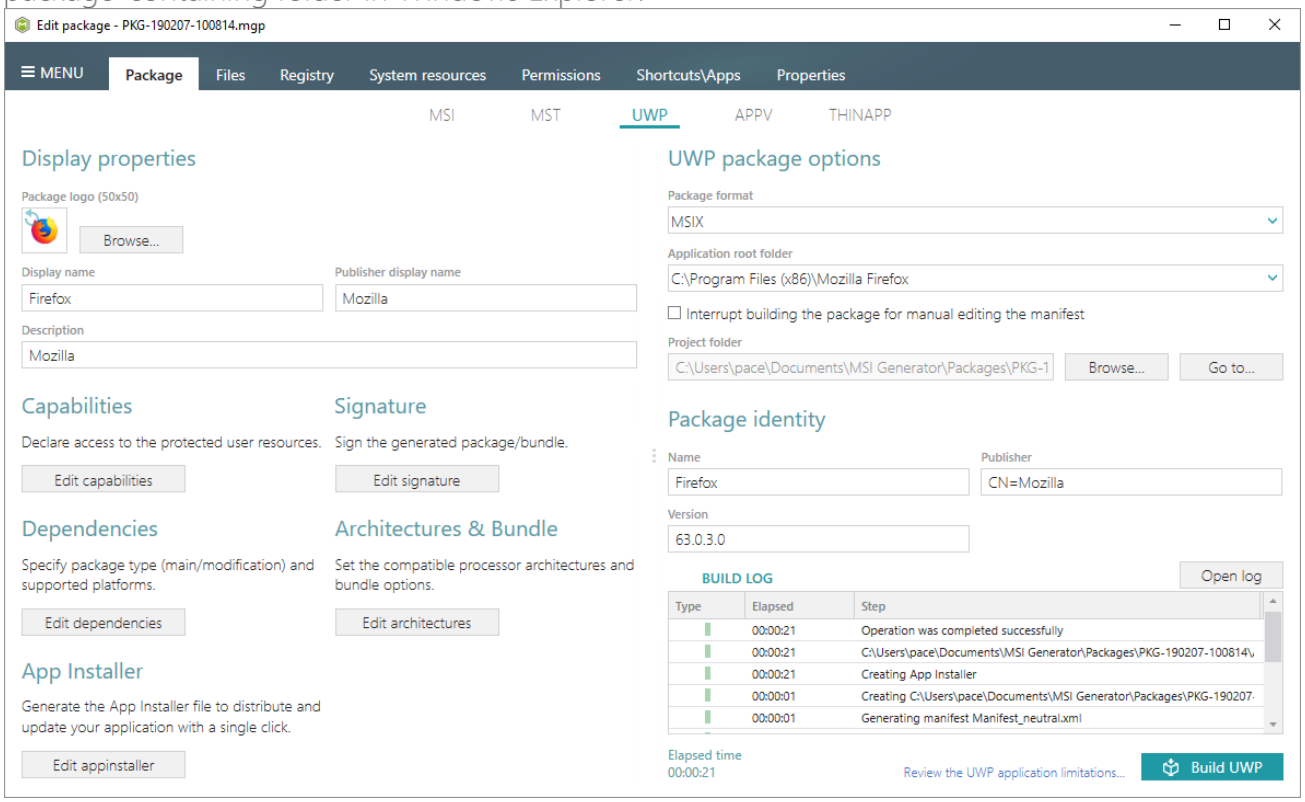


- For managing project resources such as files, folders, and registry refer to the respective section of this document:
    - Regarding File and Folders see section 3.8.2
    - Regarding Registry see section 3.8.3
- For managing MSIX/APPX Settings such as display and identity properties, apps and their visual assets, capabilities, signature, dependencies, bundle, AppInstaller, see section 3.8.7

- Finaly, to generate an MSIX/APPX package navigate to the **Package** -> **UWP** tab and click **Build UWP**.



- Once the package is built, click **Go to...**, located next to the **Project folder** field, to open the package-containing folder in Windows Explorer.



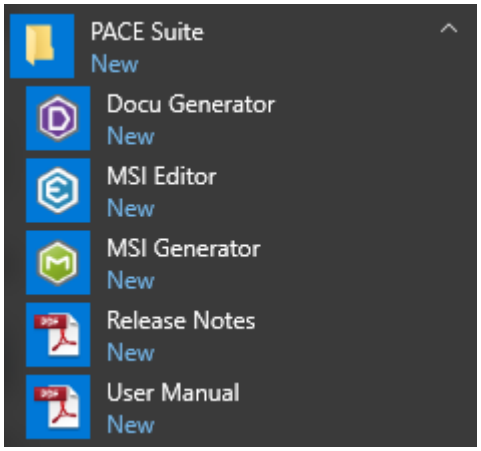
- To install the generated UWP package, (1) enable the Developer Mode or Sideloading apps on the target system and (2) ensure that the certificate, used for signing, is trusted on that system. If you are using the self signed certificate, install it manually to the Trusted People or Trusted Publishers local machine certificate store. By adding a certificate to local machine certificate stores, you affect the certificate trust of all users on the computer. It is recommended that you

remove those certificates when they are no longer necessary to prevent them from being used to compromise system trust.

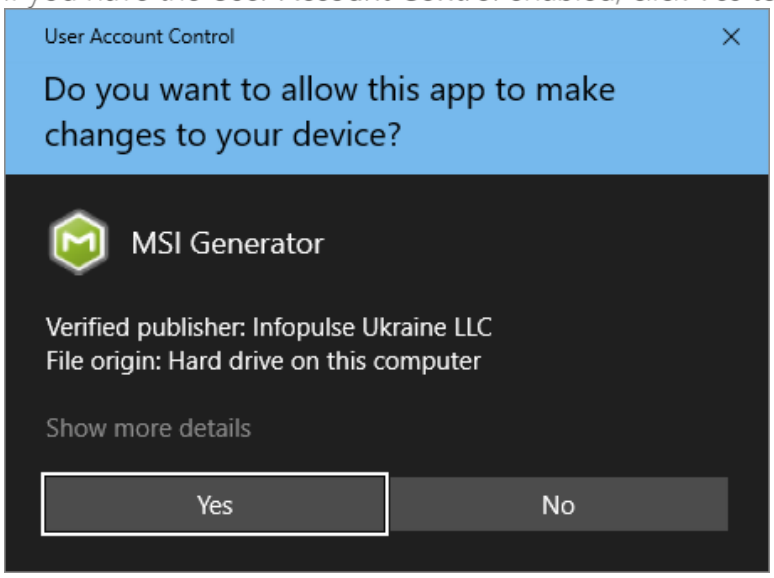
3.4.2 Repackage Installation to MSIX/APPX

Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into MSIX/APPX package using MSI Generator.

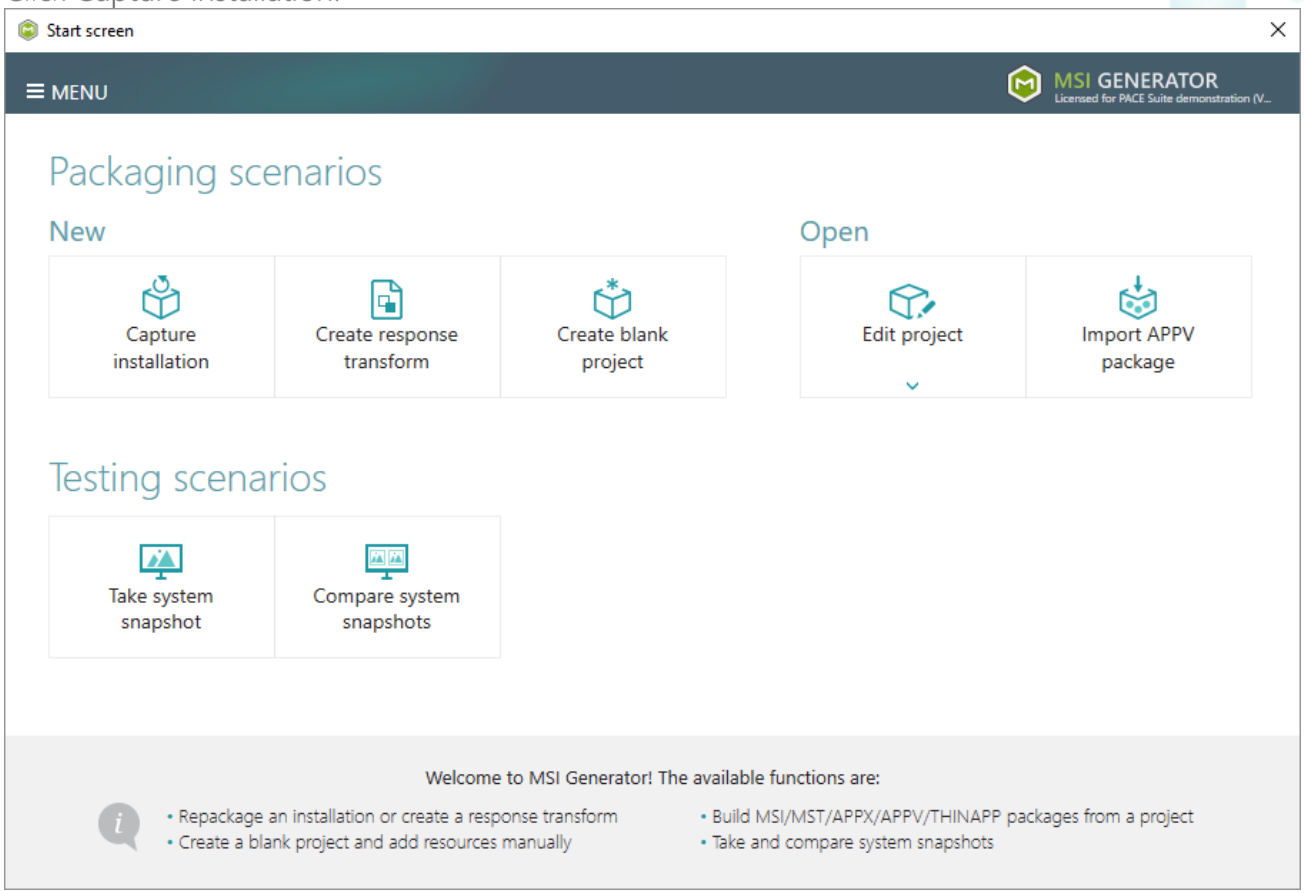
- Launch MSI Generator from the desktop of the start menu shortcut.



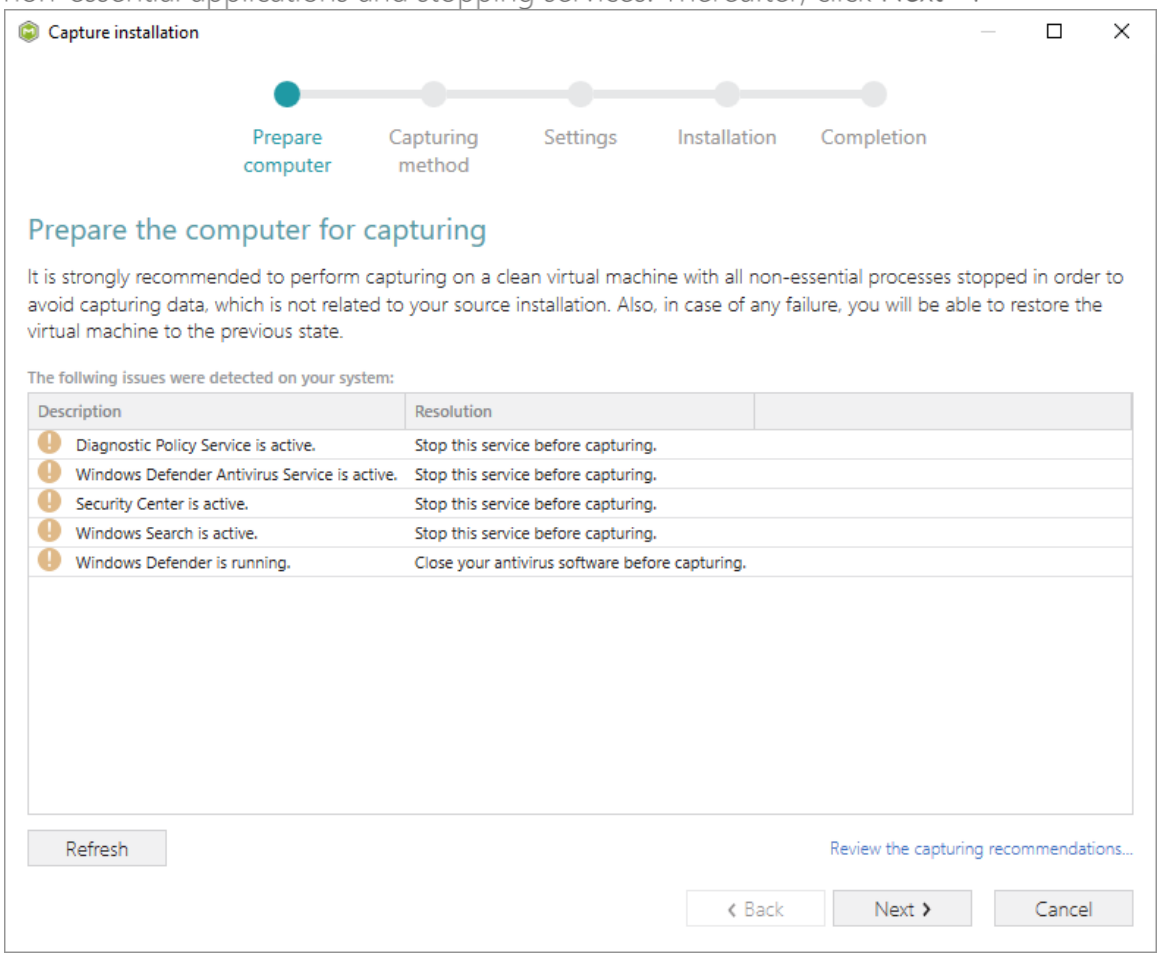
- If you have the User Account Control enabled, click Yes to launch the application.



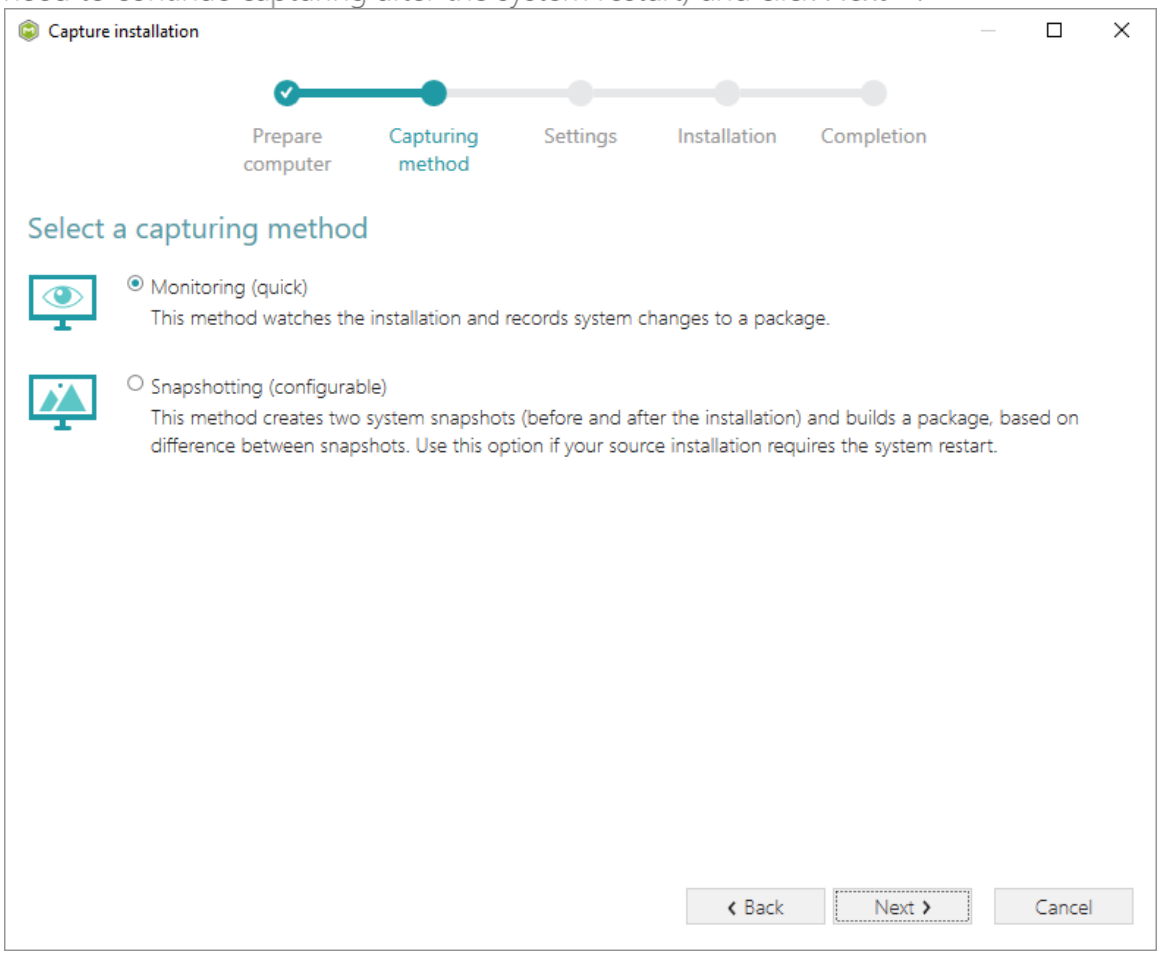
- Click Capture installation.



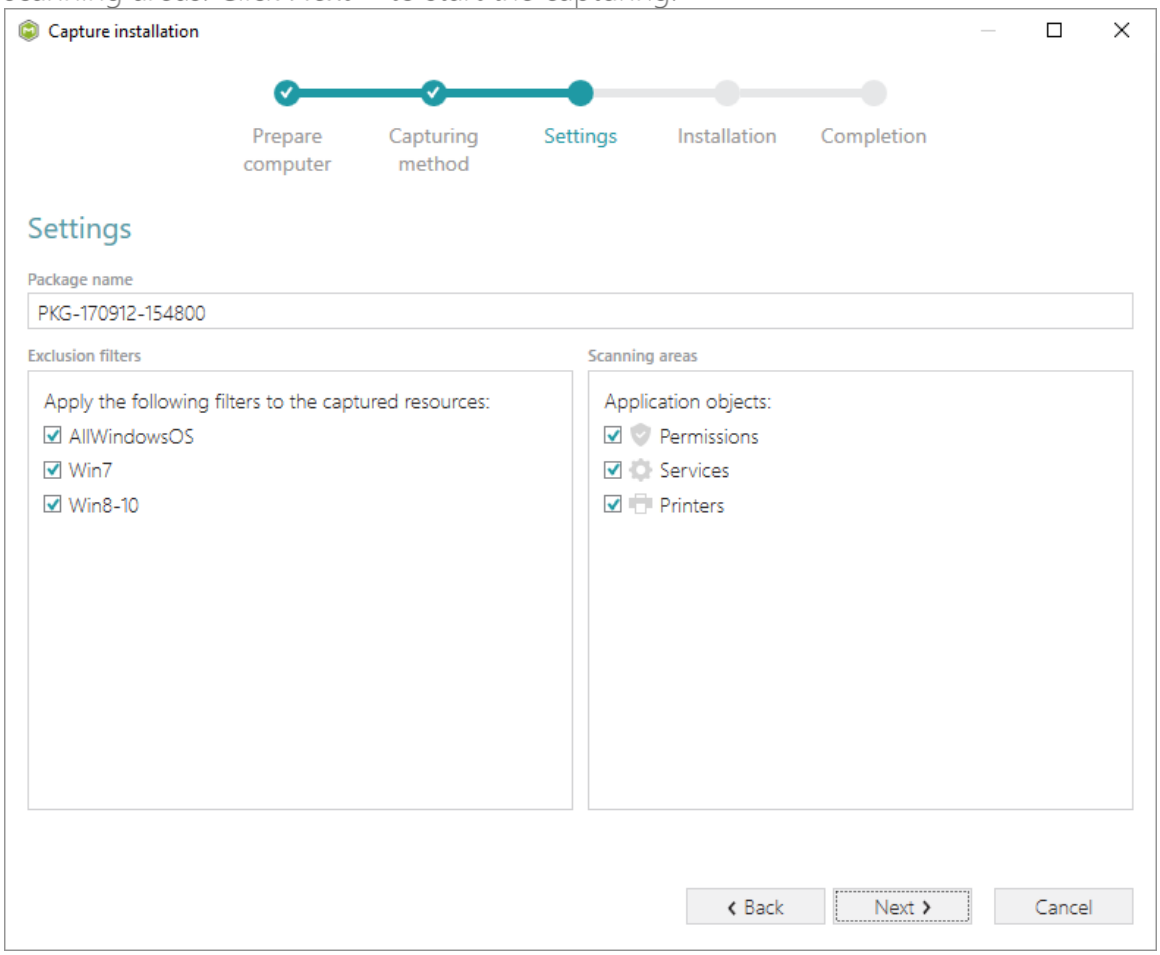
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



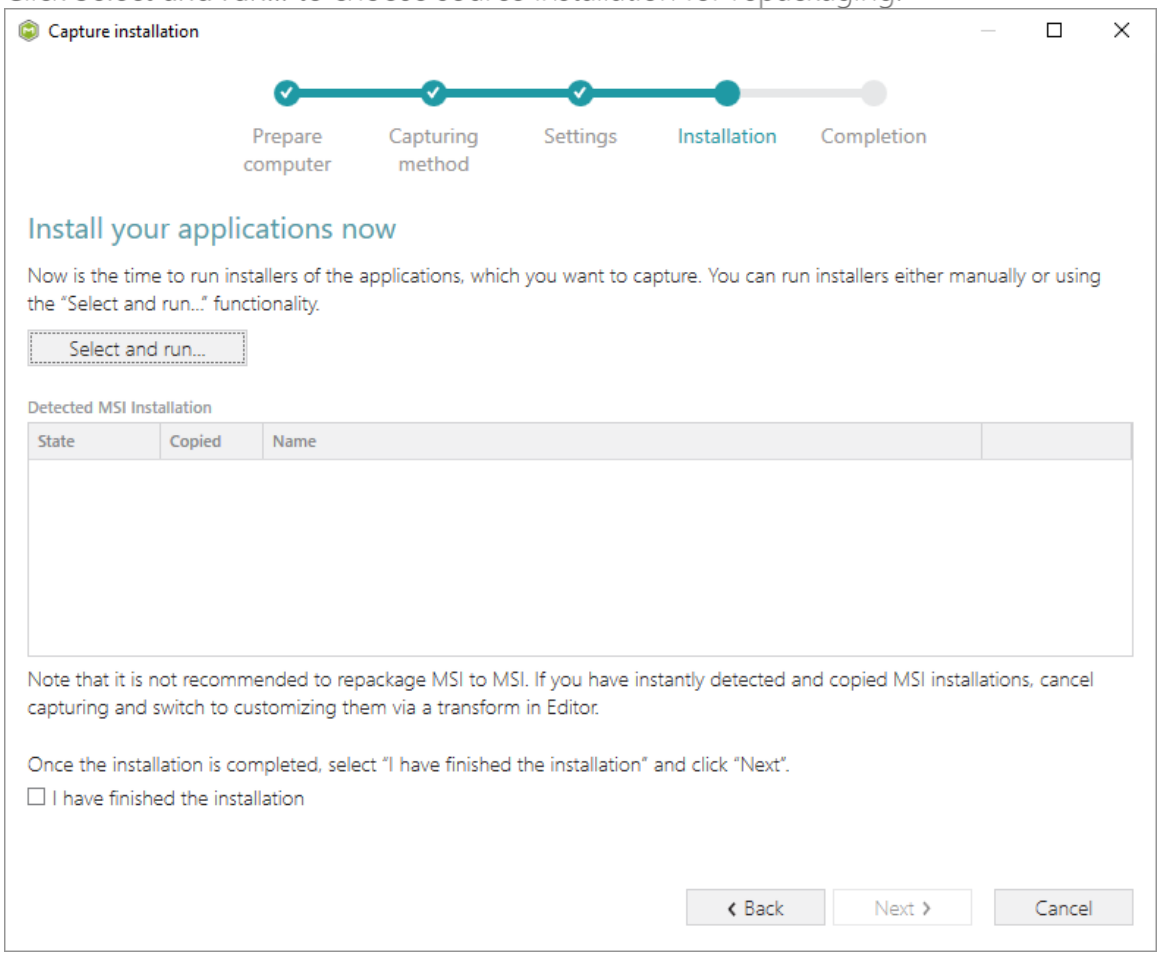
- Select the **Monitoring** method for the quicker capturing (or use the **Snapshotting** one if you need to continue capturing after the system restart) and click **Next >**.



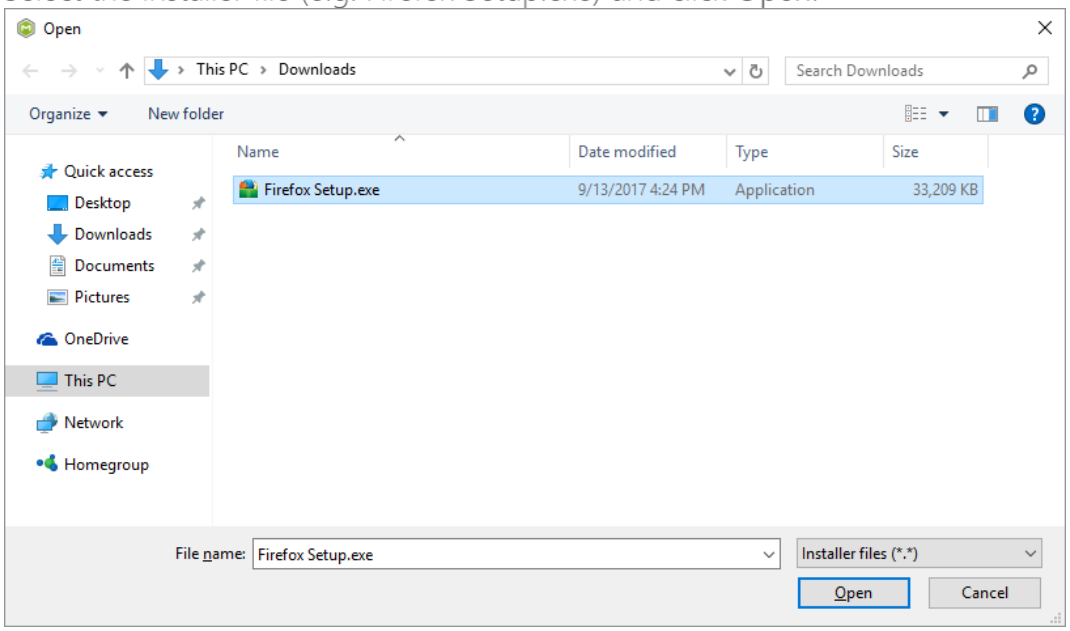
- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.



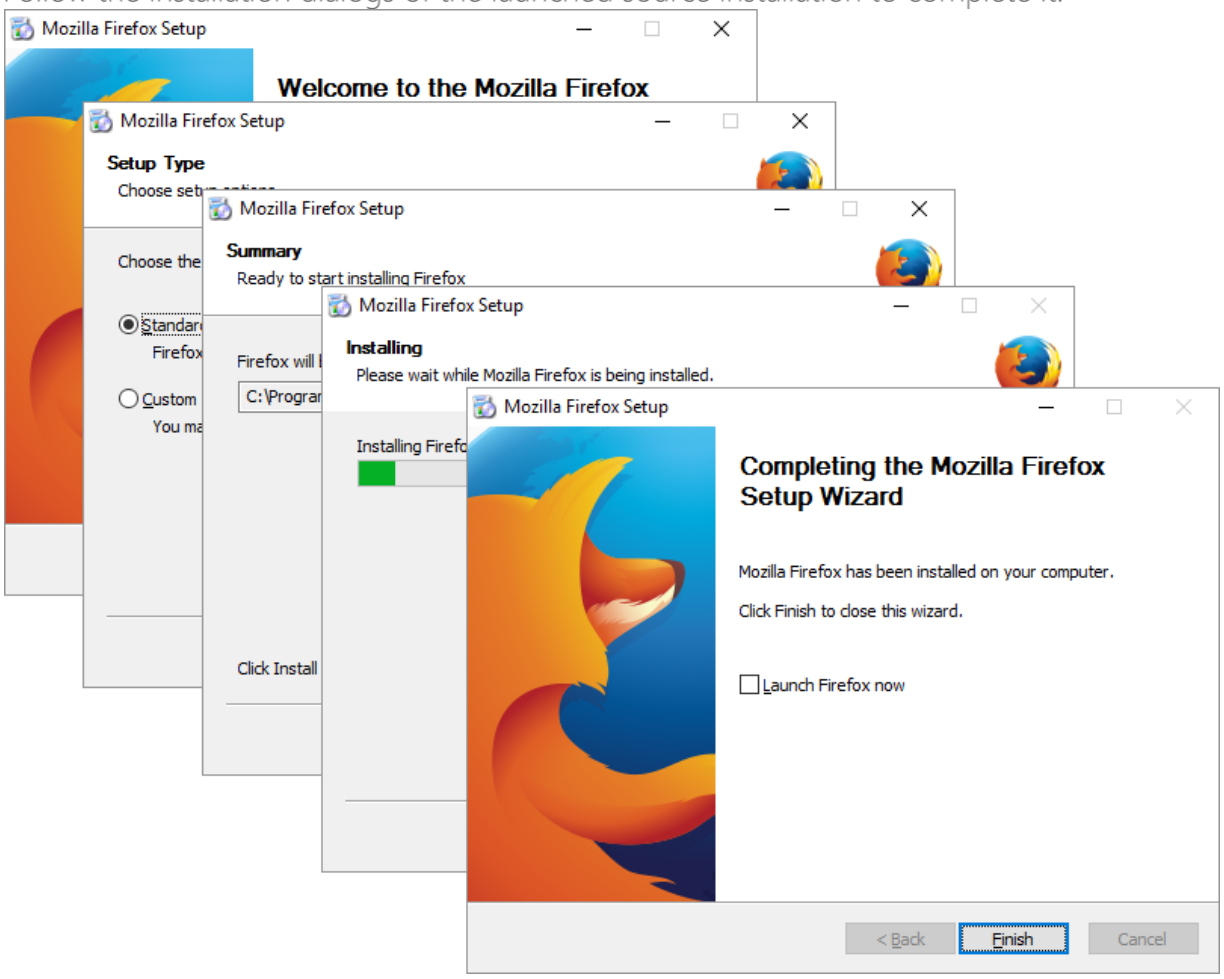
- Click Select and run... to choose source installation for repackaging.



- Select the installer file (e.g. Firefox Setup.exe) and click Open.

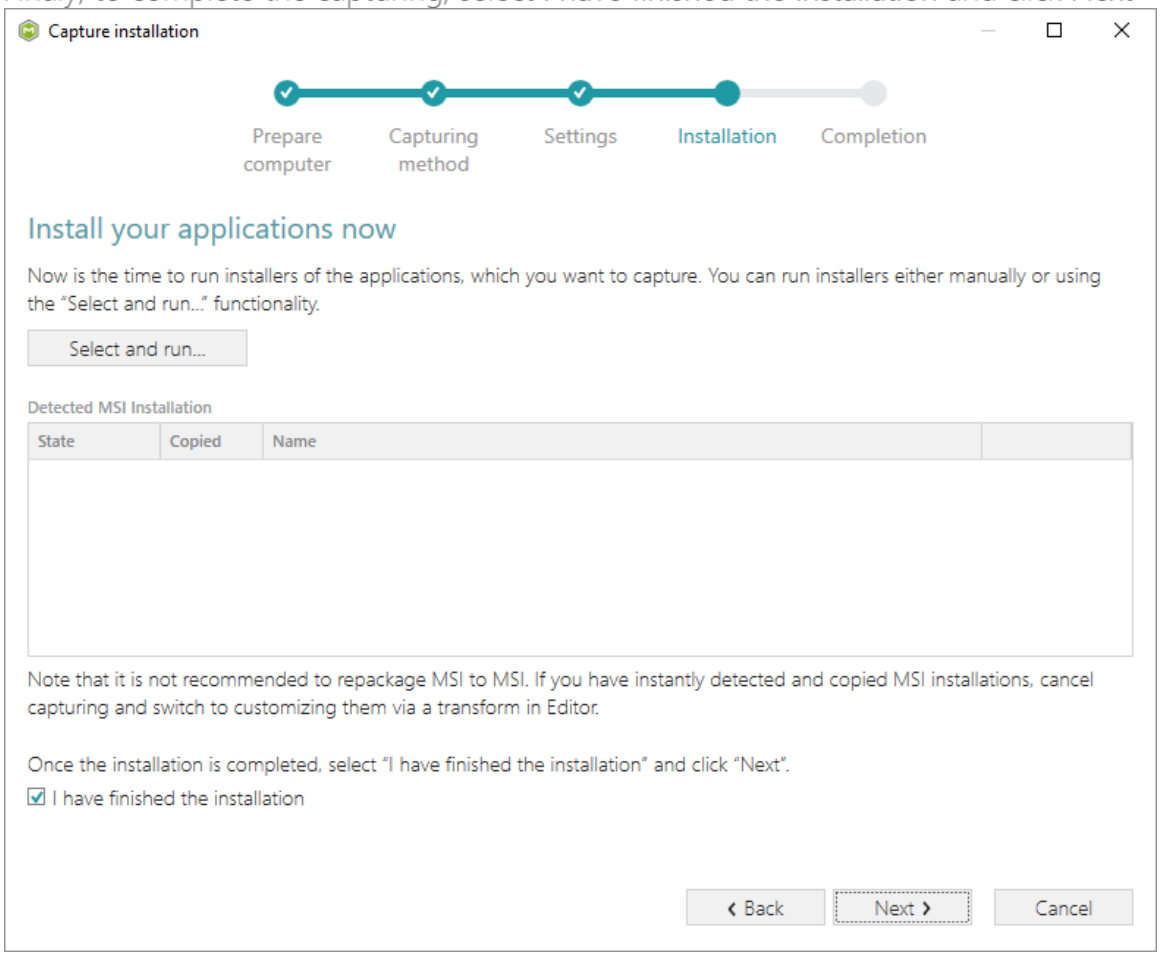


- Follow the installation dialogs of the launched source installation to complete it.

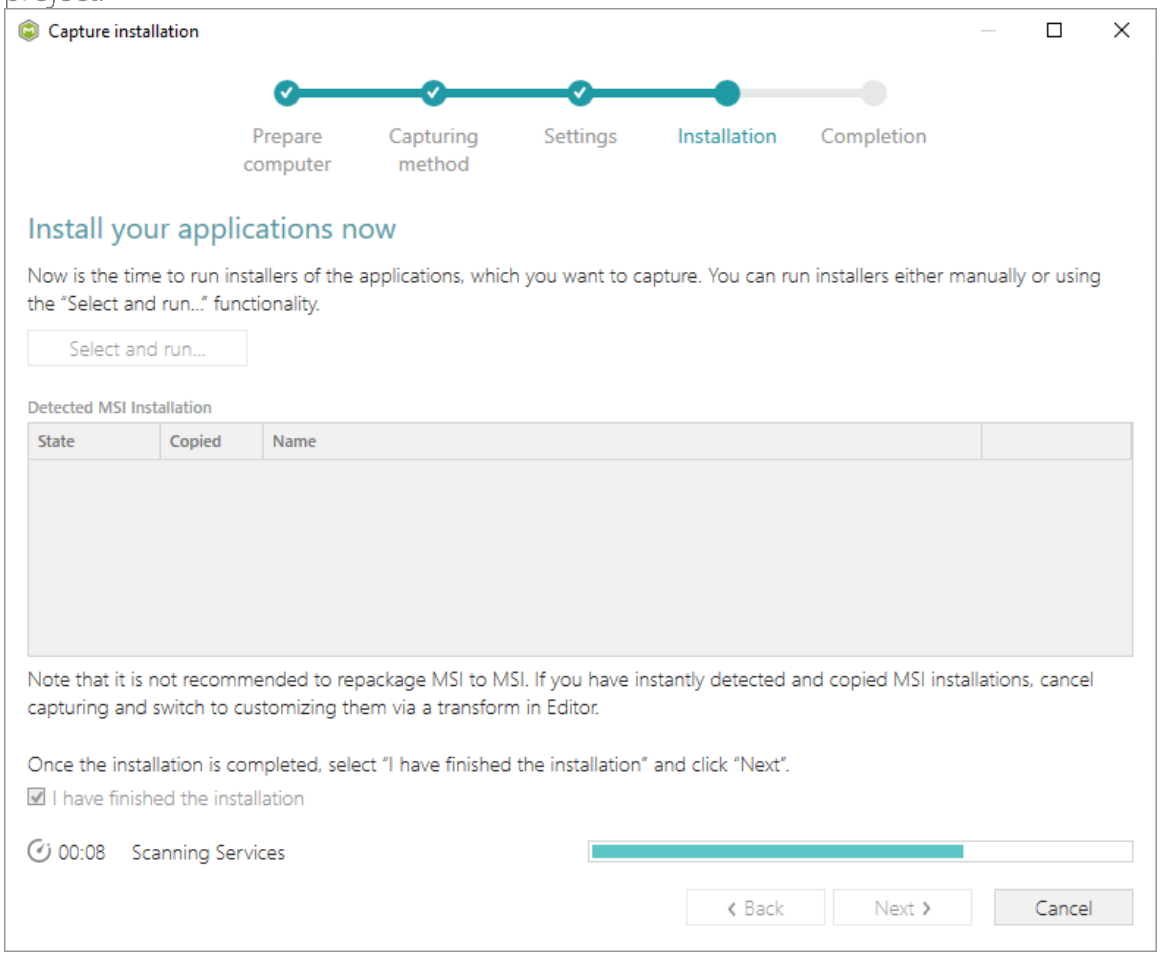


- Now you can make any additional changes to the file system and registry, which you want to capture along with source installation and include to the package. For instance, you can copy or edit configuration files, import REG file to the system registry, change file, folder permissions, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.

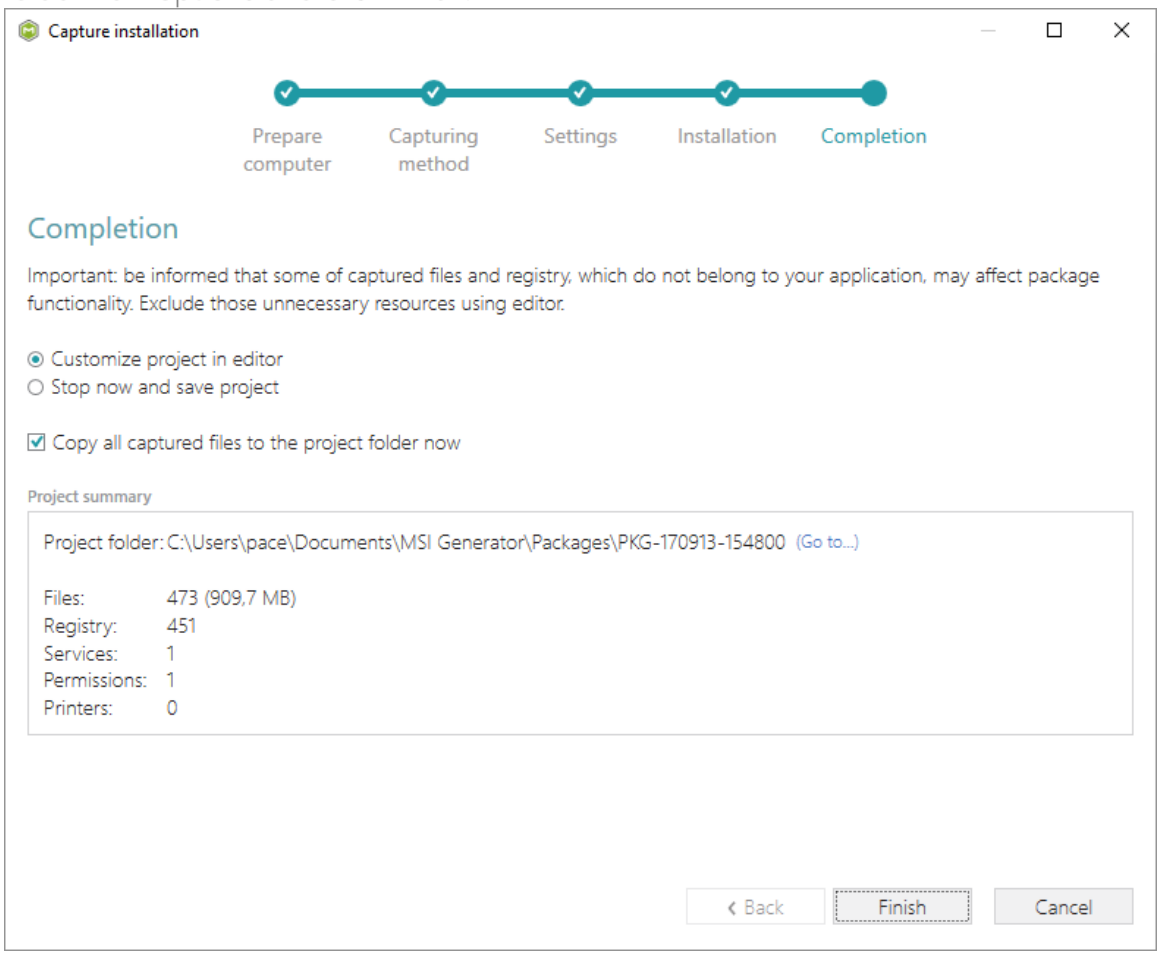
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



- Wait a little, while the capturing process is finishing, filtering captured data and creating the project.

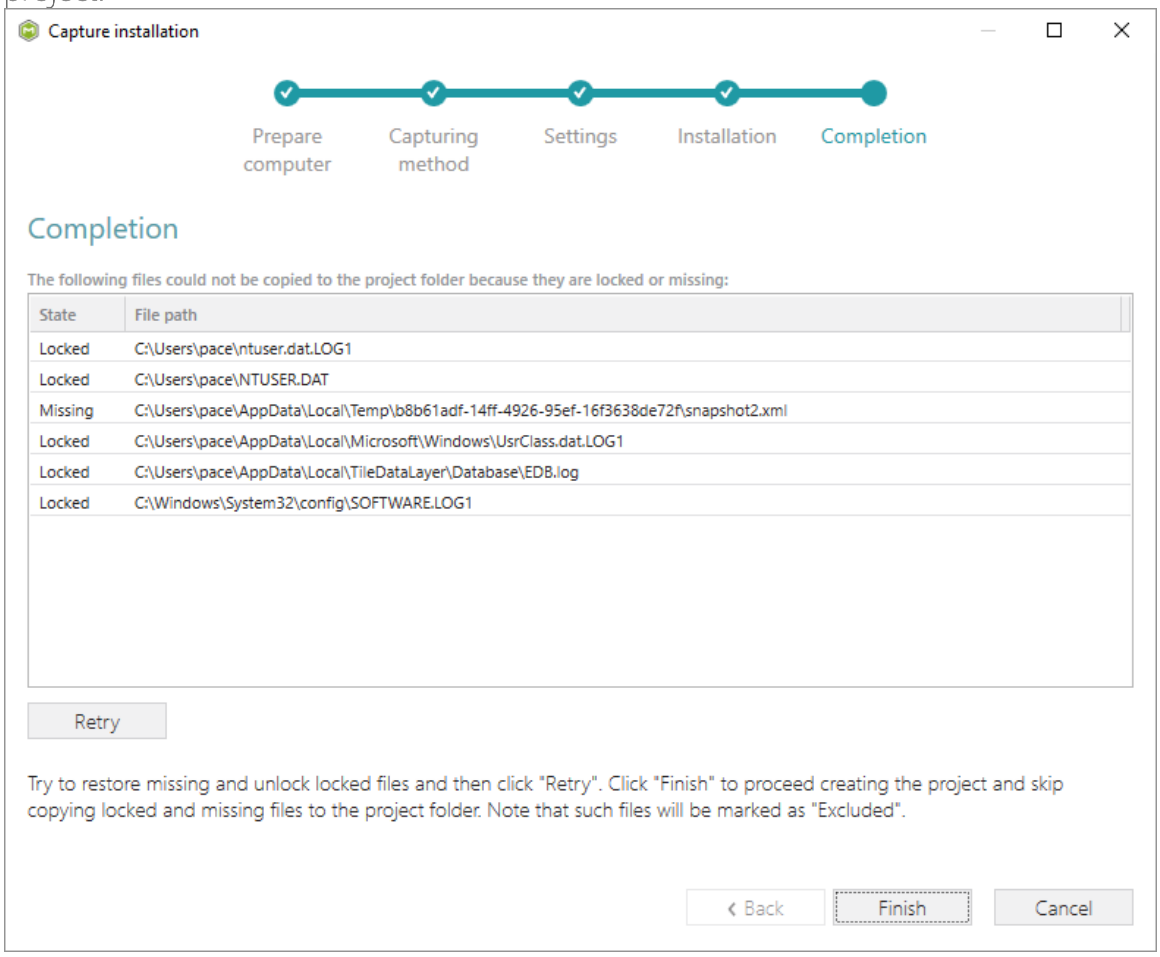


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



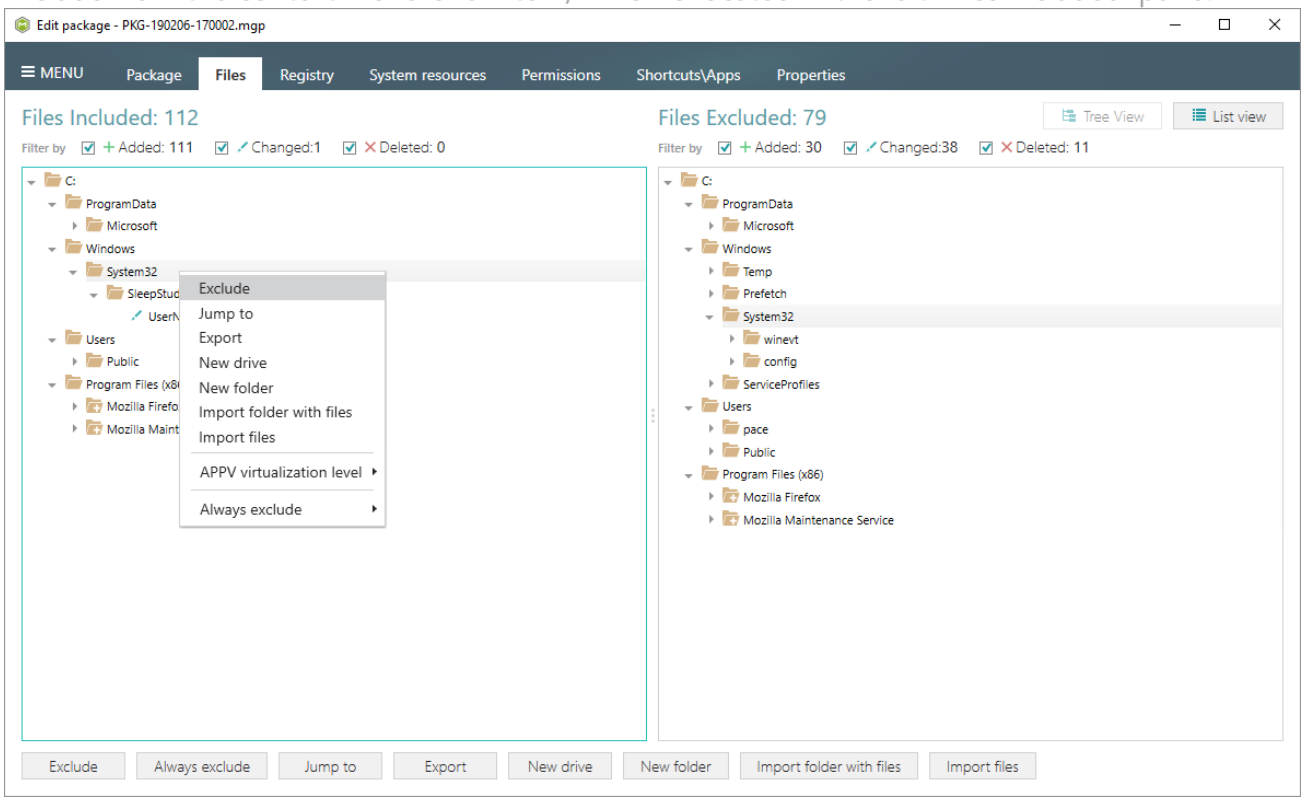
- If some of captured files are locked or missing and as result they could not be copied to the project folder, they will be displayed as illustrated below. Usually, the displayed files are system noise and do not refer to the captured application. Click Finish to continue with skiping locked and missing files. Note that such files will be automatically marked as 'Excluded' in your

project.

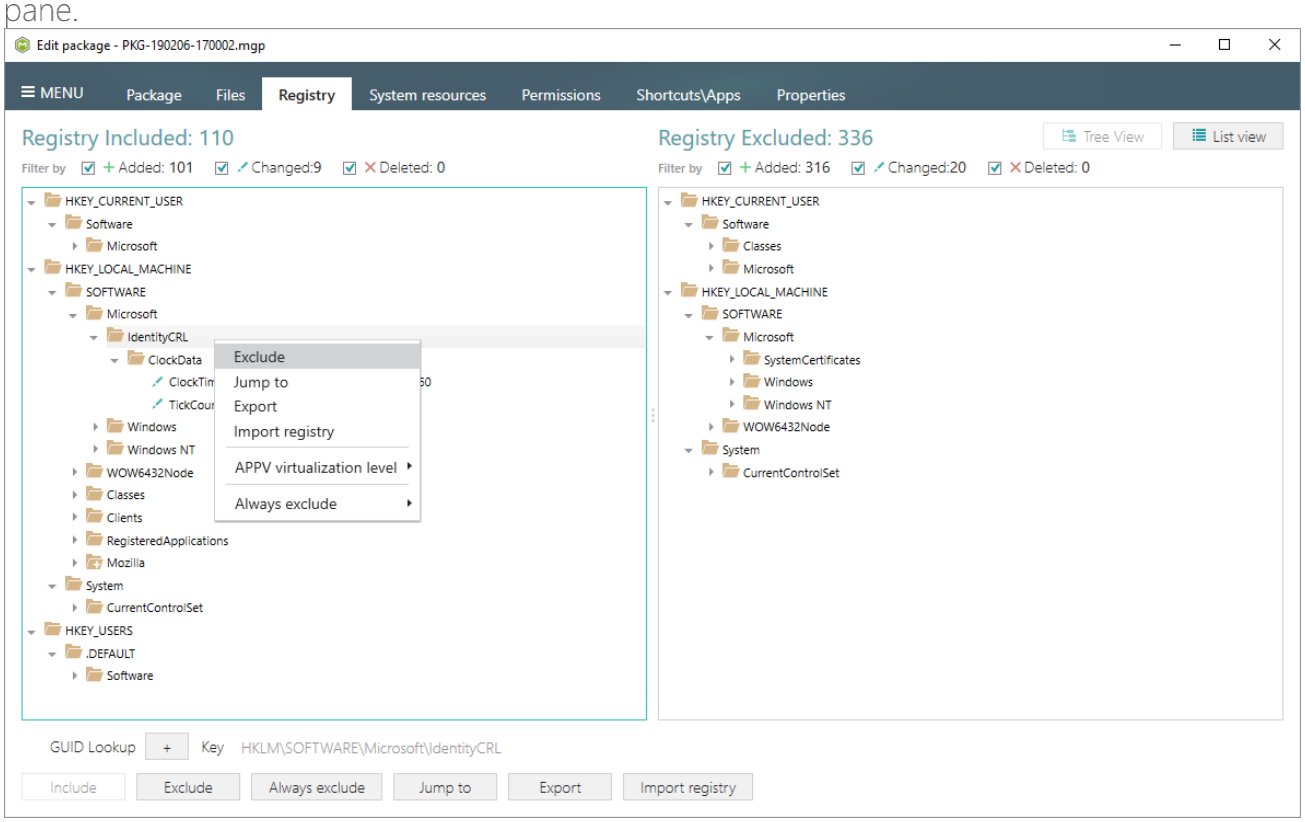


- Once the project is opened in the project editor, it is strongly recommended to review the Files, Registry, System resources, Permissions, and Shortcuts\Apps tabs and exclude the accidentally captured resources, which do not belong to the application. Such accidentally captured resources are also called system noise and it could be Windows Updates, Search indexing, Antivirus software and so on.

- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.

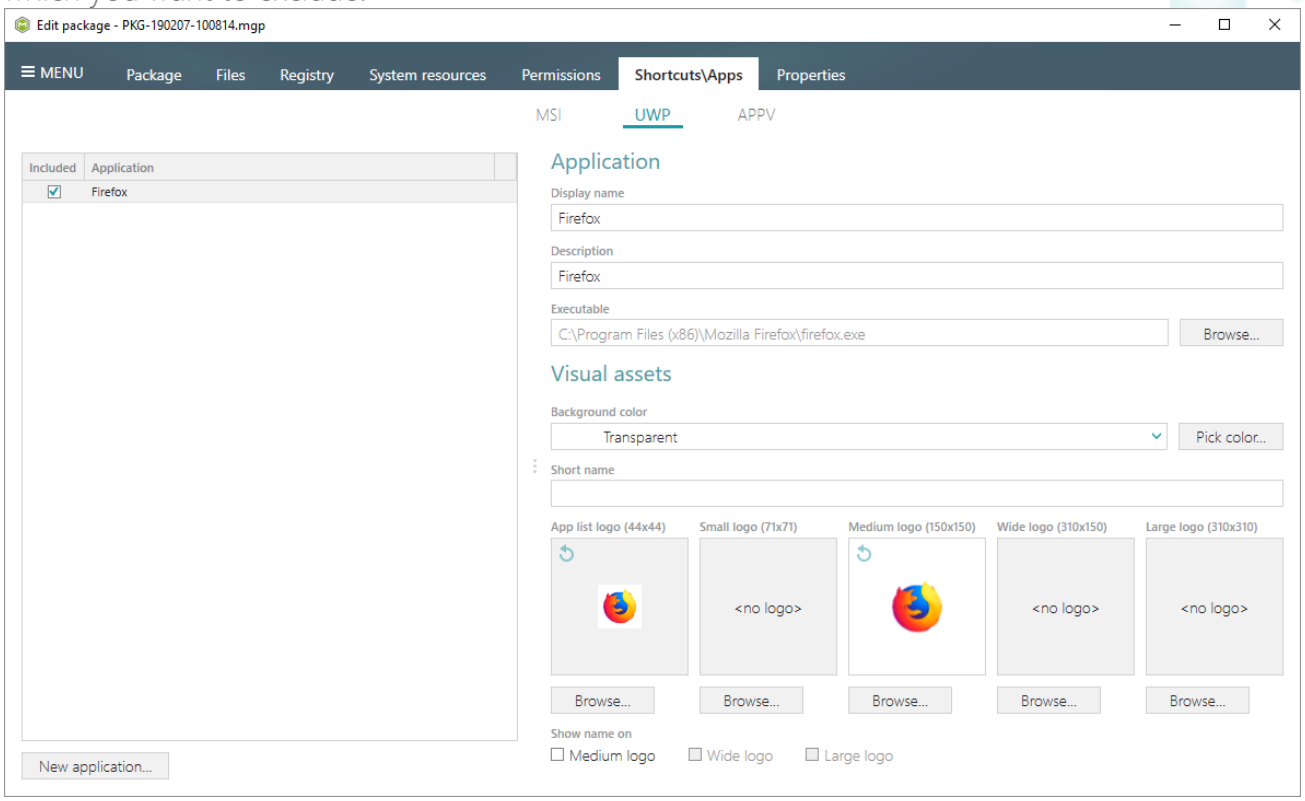


- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.

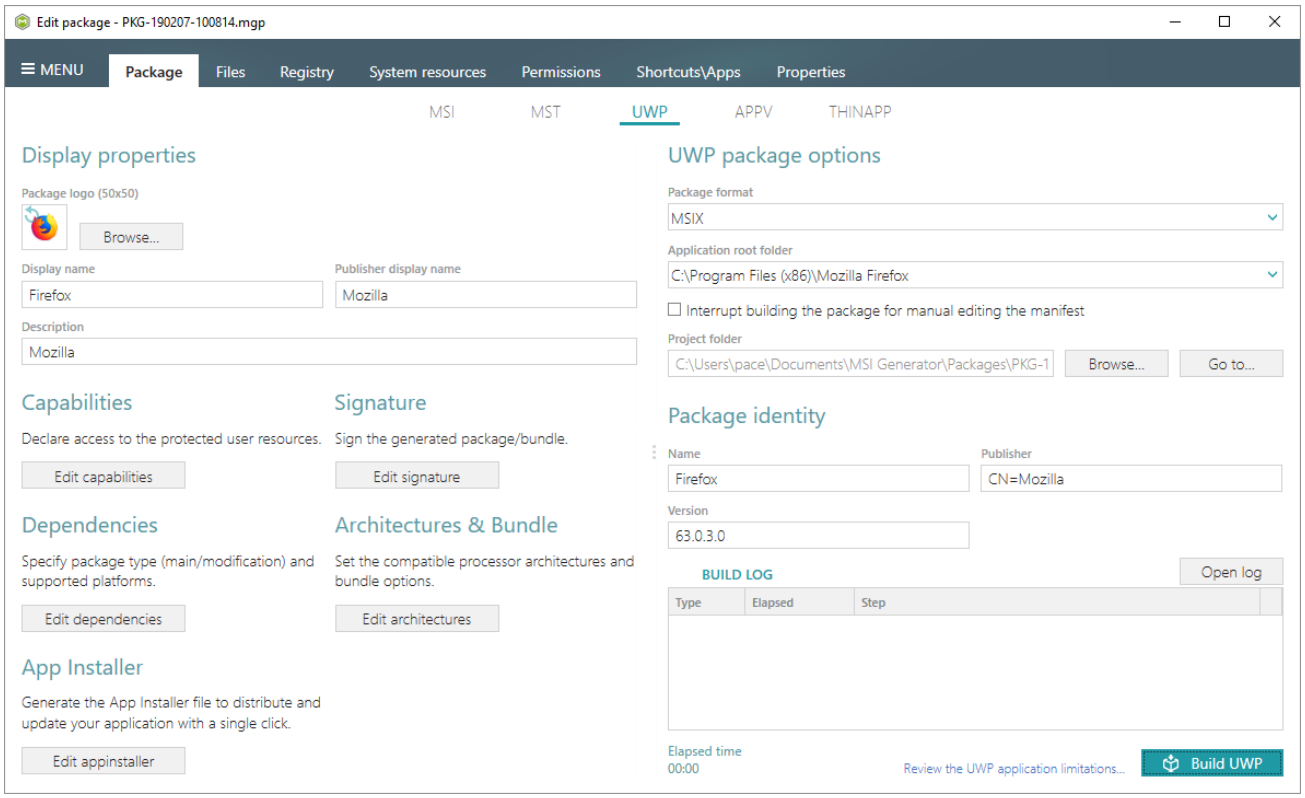


- In order to review and exclude the unnecessary MSIX/APPX apps, go to the **Shortcuts\Apps** -> **UWP** tab, and untick the checkbox, located before the app name in the list, for those apps,

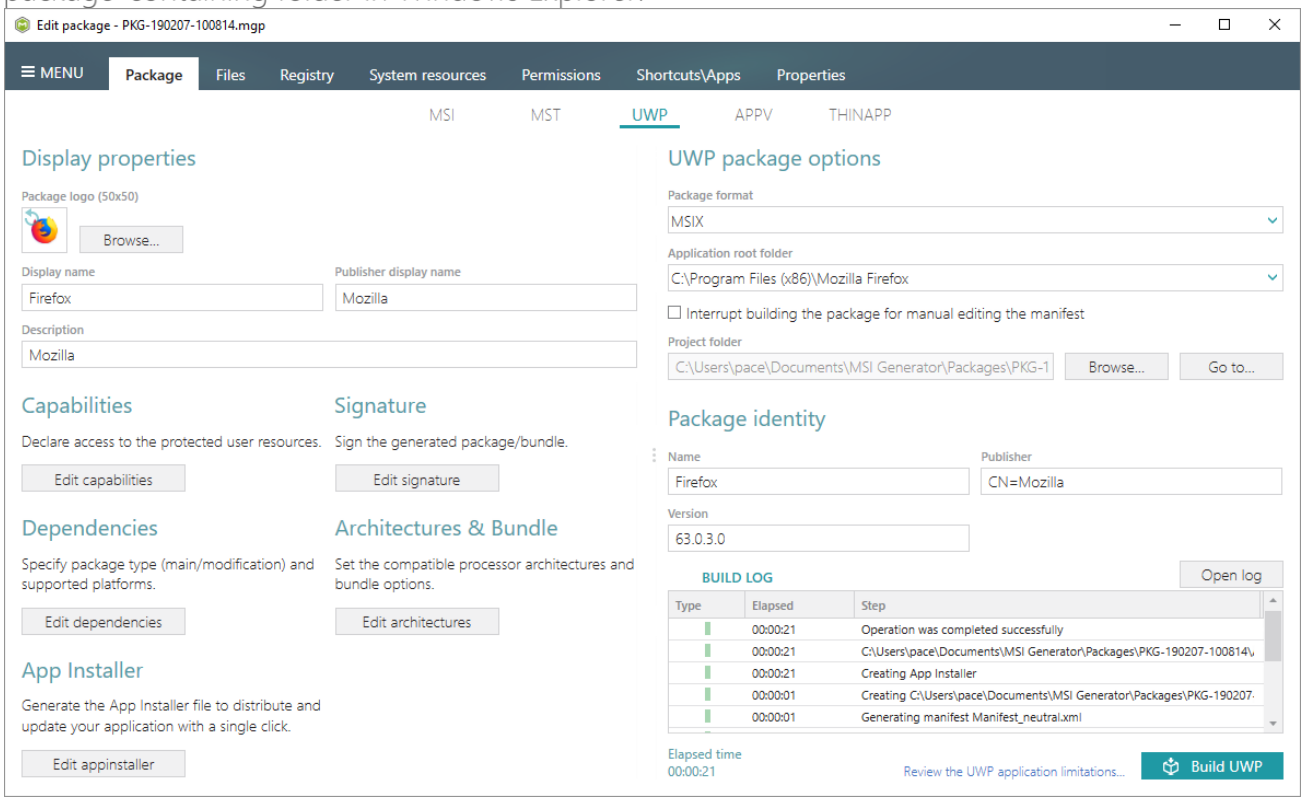
which you want to exclude.



- For managing MSIX/APPX Settings such as display and identity properties, apps and their visual assets, capabilities, signature, dependencies, bundle, AppInstaller, see section 3.8.7
- Finaly, to generate an MSIX/APPX package navigate to the Package -> UWP tab and click Build UWP.



- Once the package is built, click Go to..., located next to the Project folder field, to open the package-containing folder in Windows Explorer.



- To install the generated UWP package, (1) enable the Developer Mode or Sideloading apps on the target system and (2) ensure that the certificate, used for signing, is trusted on that system. If you are using the self signed certificate, install it manually to the Trusted People or Trusted Publishers local machine certificate store. By adding a certificate to local machine certificate stores, you affect the certificate trust of all users on the computer. It is recommended that you remove those certificates when they are no longer necessary to prevent them from being used to compromise system trust.

3.5 Create APPV

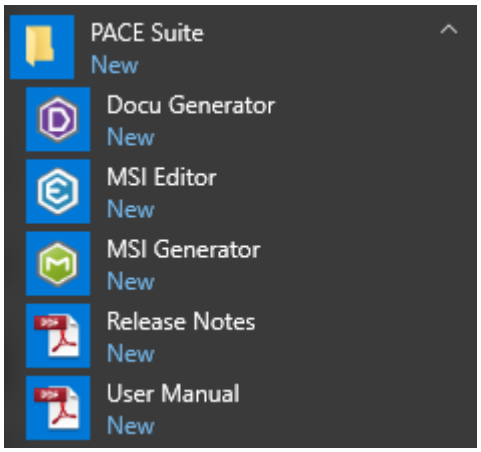
Choose a scenario that better suits your needs:

- Create APPV from Scratch, described in section 3.5.1
  Create a new empty project in MSI Generator, add folders, files, registry entries and APPV apps, shortcuts and FTAs to this project and then generate the APPV 5.0/5.1/1607 (5.2) package from this project.
- Repackage Installation to APPV, described in section 3.5.2
  Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into APPV 5.0/5.1/1607 (5.2) package using MSI Generator.

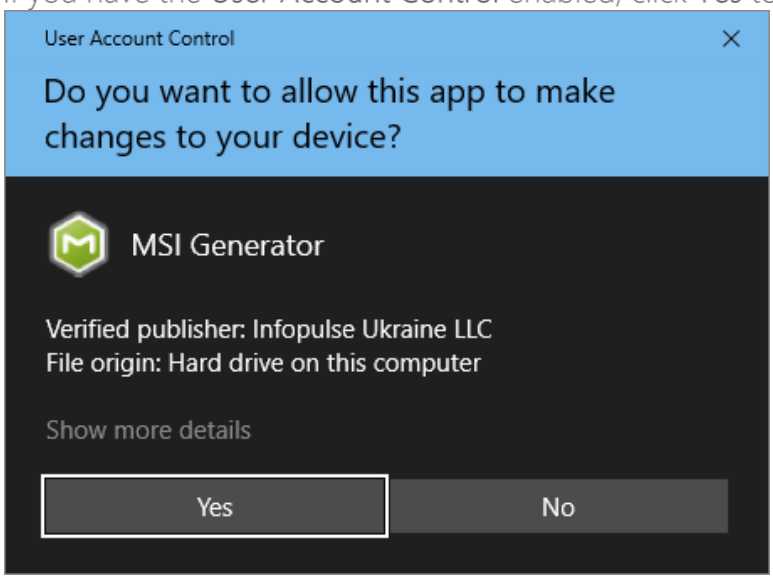
3.5.1 Create APPV from Scratch

Create a new empty project in MSI Generator, add folders, files, registry entries and APPV apps, shortcuts and FTAs to this project and then generate the APPV 5.0/5.1/1607 (5.2) package from this project.

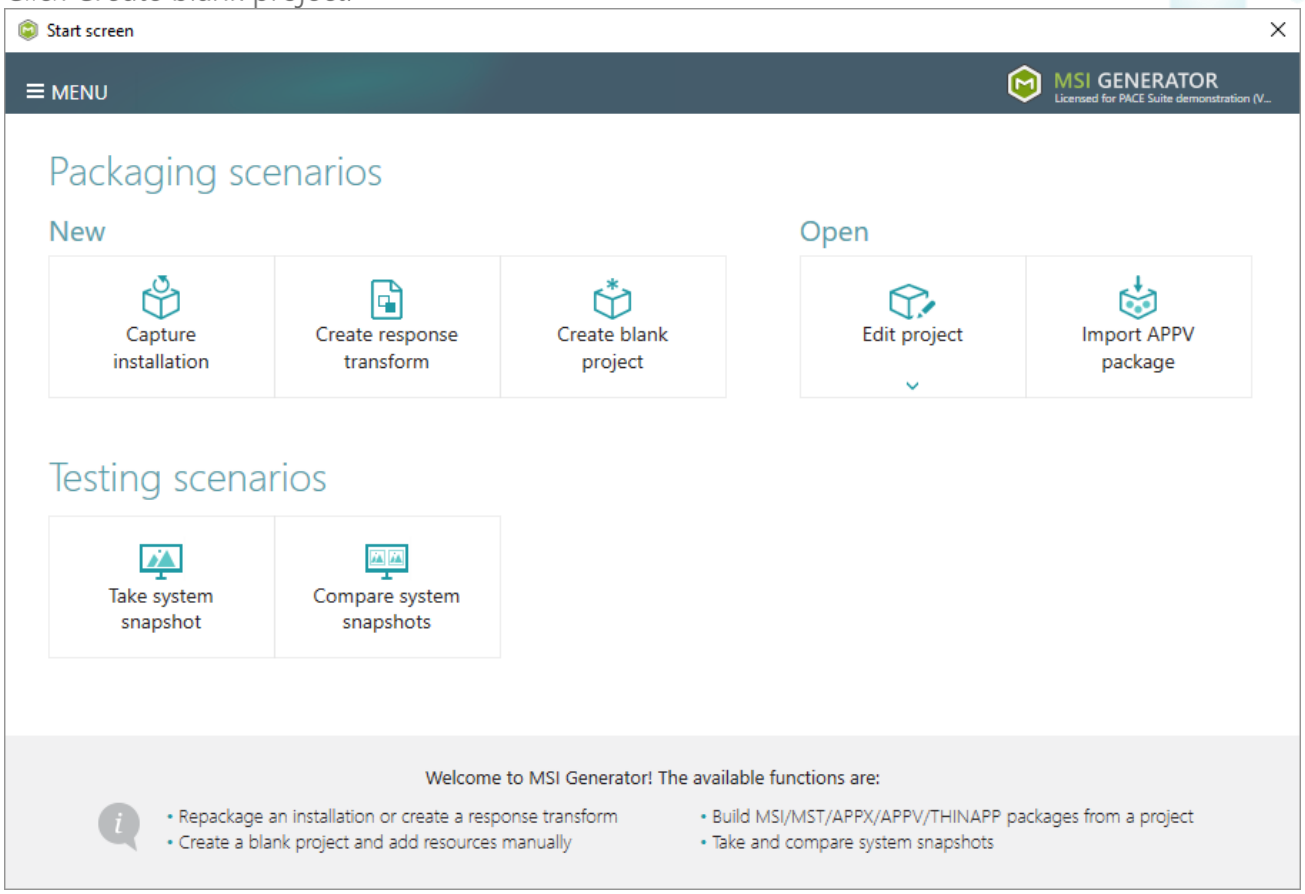
- Launch MSI Generator from the desktop of the start menu shortcut.



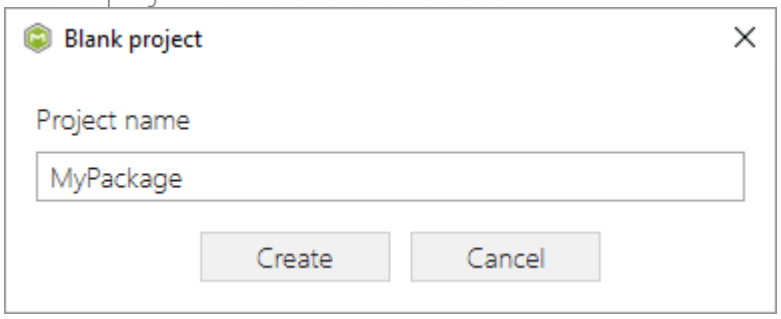
- If you have the User Account Control enabled, click Yes to launch the application.



- Click Create blank project.

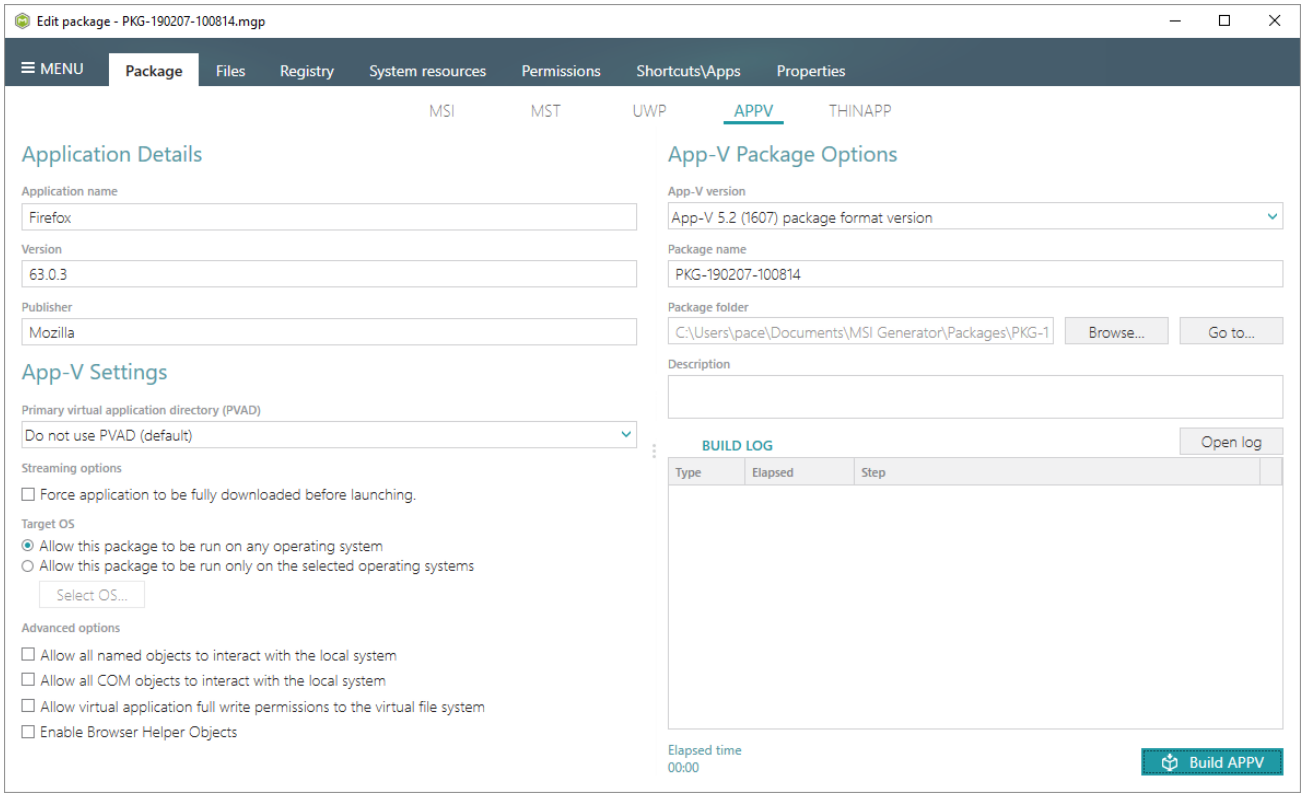


- Enter a project name and click Create.

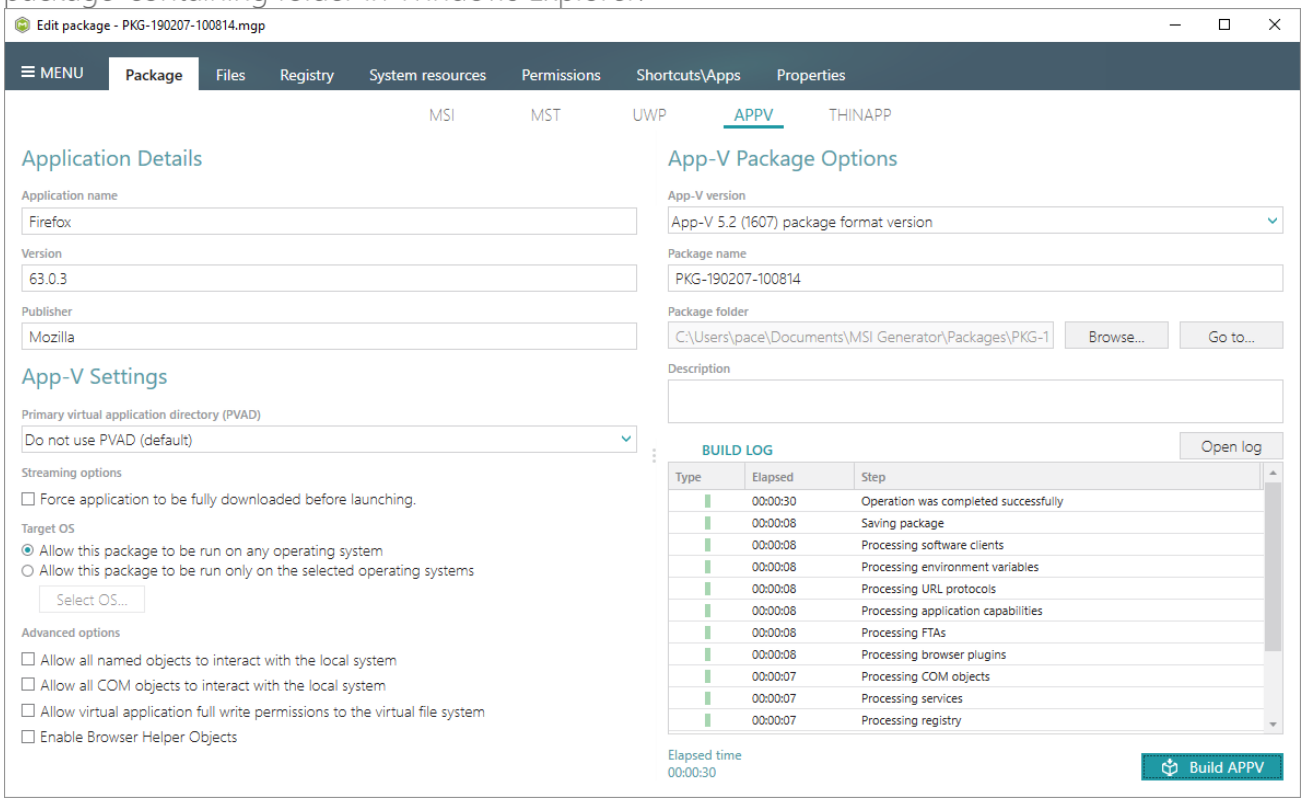


- For managing project resources such as files, folders, and registry refer to the respective section of this document:
    - Regarding File and Folders see section 3.8.2
    - Regarding Registry see section 3.8.3
- For managing APPV Settings such as application details, target OS, PVAD, streamin settings, APPV version, virtualization level for the folders and registry keys, APPV apps, their shortcuts and file type associations (FTA), see section 3.8.8

- Finaly, to generate an APPV package navigate to the **Package** -> **APPV** tab and click **Build APPV**.



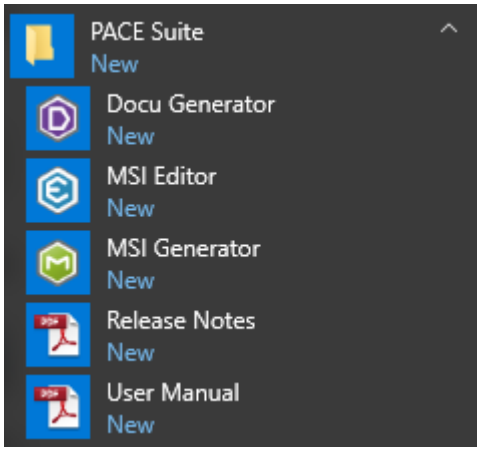
- Once the package is built, click **Go to...**, located next to the **Project folder** field, to open the package-containing folder in Windows Explorer.



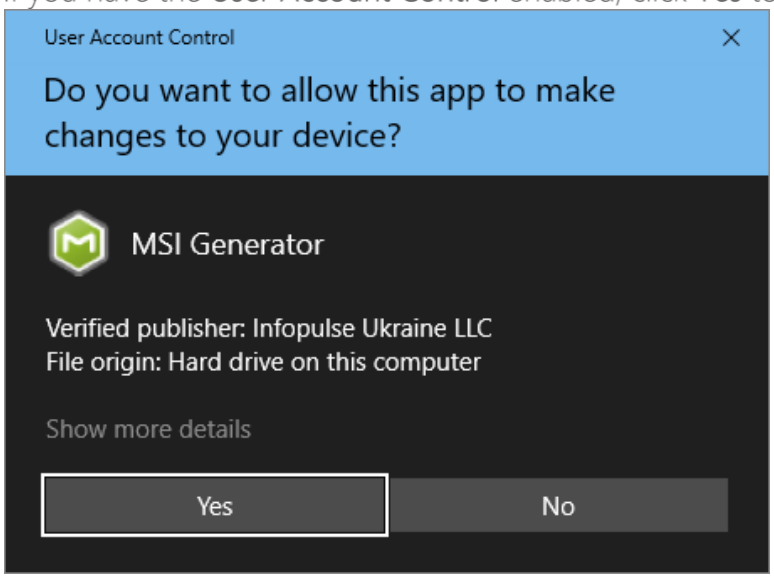
3.5.2 Repackage Installation to APPV

Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into APPV 5.0/5.1/1607 (5.2) package using MSI Generator.

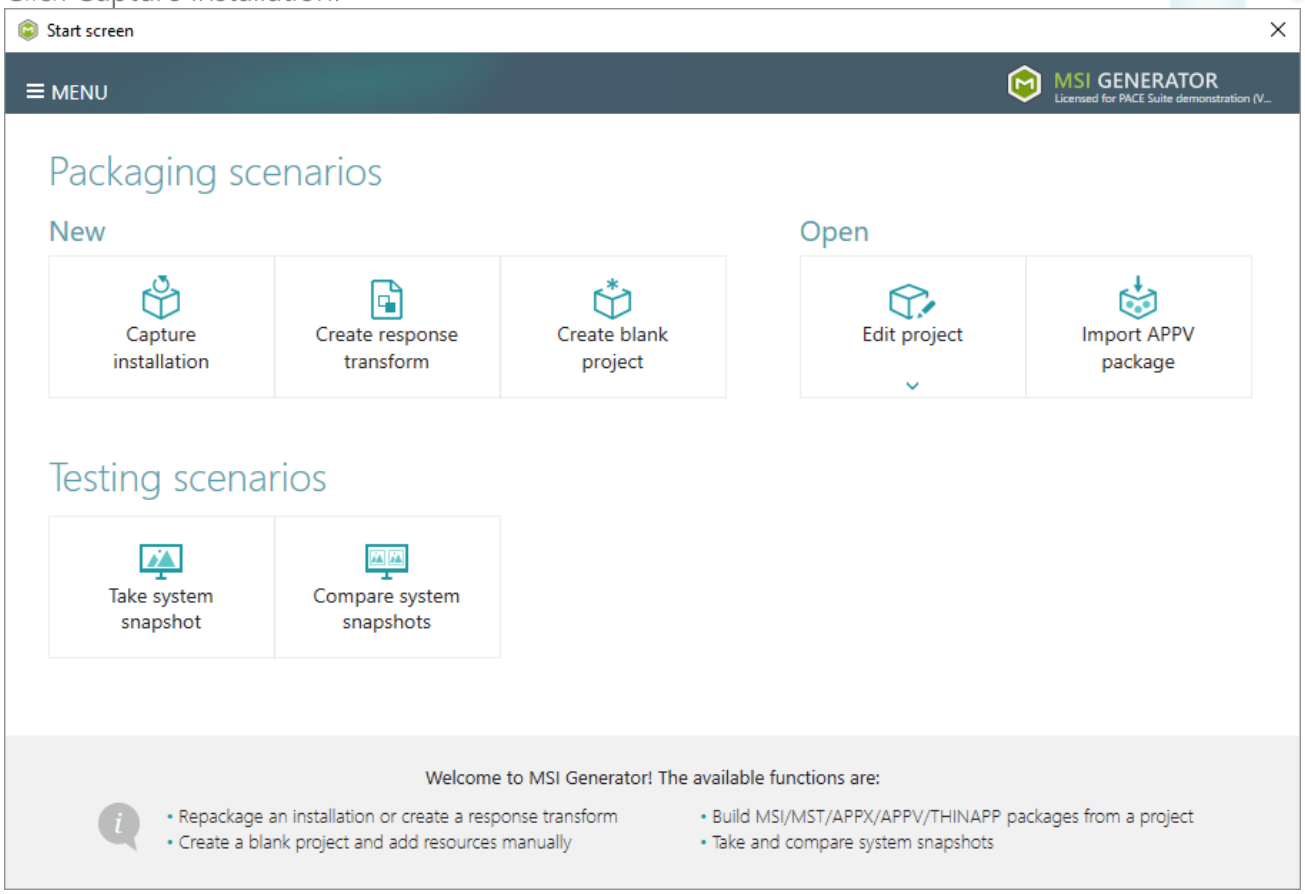
- Launch MSI Generator from the desktop of the start menu shortcut.



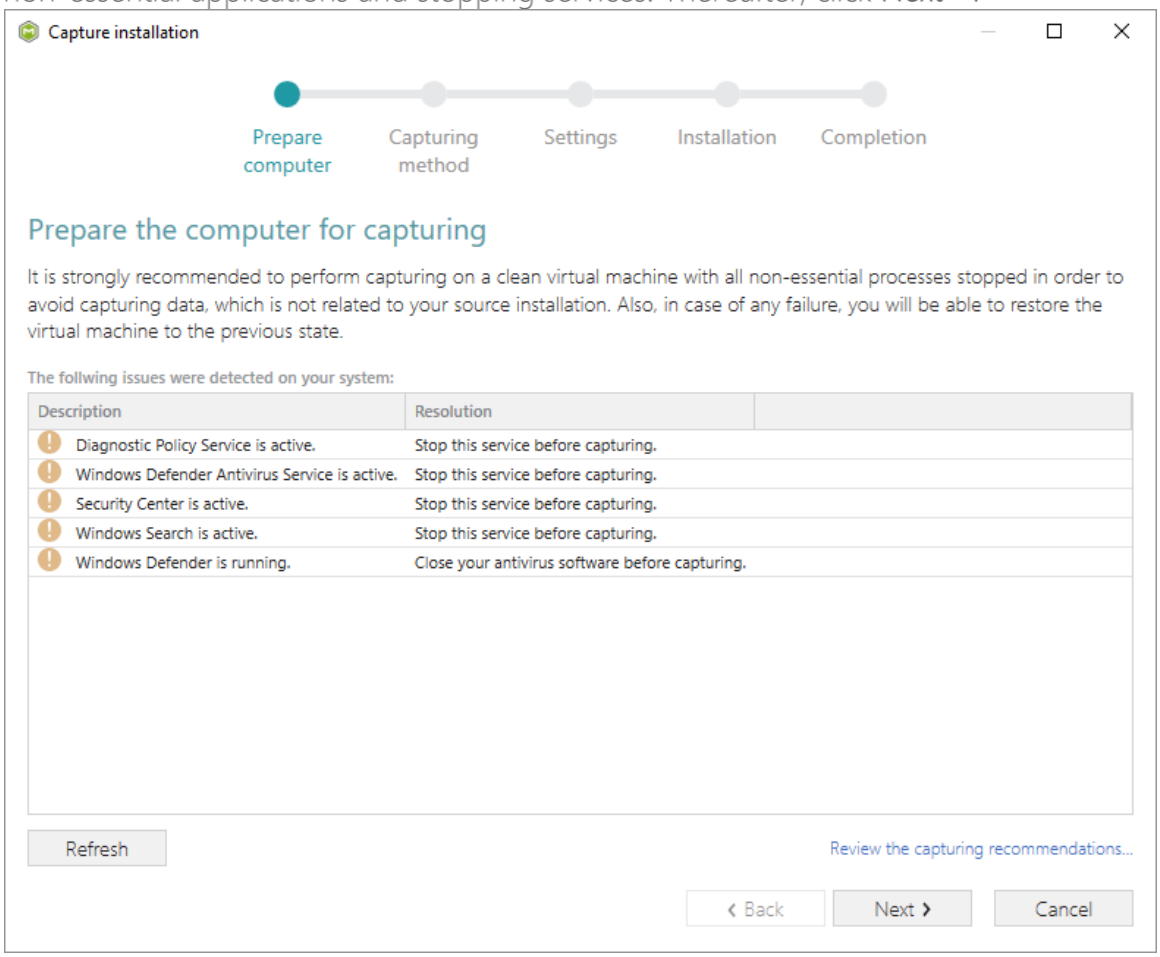
- If you have the User Account Control enabled, click Yes to launch the application.



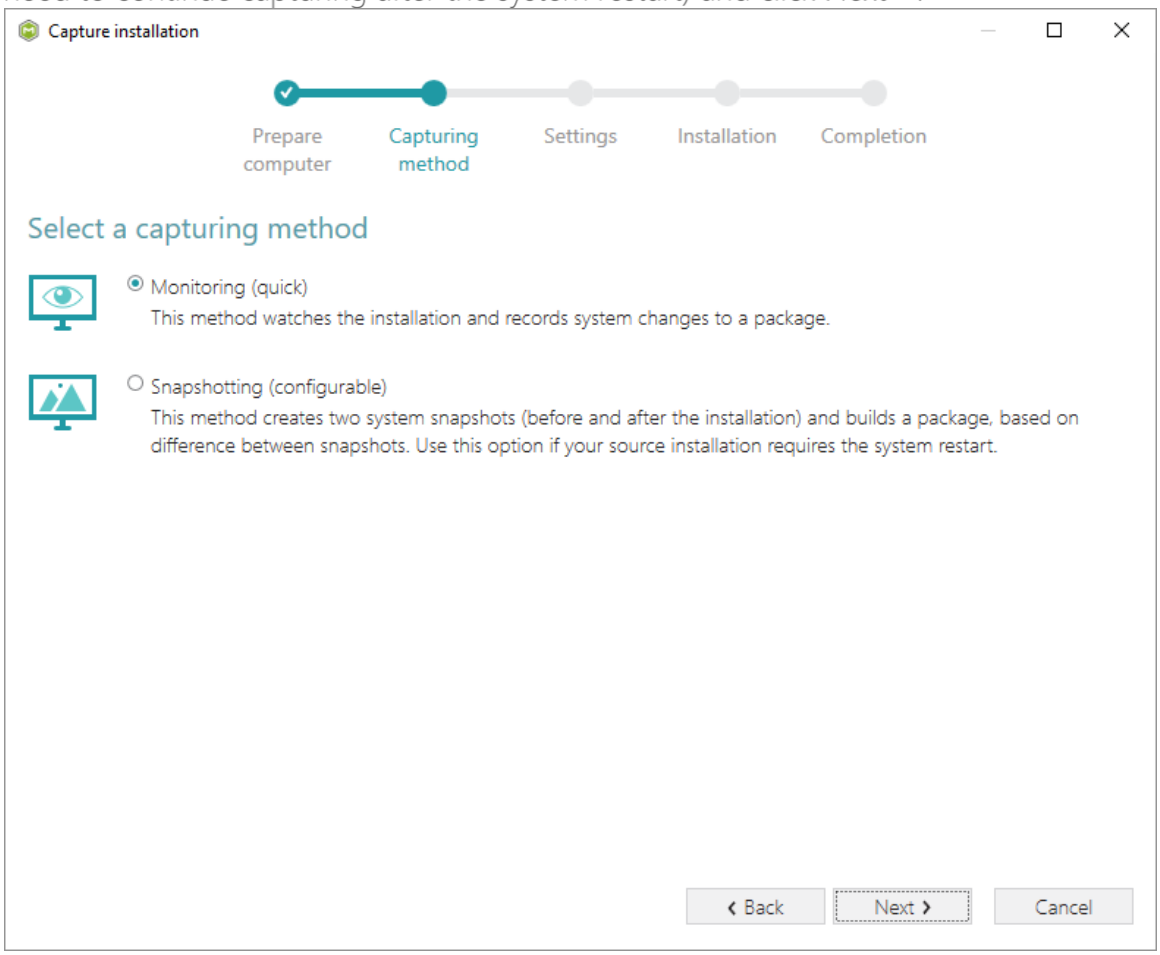
- Click Capture installation.



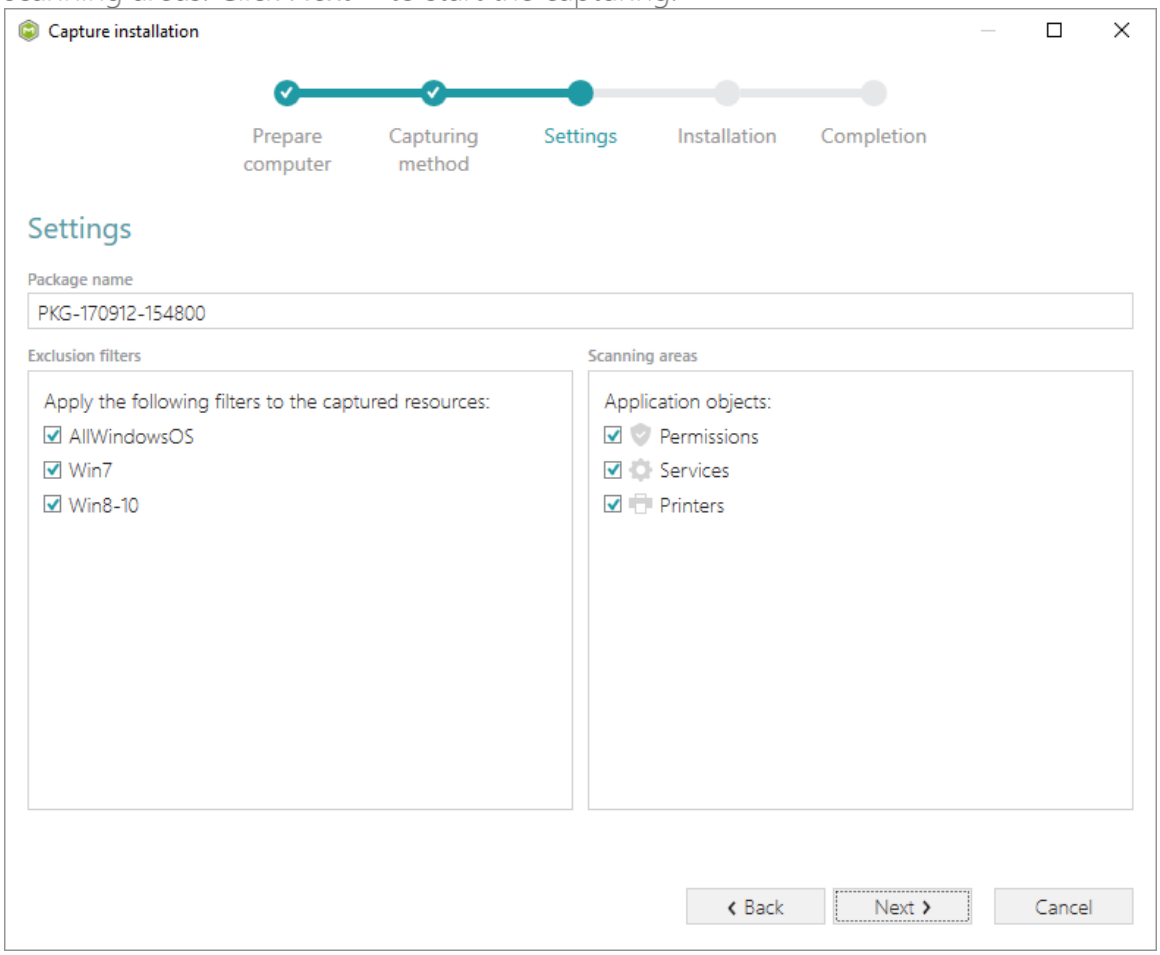
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



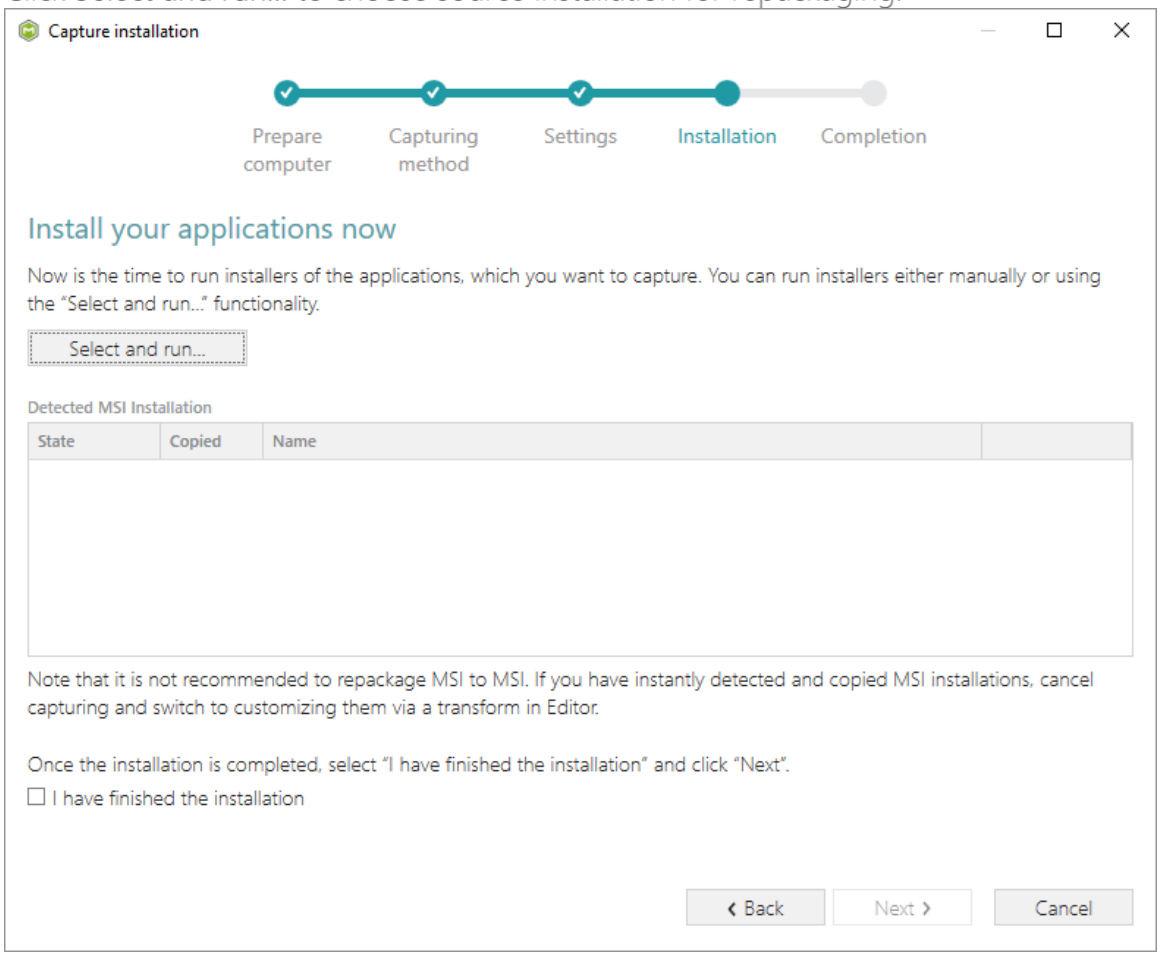
- Select the **Monitoring** method for the quicker capturing (or use the **Snapshotting** one if you need to continue capturing after the system restart) and click **Next >**.



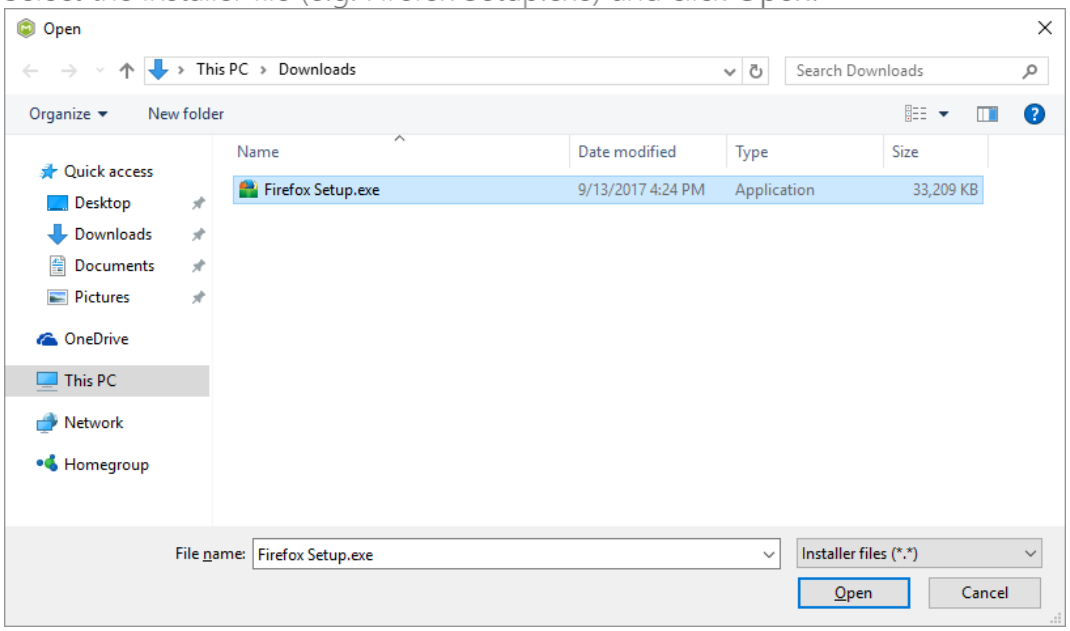
- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.



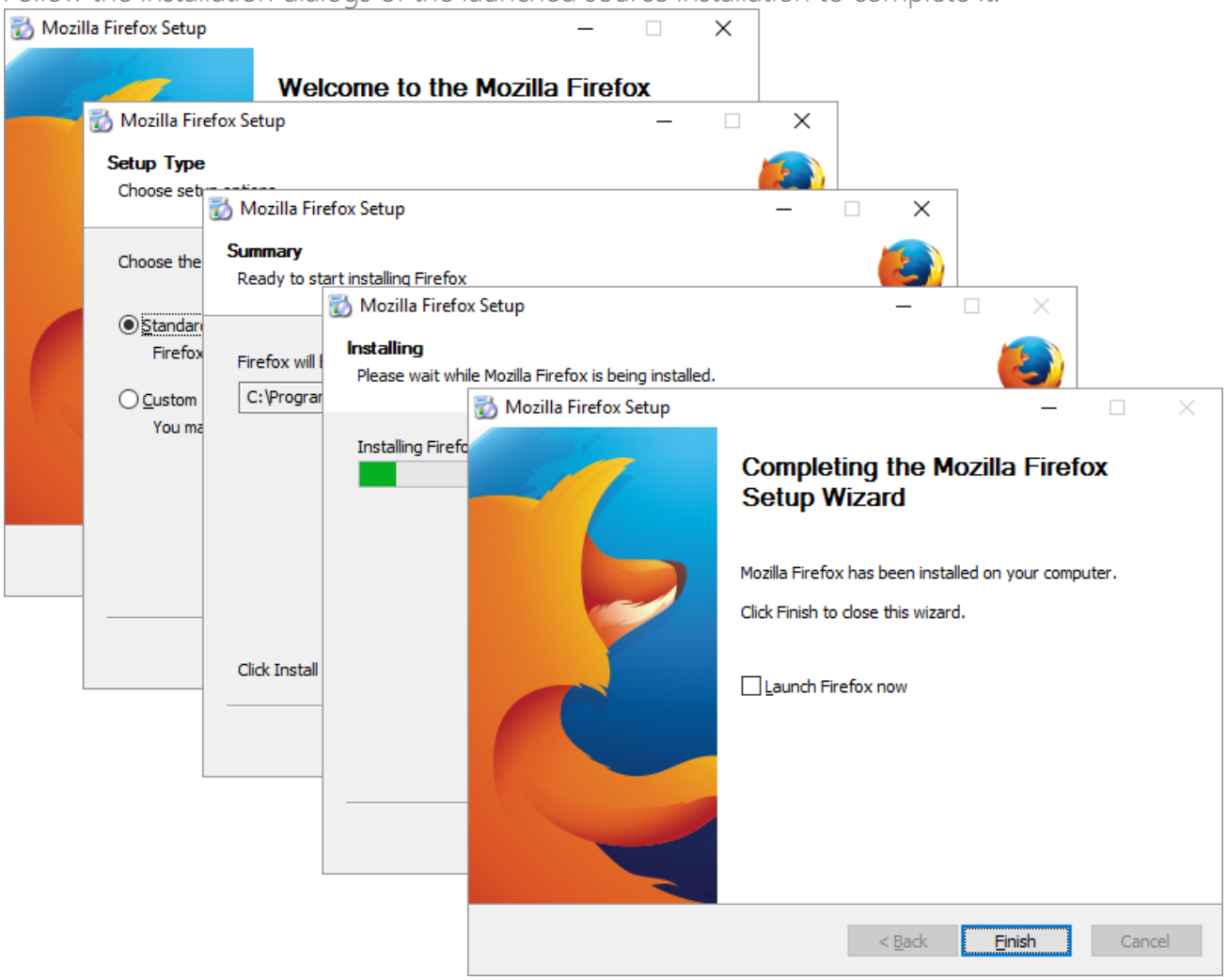
- Click Select and run... to choose source installation for repackaging.



- Select the installer file (e.g. Firefox Setup.exe) and click Open.

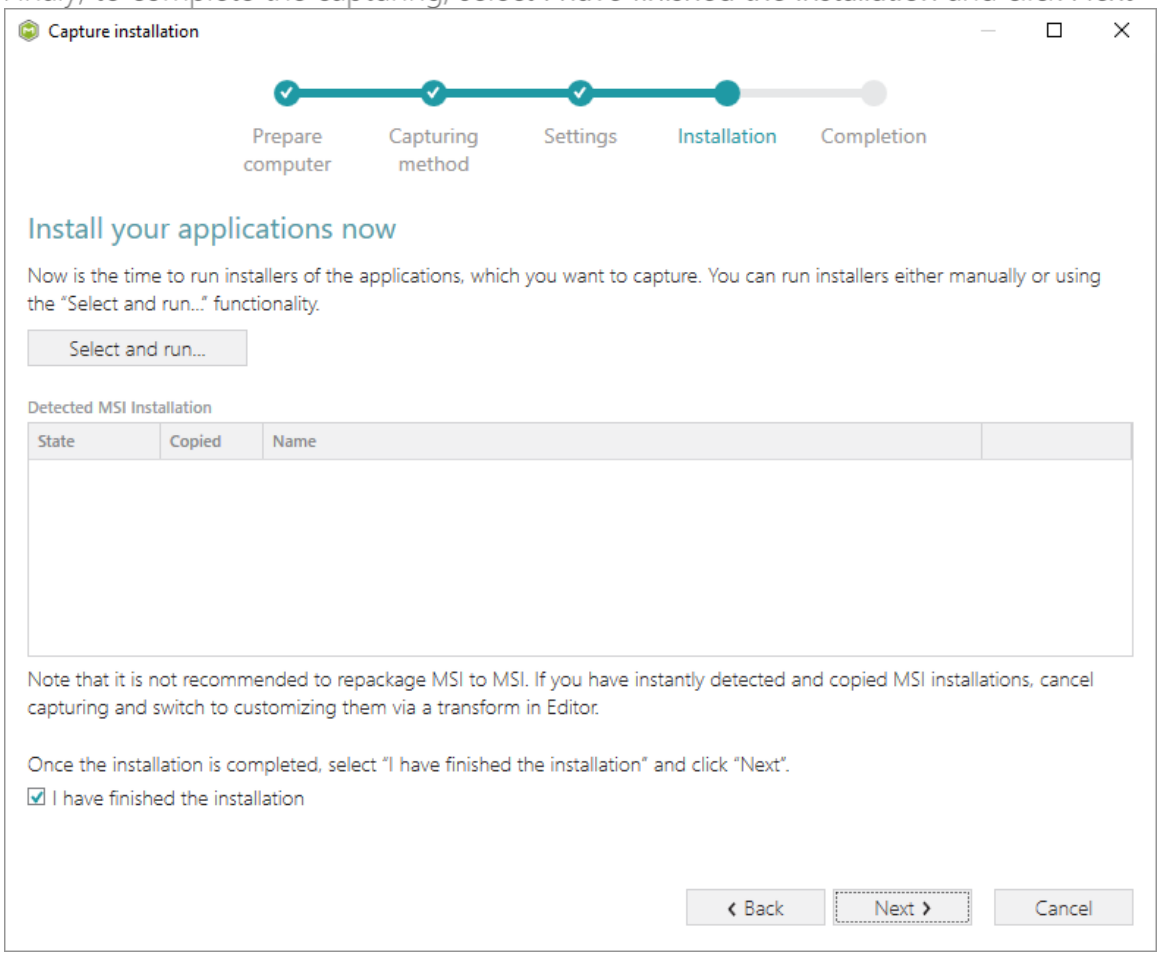


- Follow the installation dialogs of the launched source installation to complete it.

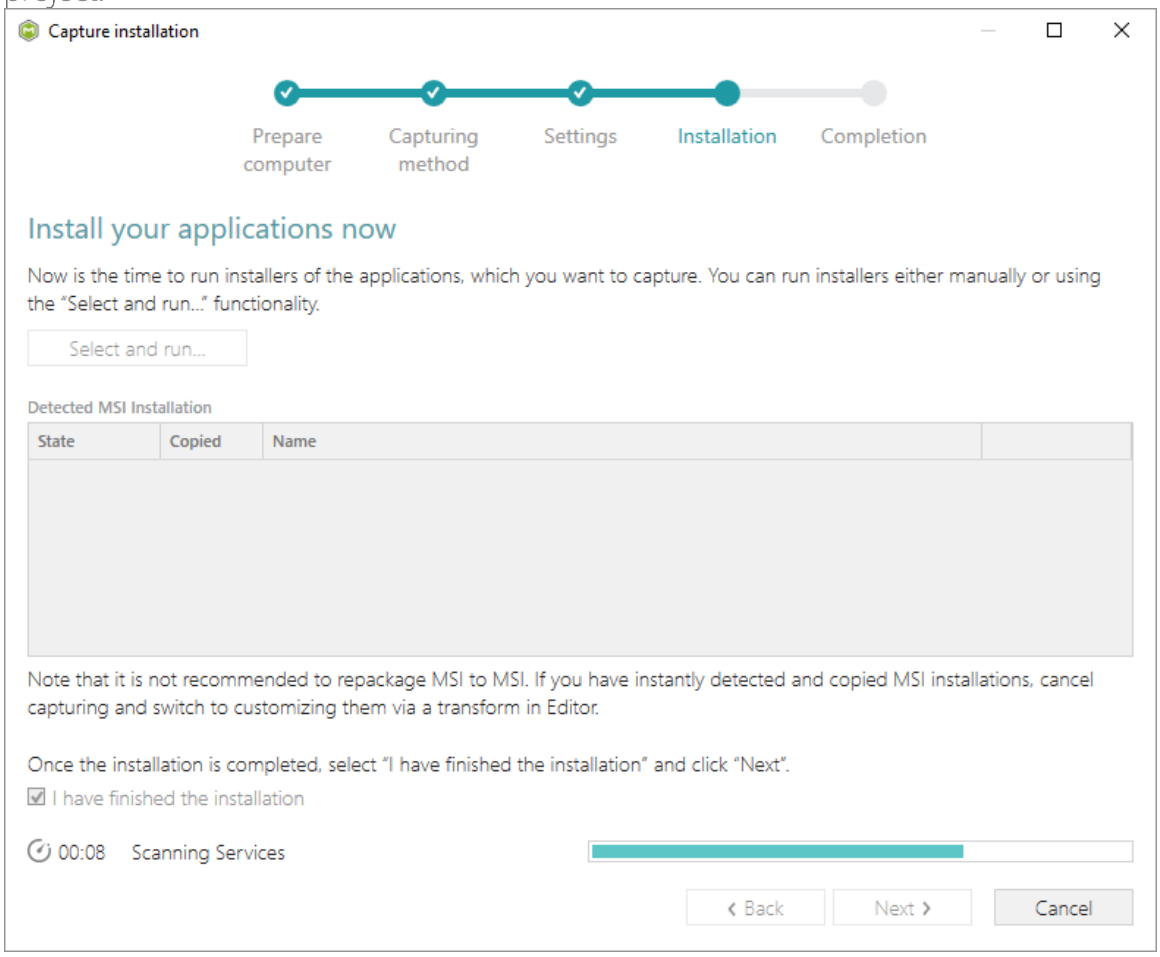


- Now you can make any additional changes to the file system and registry, which you want to capture along with source installation and include to the package. For instance, you can copy or edit configuration files, import REG file to the system registry, change file, folder permissions, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.

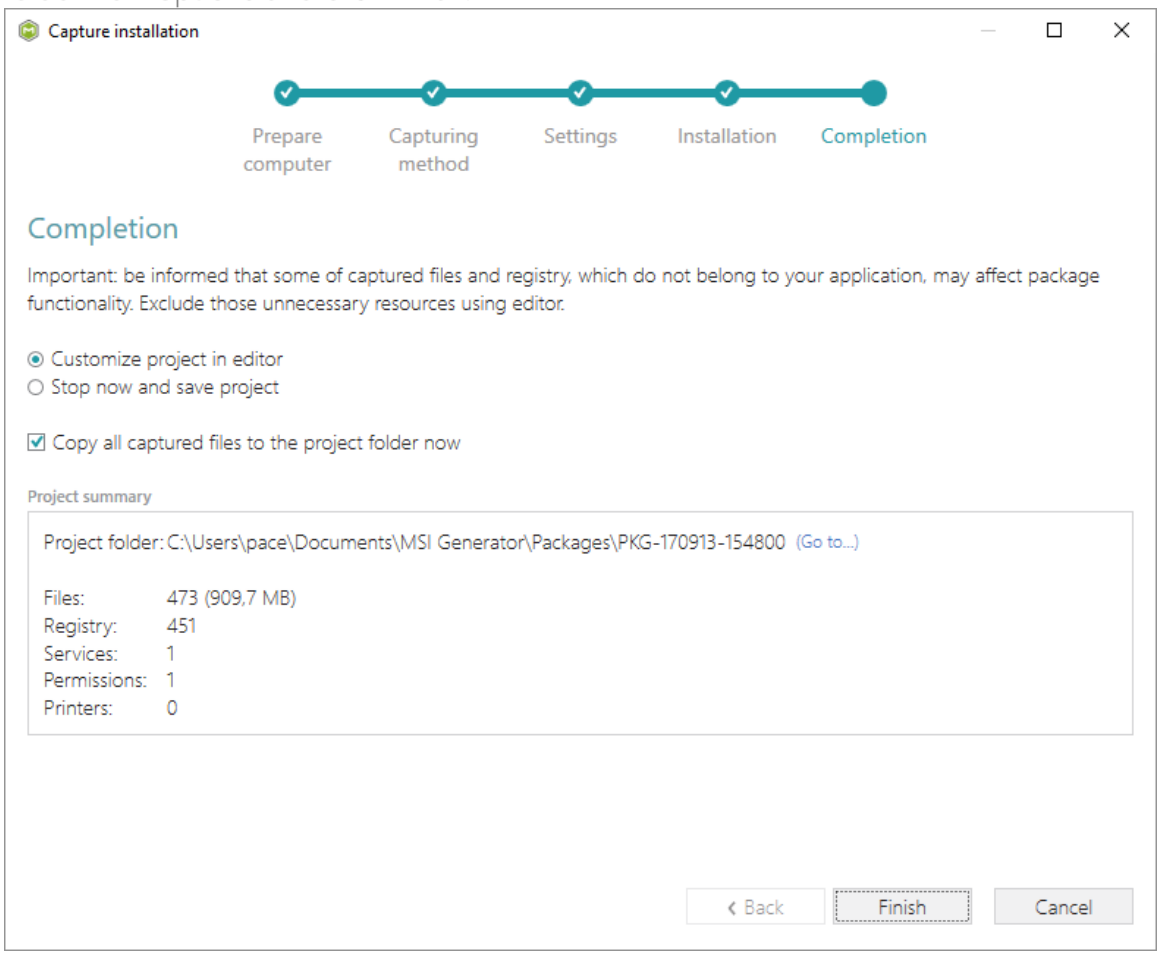
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



- Wait a little, while the capturing process is finishing, filtering captured data and creating the project.

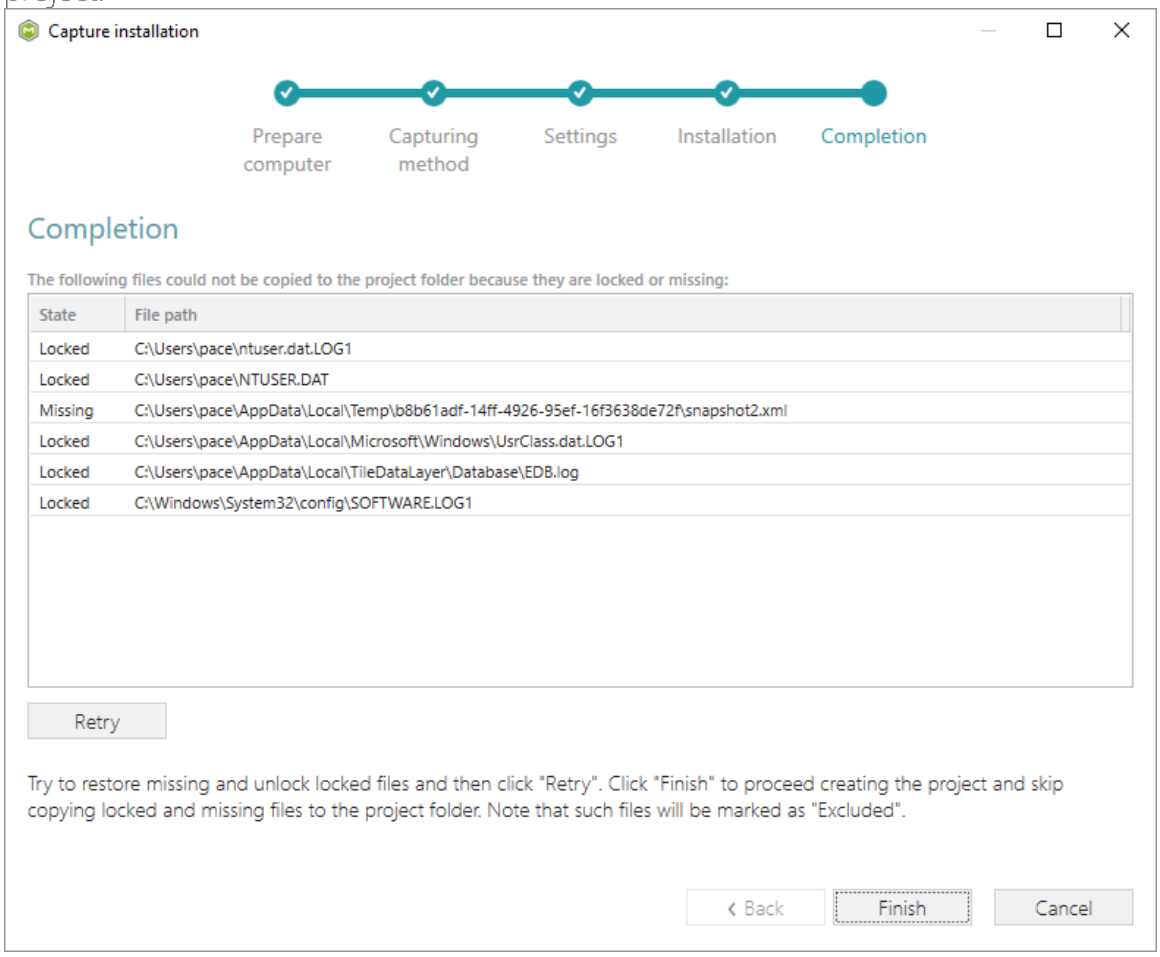


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



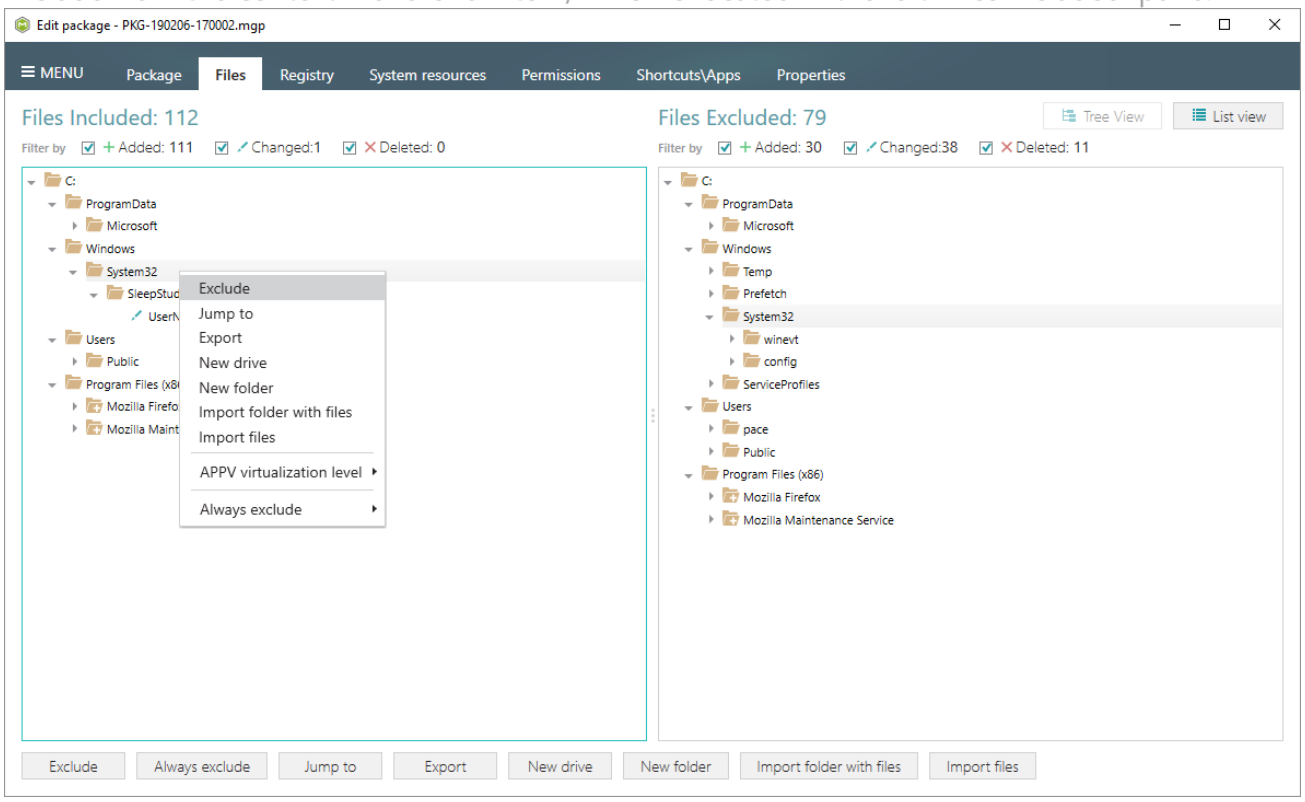
- If some of captured files are locked or missing and as result they could not be copied to the project folder, they will be displayed as illustrated below. Usually, the displayed files are system noise and do not refer to the captured application. Click Finish to continue with skiping locked and missing files. Note that such files will be automatically marked as 'Excluded' in your

project.

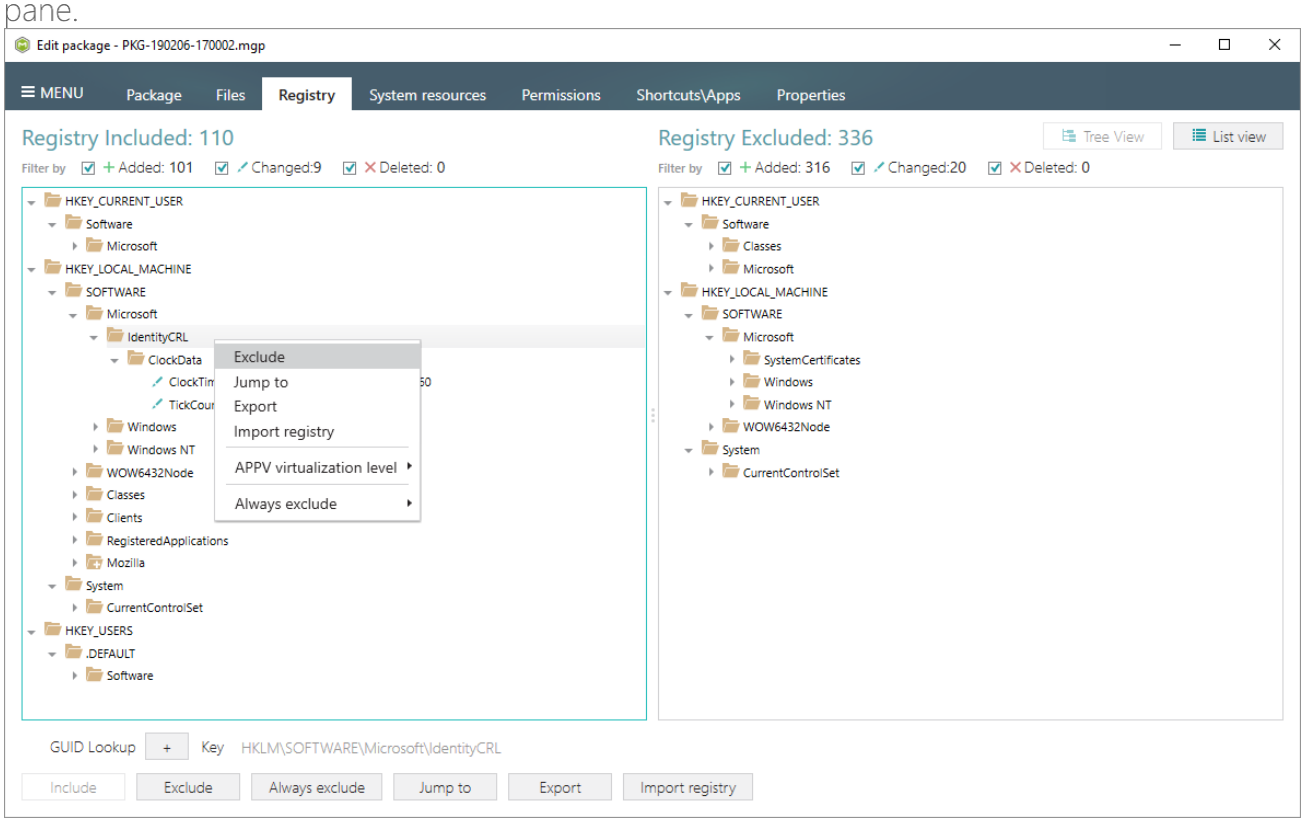


- Once the project is opened in the project editor, it is strongly recommended to review the Files, Registry, System resources, Permissions, and Shortcuts\Apps tabs and exclude the accidentally captured resources, which do not belong to the application. Such accidentally captured resources are also called system noise and it could be Windows Updates, Search indexing, Antivirus software and so on.

- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.

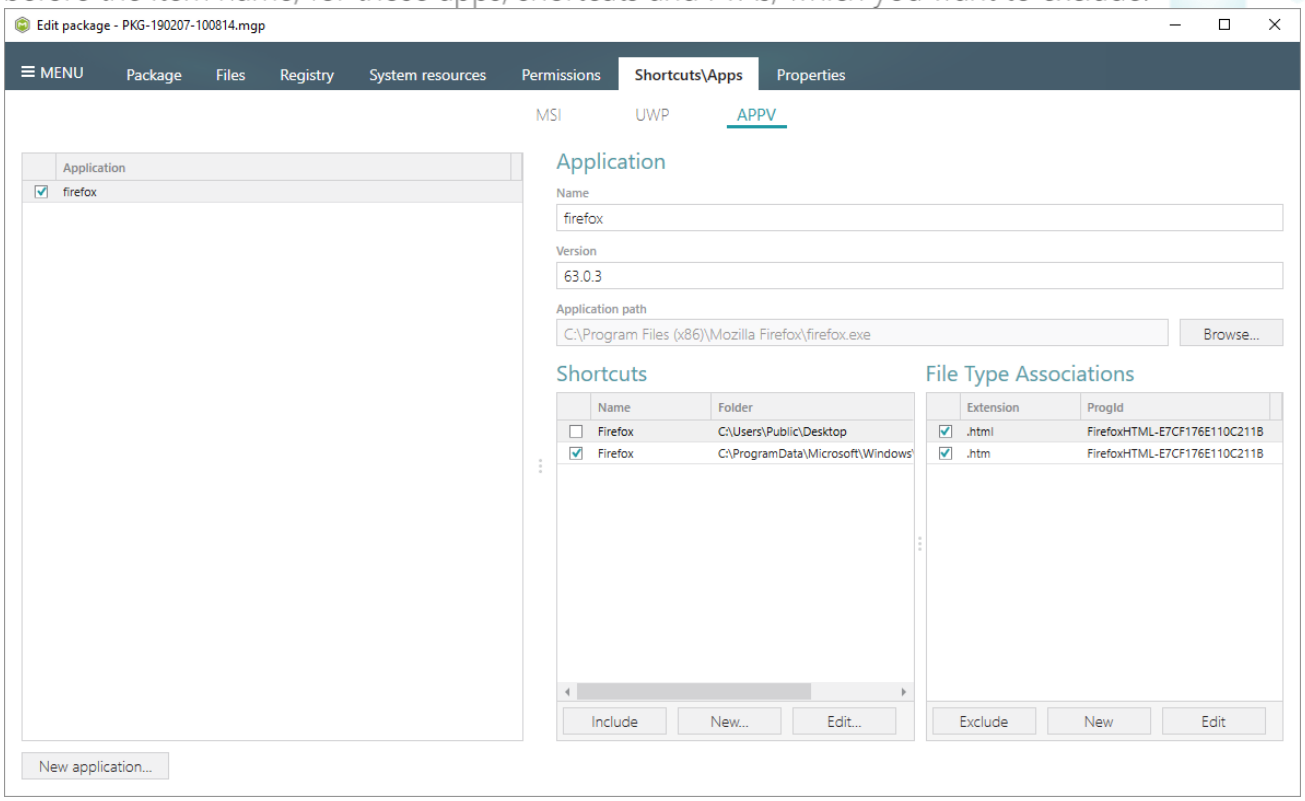


- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.



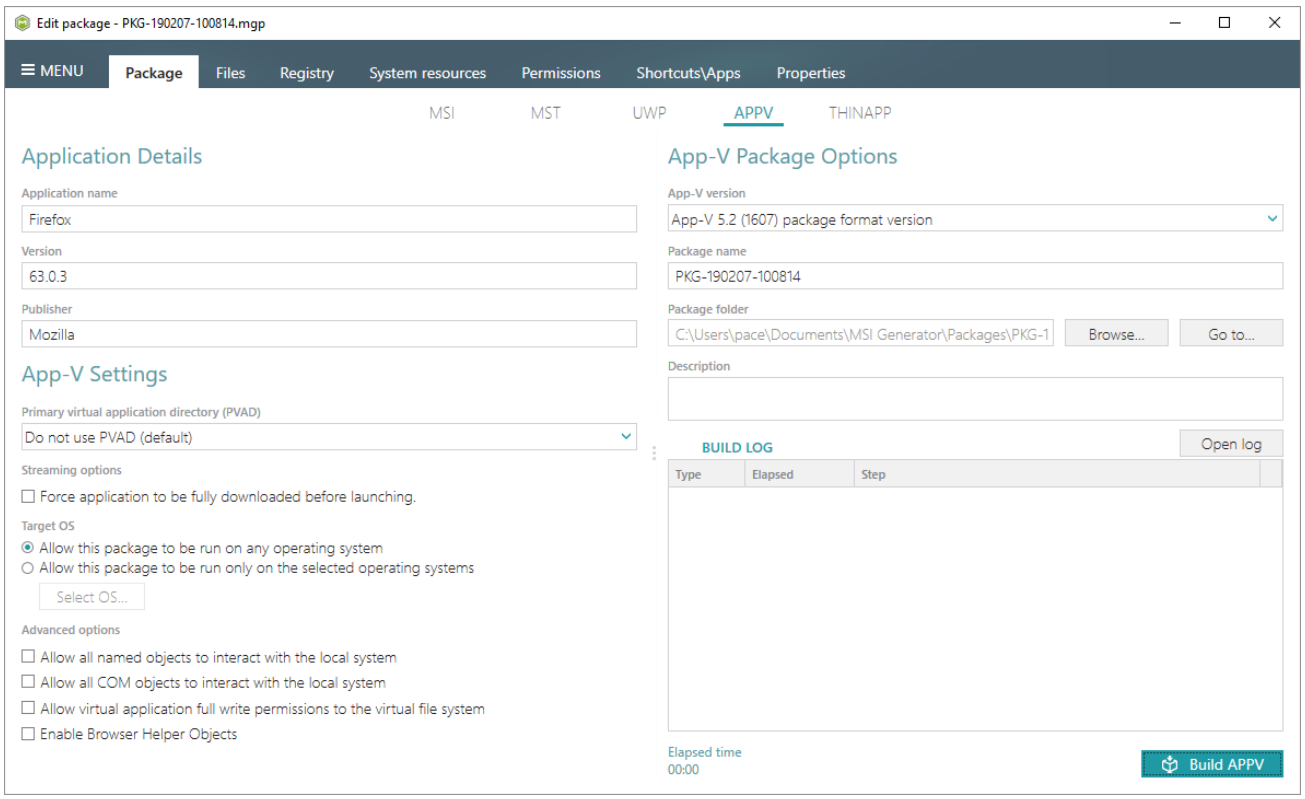
- In order to review and exclude the unnecessary APPV apps, their shortcuts and file type associations (FTA), go to the **Shortcuts\Apps** -> **APPV** tab, and untick the checkboxes, located

before the item name, for those apps, shortcuts and FTAs, which you want to exclude.

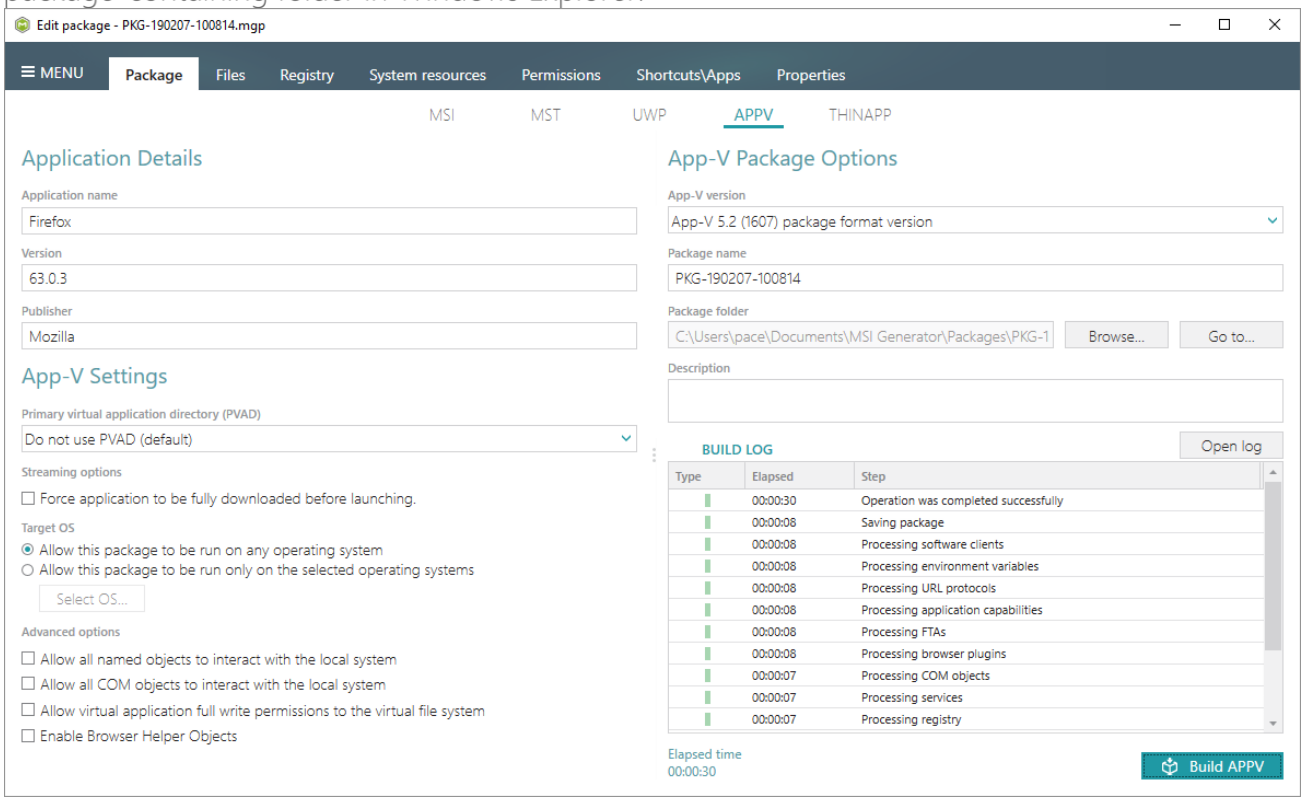


- For managing APPV Settings such as application details, target OS, PVAD, streamin settings, APPV version, virtualization level for the folders and registry keys, APPV apps, their shortcuts and file type associations (FTA), see section 3.8.8

- Finaly, to generate an APPV package navigate to the Package -> APPV tab and click Build APPV.



- Once the package is built, click Go to..., located next to the Project folder field, to open the package-containing folder in Windows Explorer.



3.6 Create THINAPP

NOTE For building THINAPP packages, ensure that your system contains VMware ThinApp (5.0.0/5.0.1/5.1.0/5.1.1/5.2.0) installed.

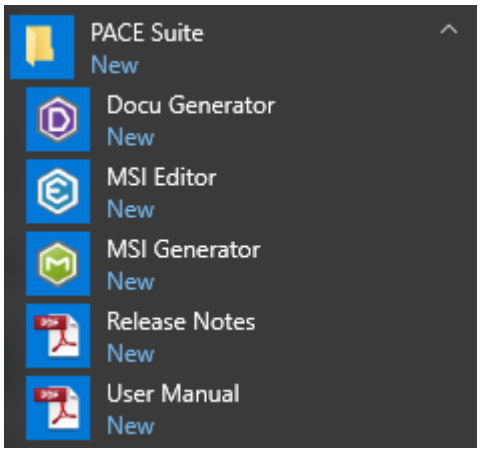
Choose a scenario that better suits your needs:

- Create THINAPP from Scratch, described in section 3.6.1
  Create a new empty project in MSI Generator, add folders, files, registry entries, select entry points and then generate the THINAPP package from this project.
- Repackage Installation to THINAPP, described in section 3.6.2
  Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into THINAPP package using MSI Generator.

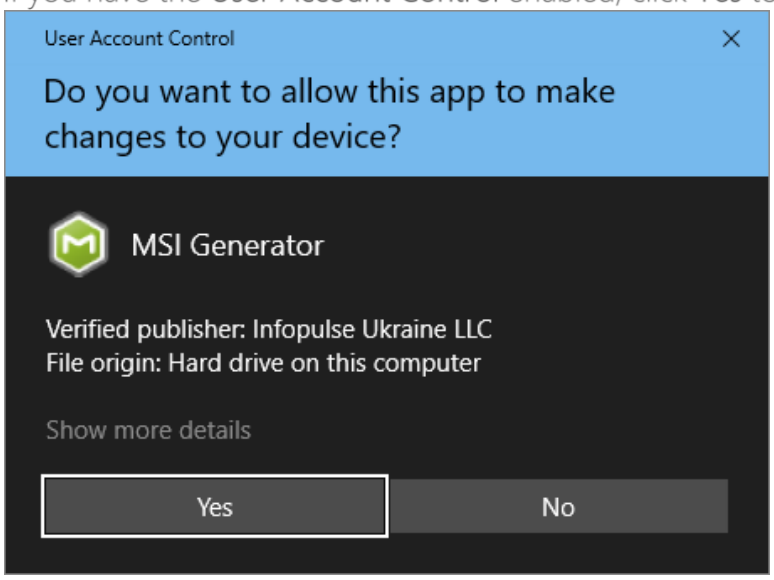
3.6.1 Create THINAPP from Scratch

Create a new empty project in MSI Generator, add folders, files, registry entries, select entry points and then generate the THINAPP package from this project.

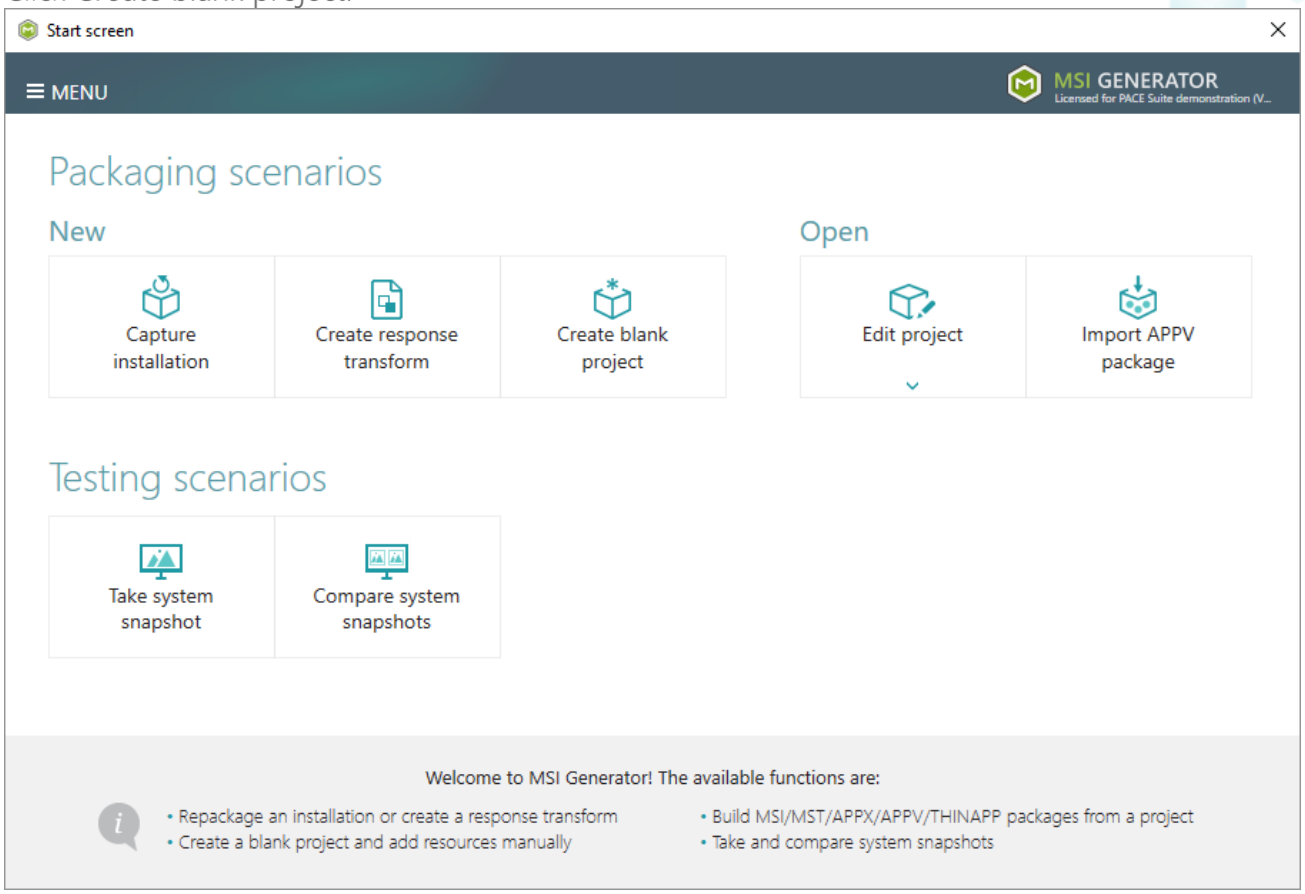
- Launch MSI Generator from the desktop of the start menu shortcut.



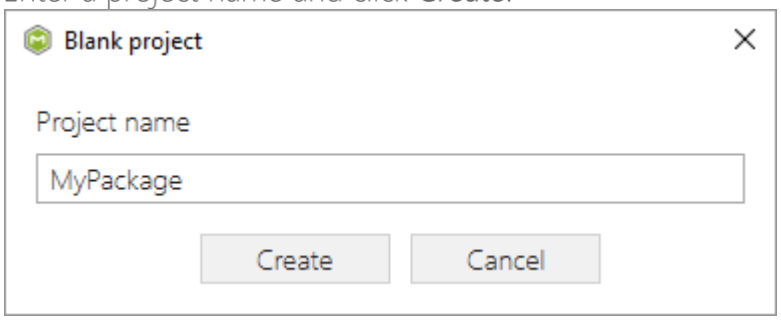
- If you have the User Account Control enabled, click Yes to launch the application.



- Click Create blank project.

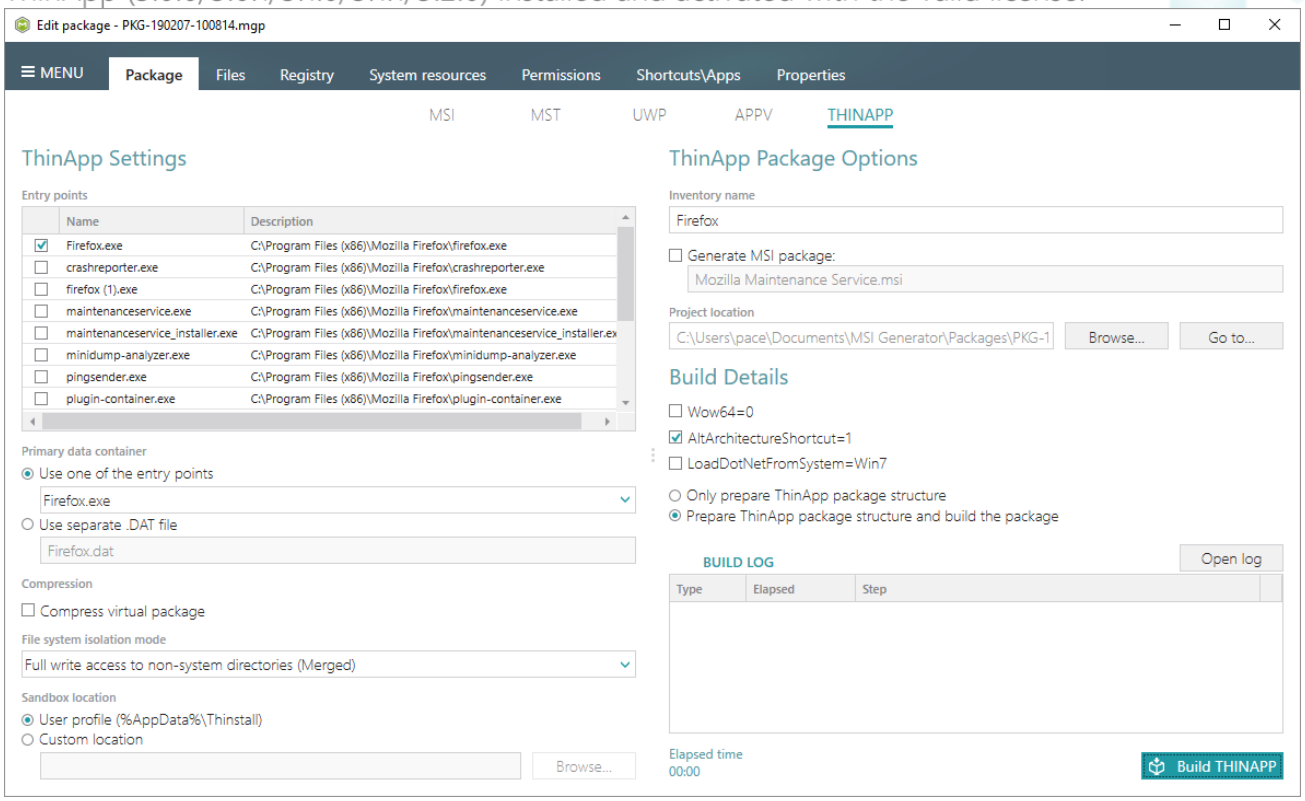


- Enter a project name and click Create.

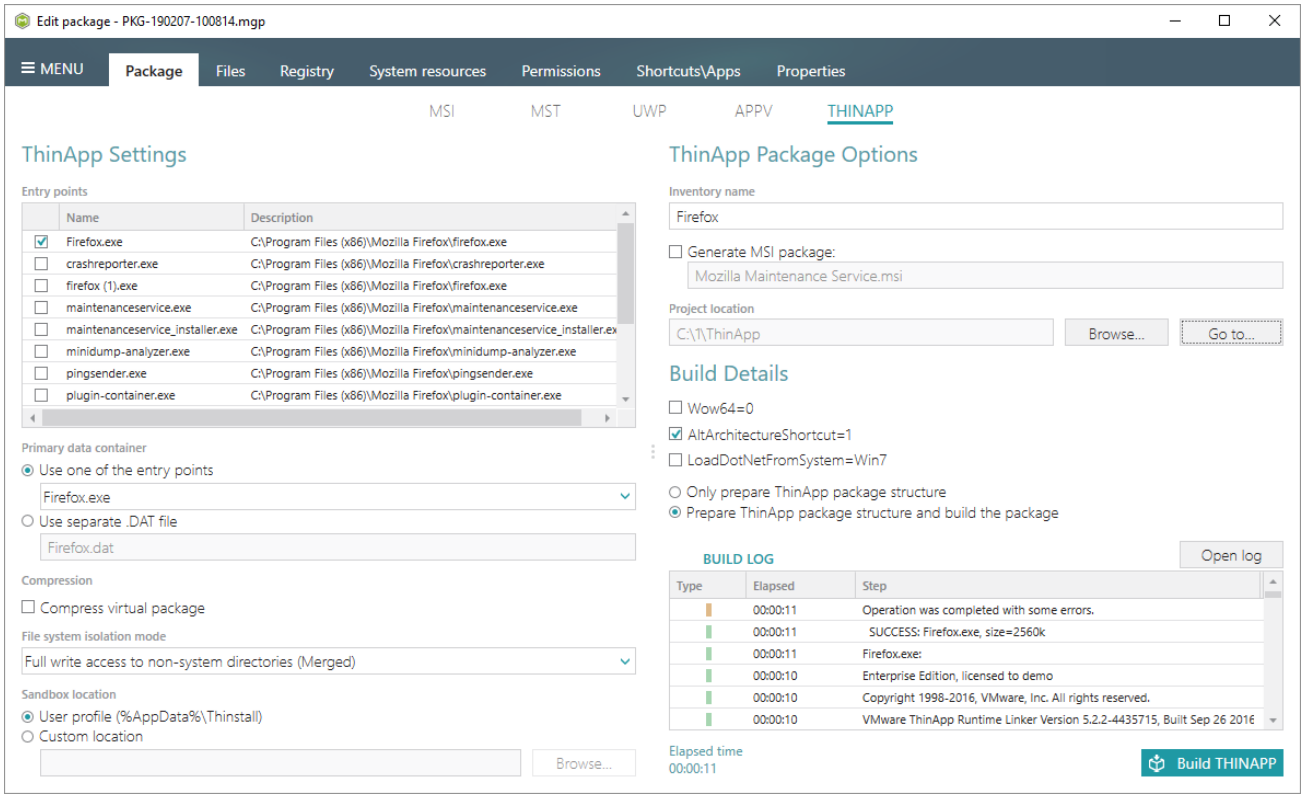


- For managing project resources such as files, folders, and registry refer to the respective section of this document:
    - Regarding File and Folders see section 3.8.2
    - Regarding Registry see section 3.8.3
- For managing THINAPP Settings such as entry points, primary data container, compression, file system isolation mode, sandbox location and the like, see section 3.8.9
- Finaly, to generate a THINAPP package navigate to the Package -> THINAPP tab and click Build THINAPP. NOTE For building THINAPP packages, PACE Suite requires the VMware

ThinApp (5.0.0/5.0.1/5.1.0/5.1.1/5.2.0) installed and activated with the valid license.



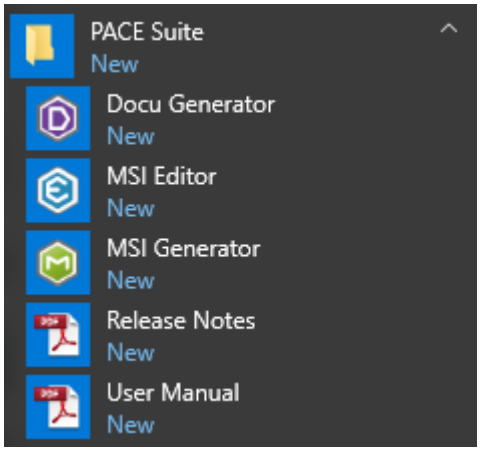
- Once the package is built, click Go to..., located next to the Project location field, to open the package-containing folder in Windows Explorer. Find the generated package in the bin subfolder.



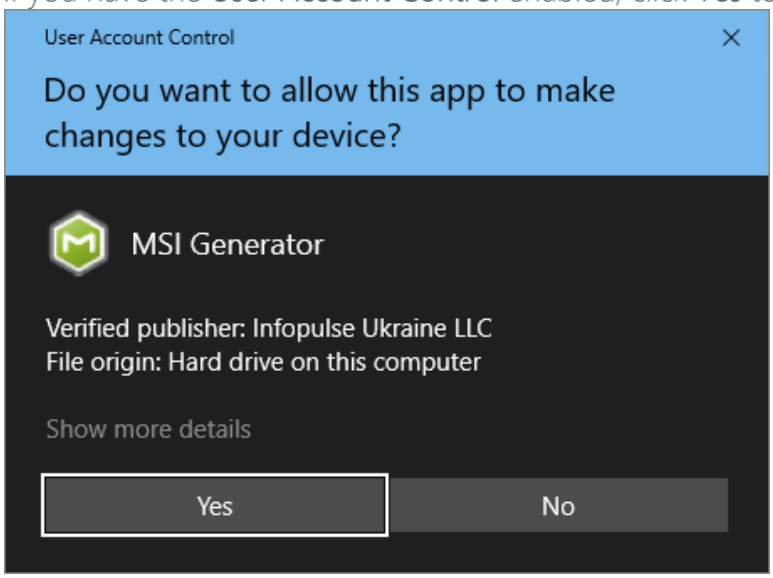
3.6.2 Repackage Installation to THINAPP

Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into THINAPP package using MSI Generator.

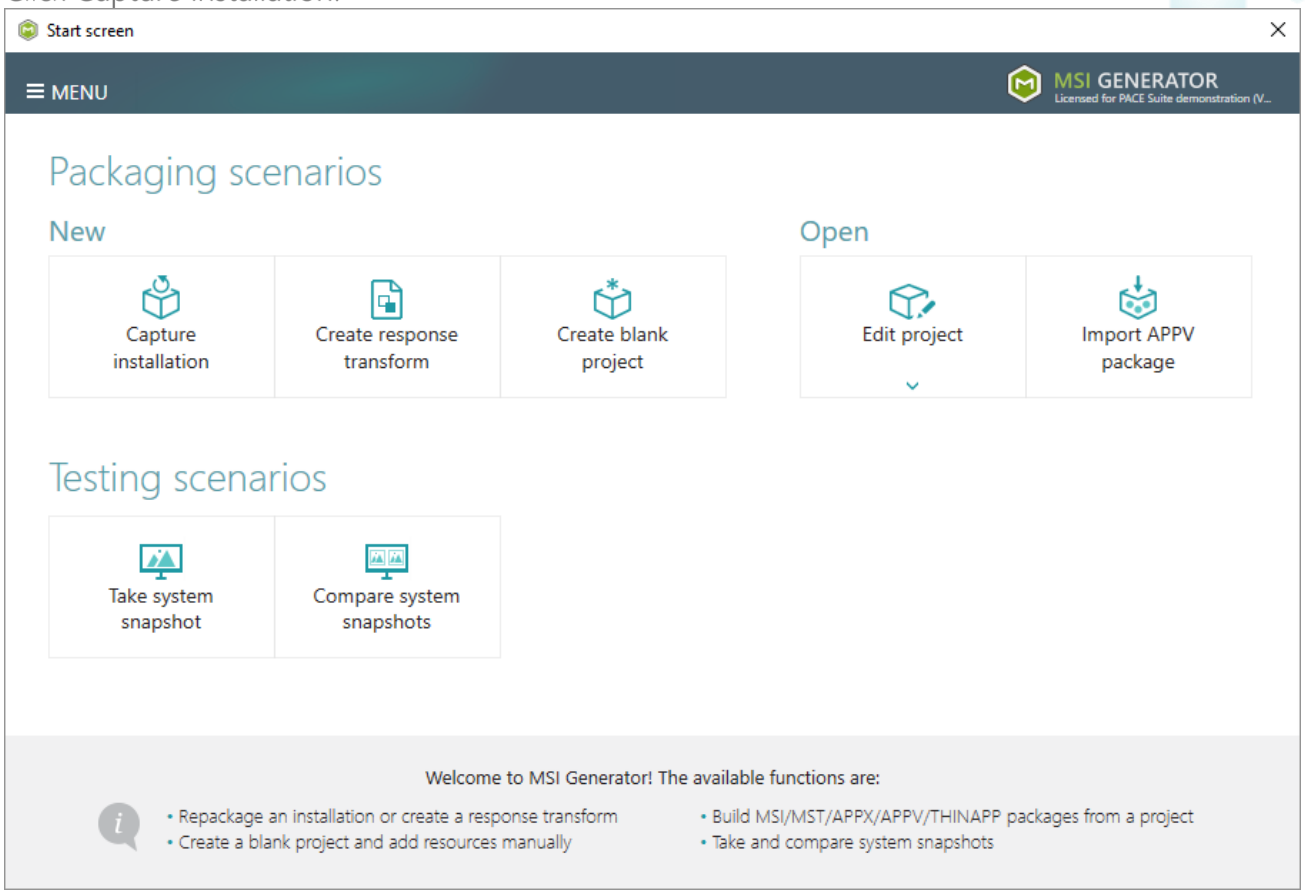
- Launch MSI Generator from the desktop of the start menu shortcut.



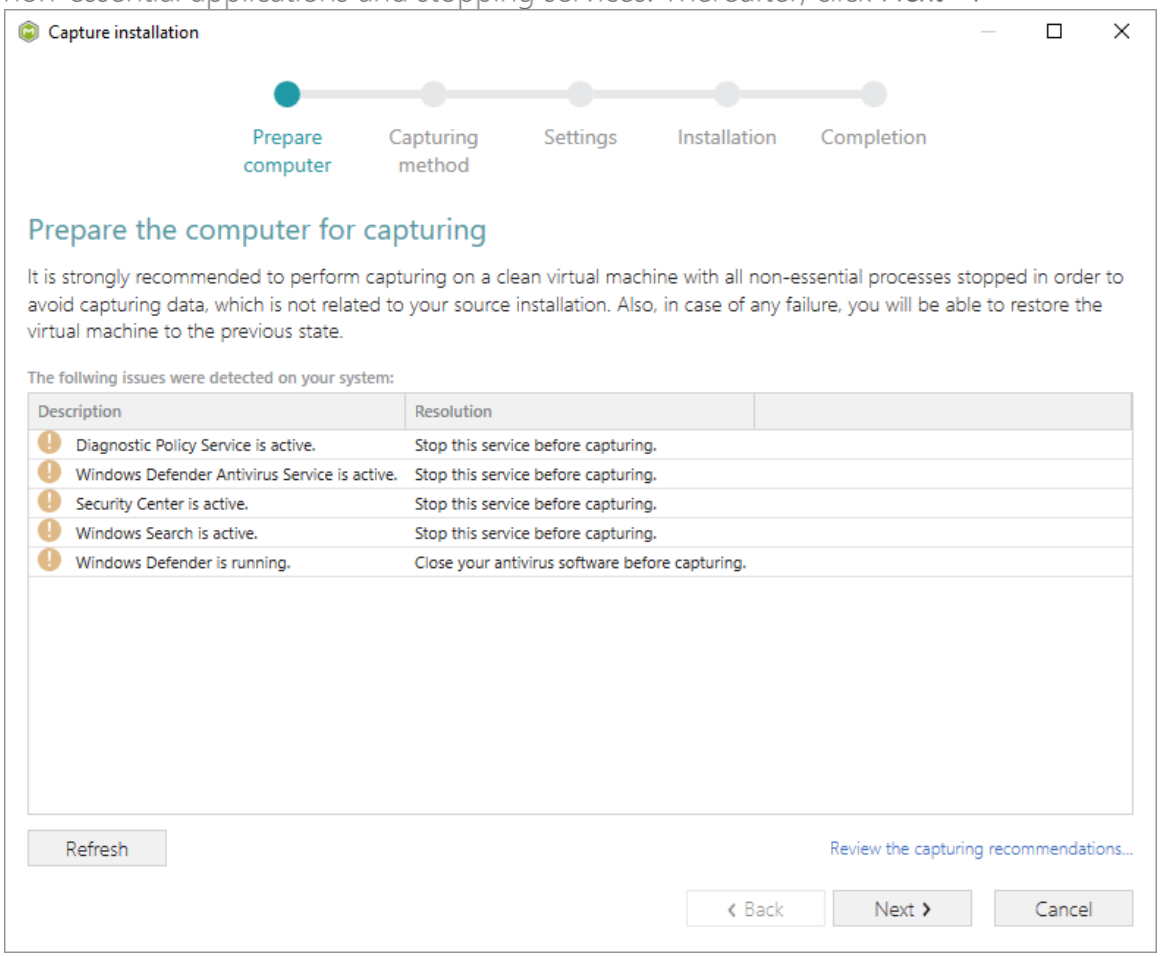
- If you have the User Account Control enabled, click Yes to launch the application.



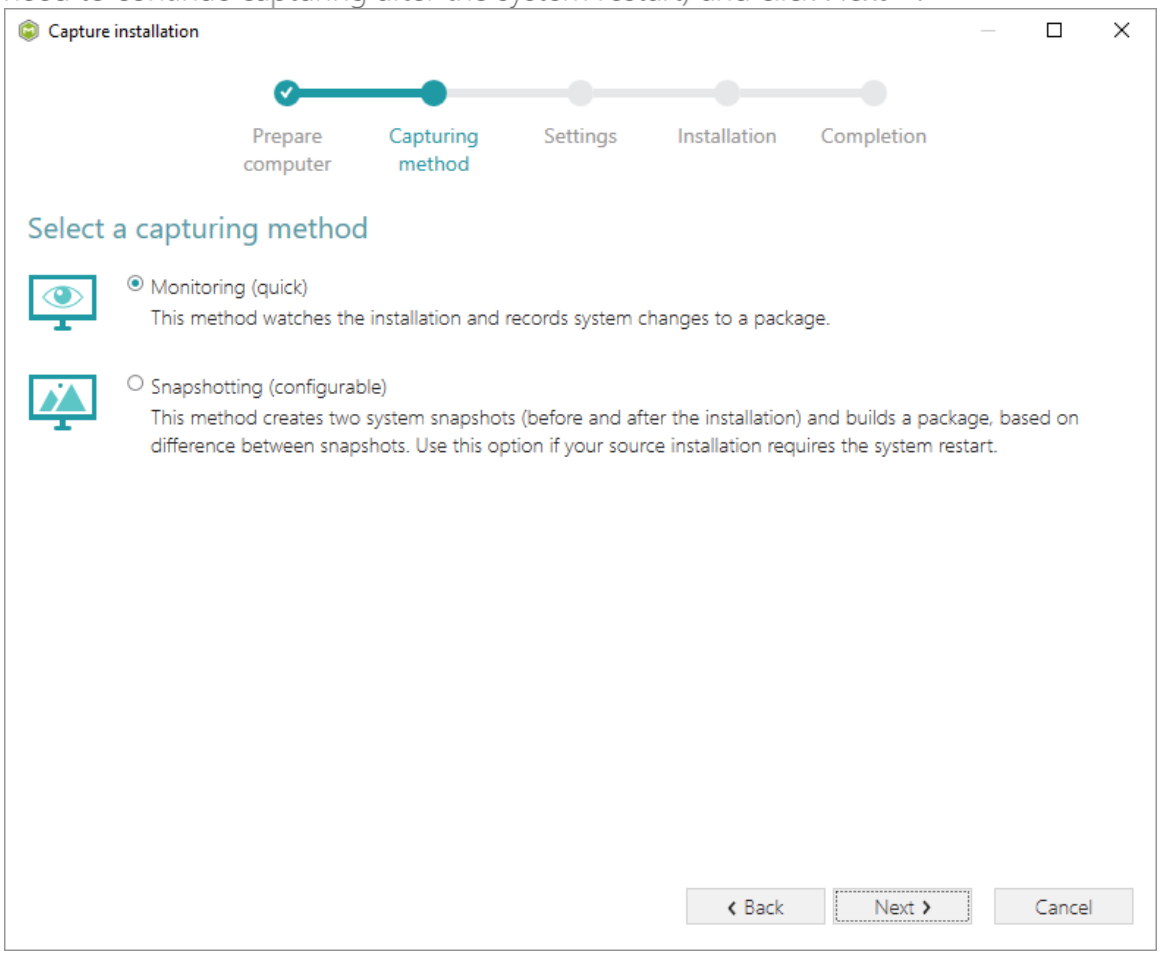
- Click Capture installation.



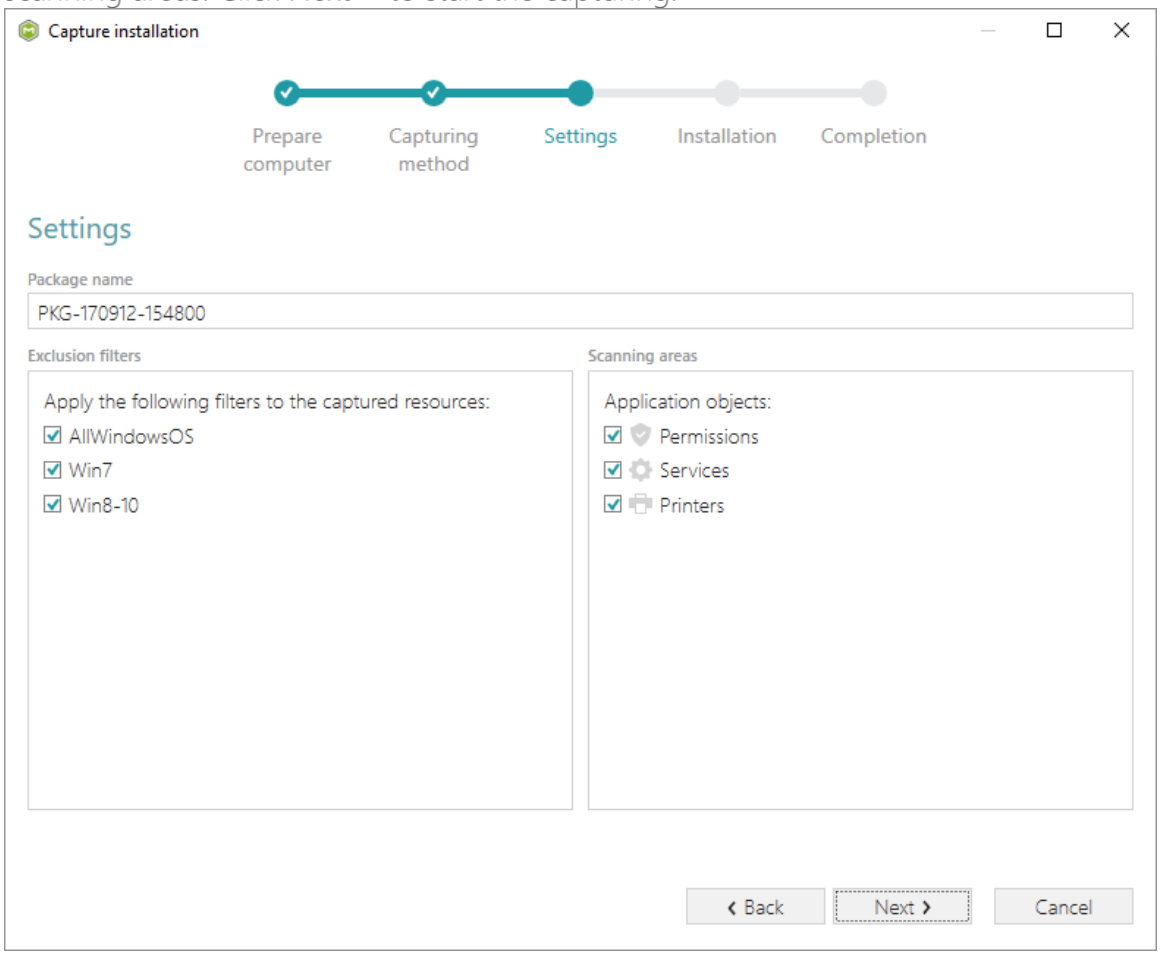
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



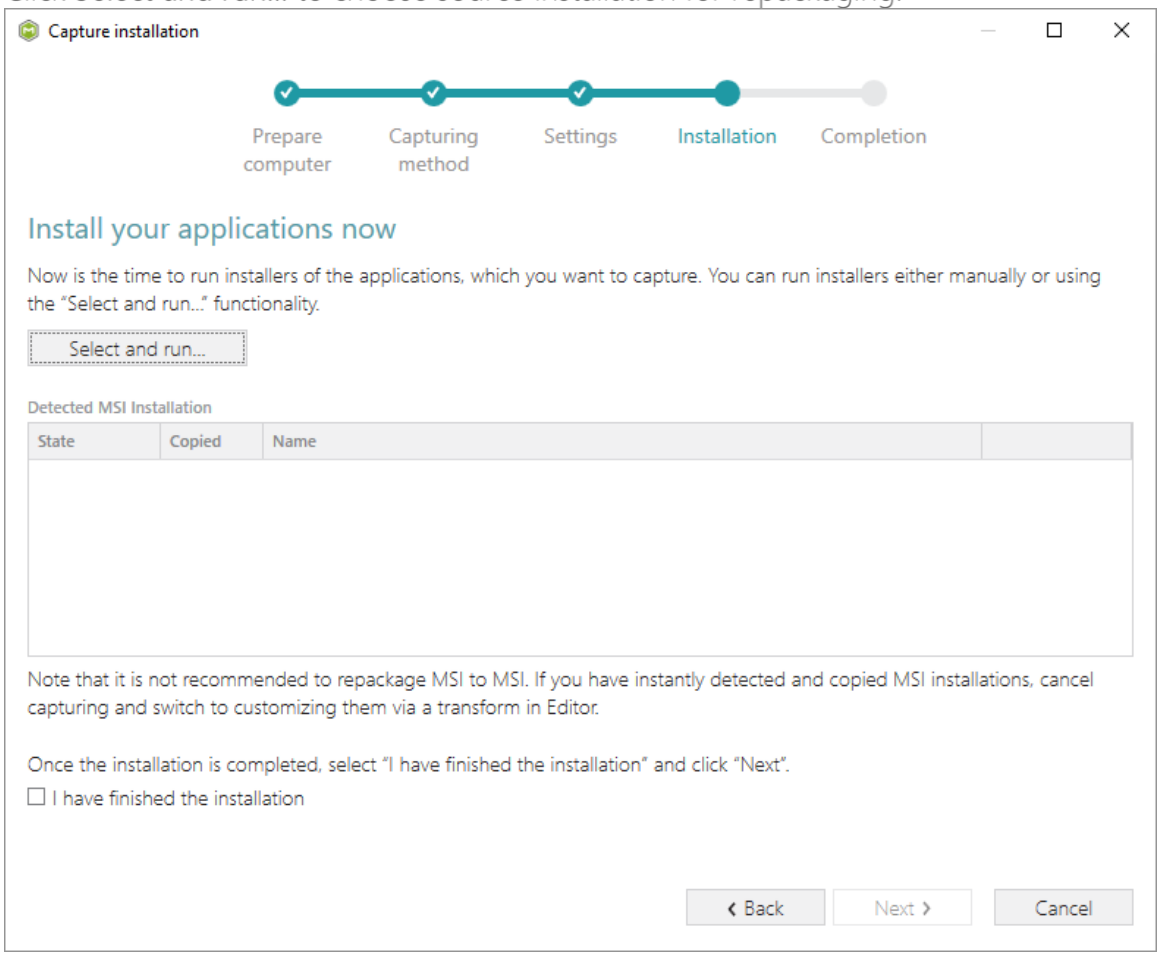
- Select the **Monitoring** method for the quicker capturing (or use the **Snapshotting** one if you need to continue capturing after the system restart) and click **Next >**.



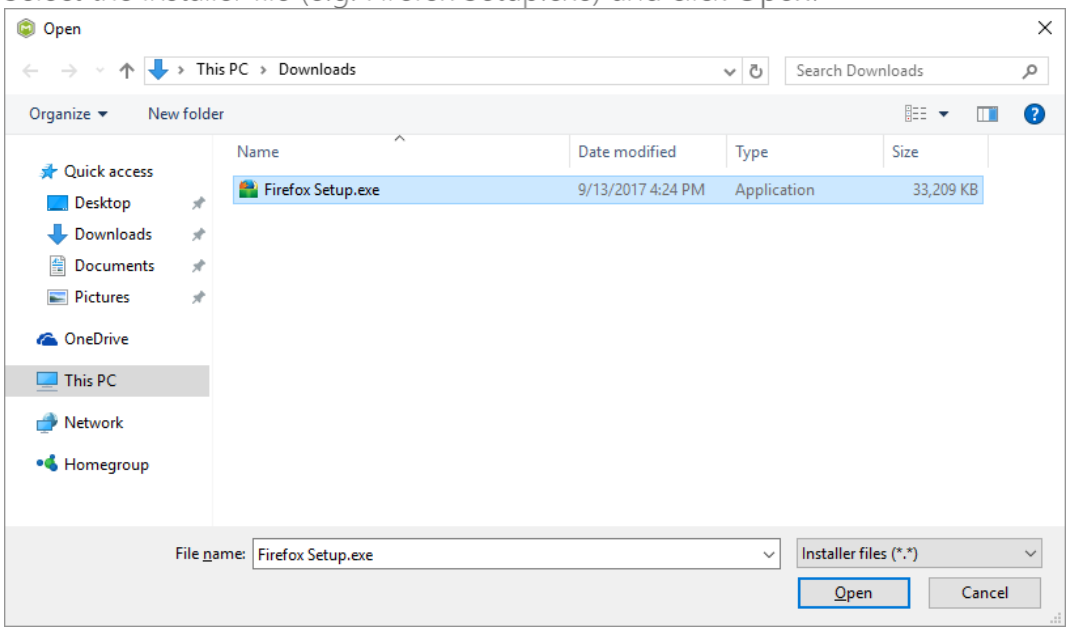
- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.



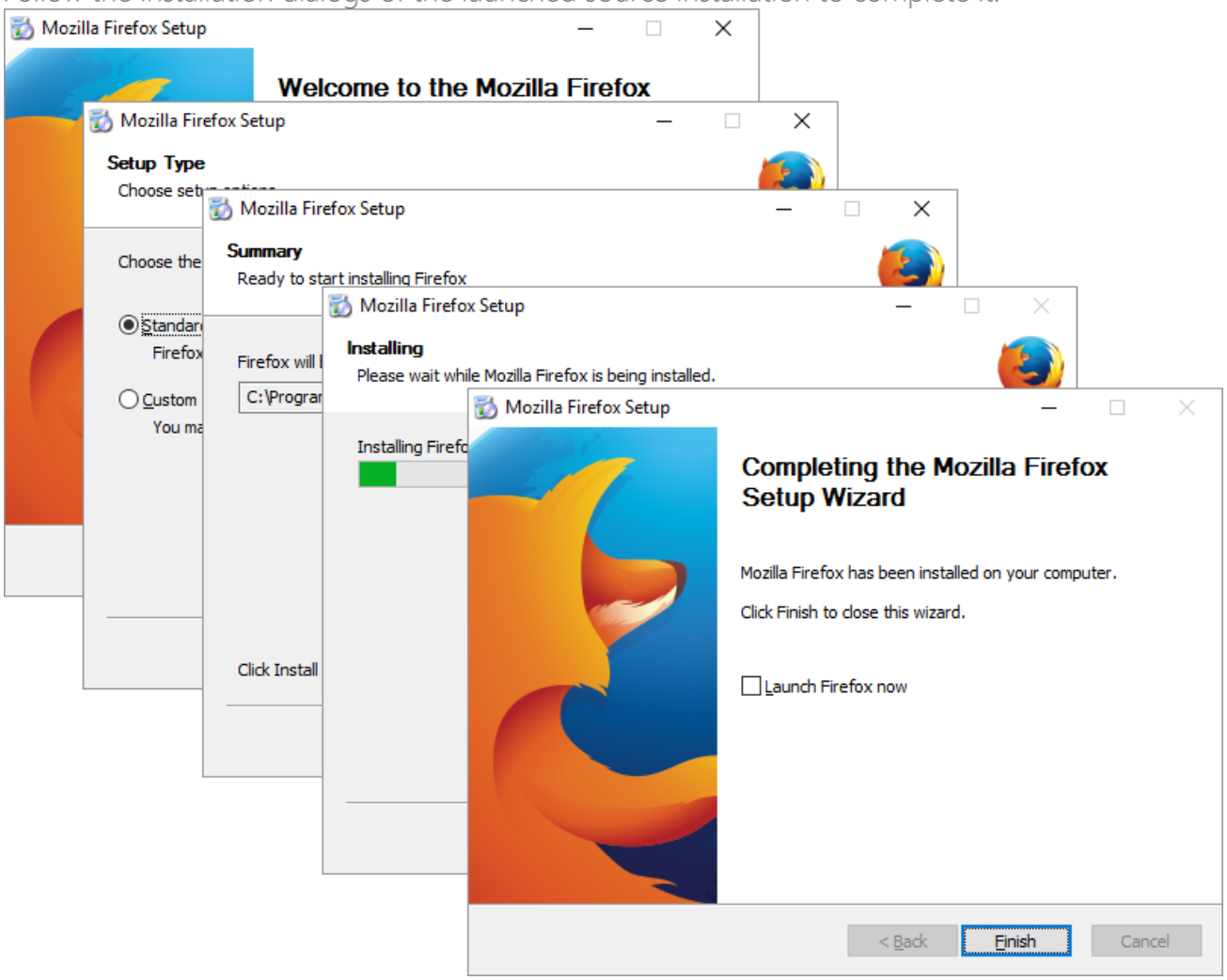
- Click Select and run... to choose source installation for repackaging.



- Select the installer file (e.g. Firefox Setup.exe) and click Open.

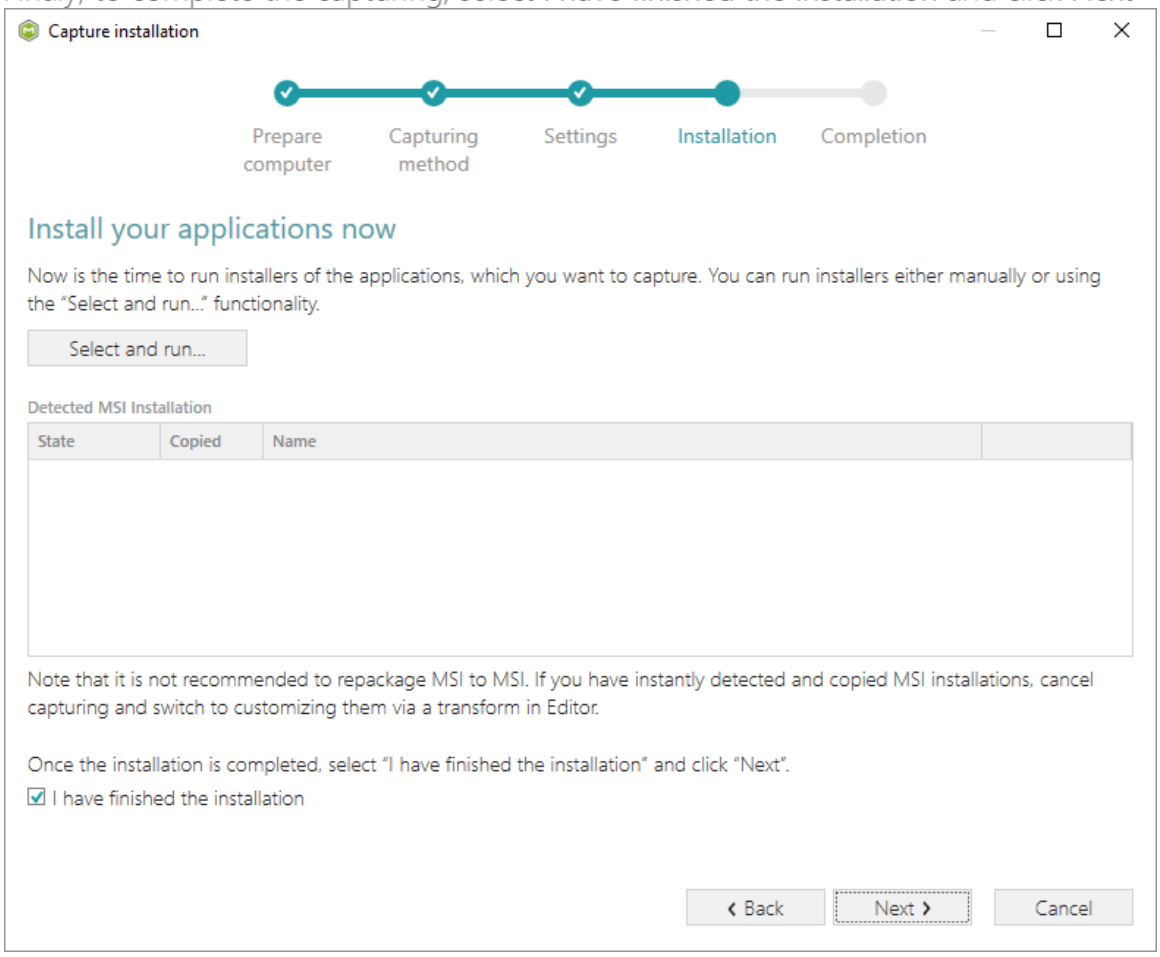


- Follow the installation dialogs of the launched source installation to complete it.

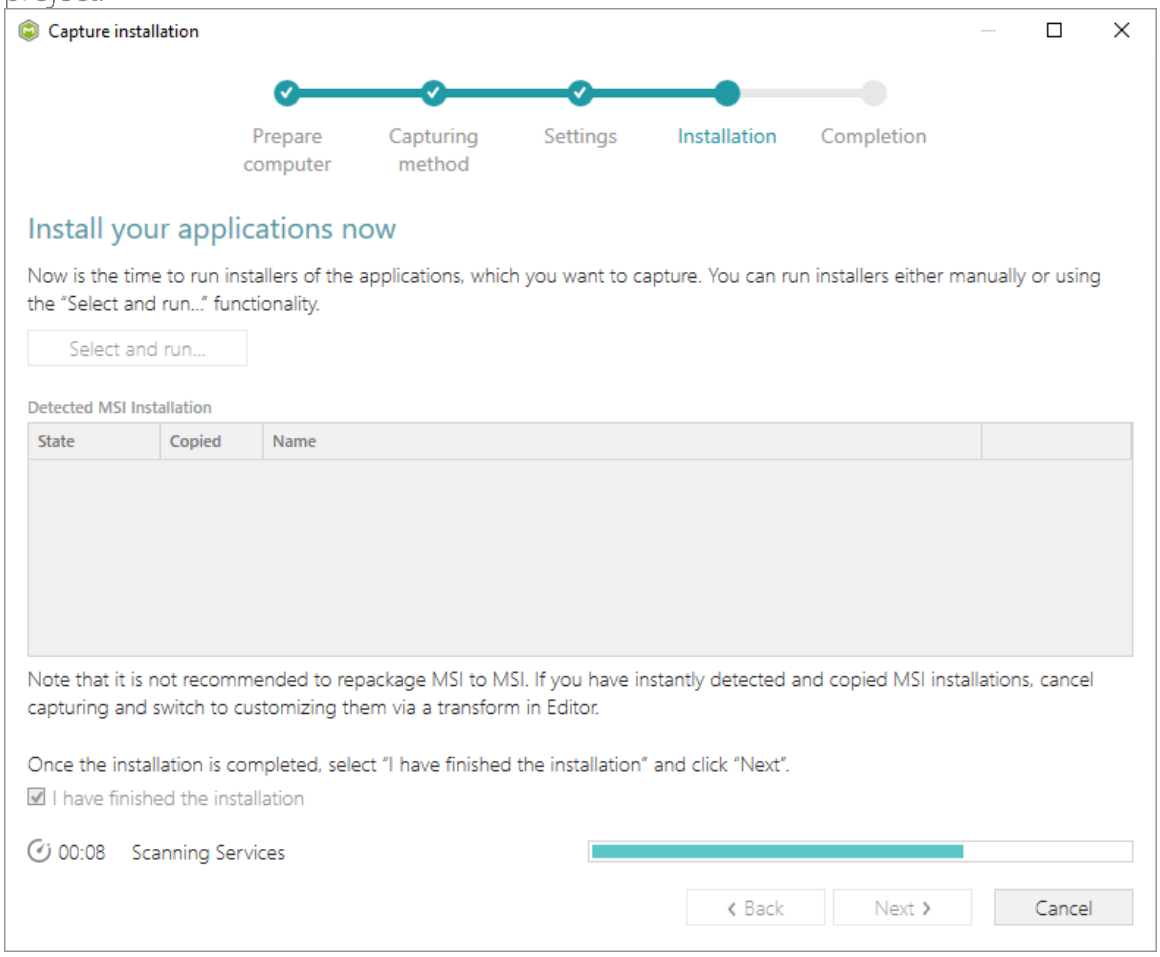


- Now you can make any additional changes to the file system and registry, which you want to capture along with source installation and include to the package. For instance, you can copy or edit configuration files, import REG file to the system registry, change file, folder permissions, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.

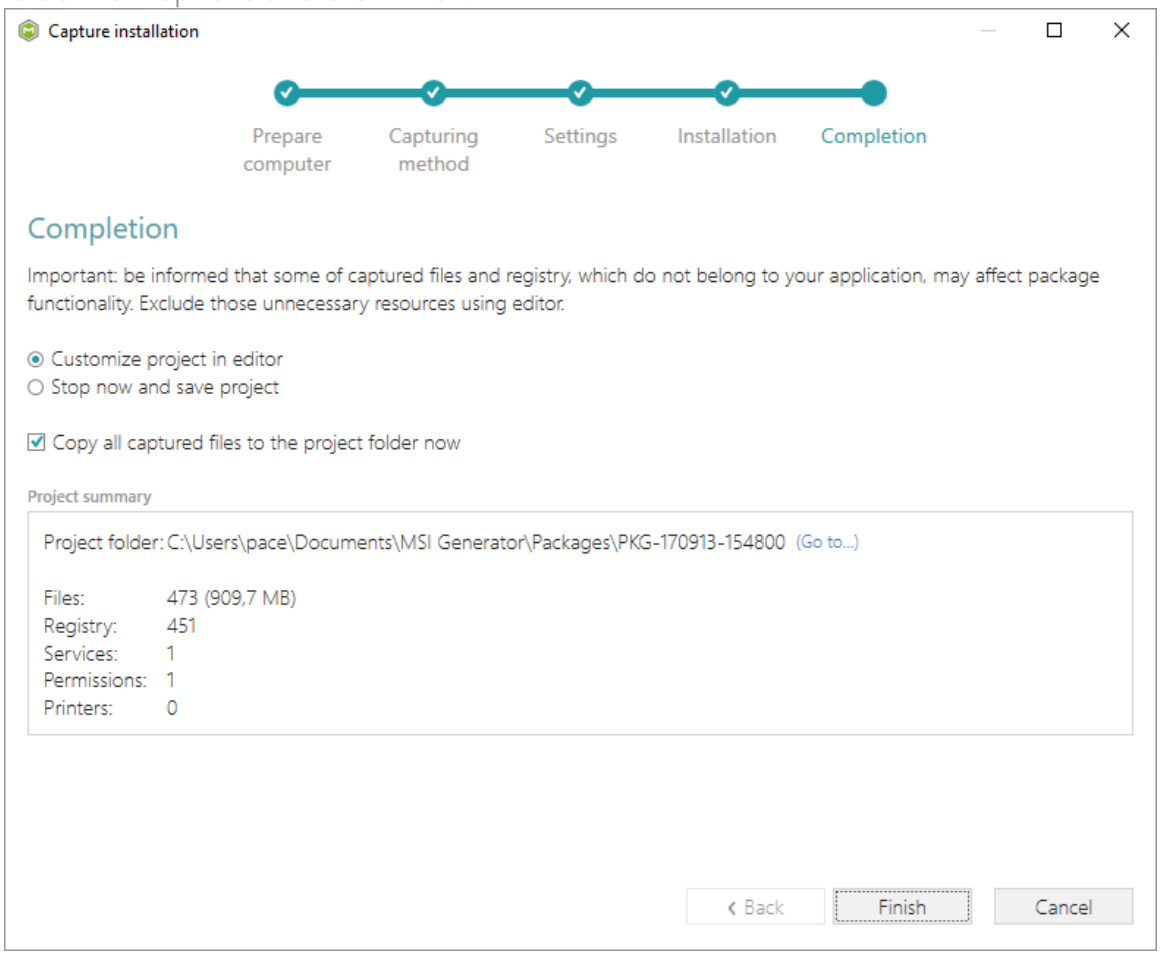
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



- Wait a little, while the capturing process is finishing, filtering captured data and creating the project.

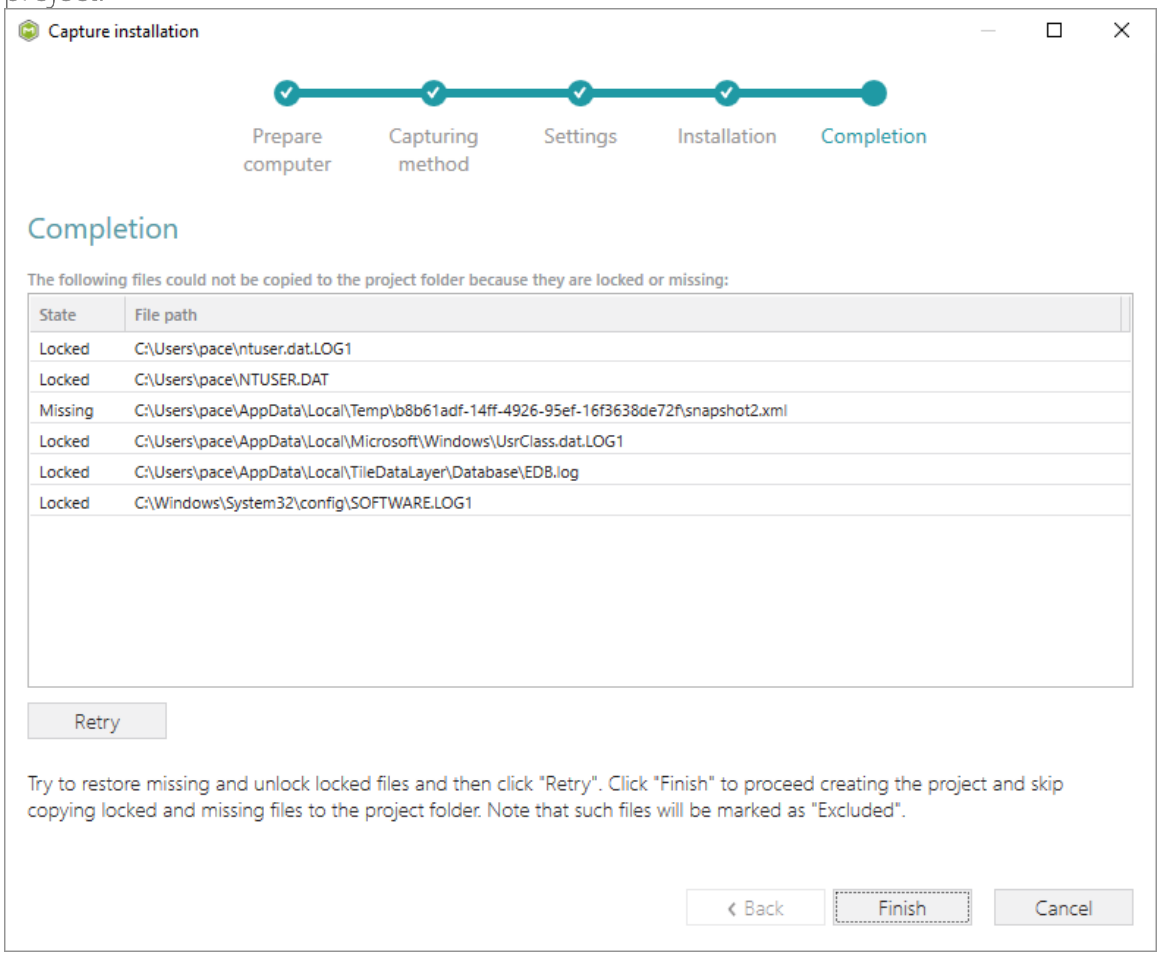


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



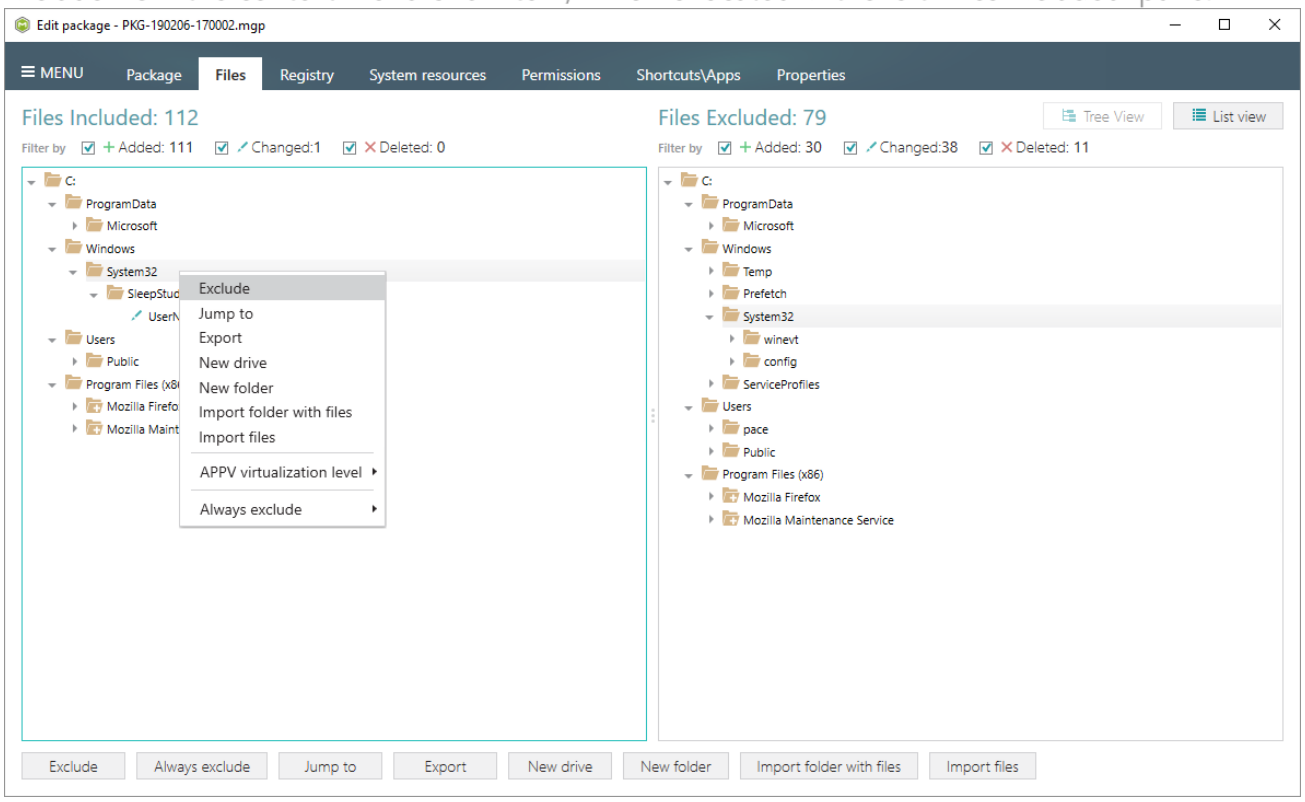
- If some of captured files are locked or missing and as result they could not be copied to the project folder, they will be displayed as illustrated below. Usually, the displayed files are system noise and do not refer to the captured application. Click Finish to continue with skiping locked and missing files. Note that such files will be automatically marked as 'Excluded' in your

project.

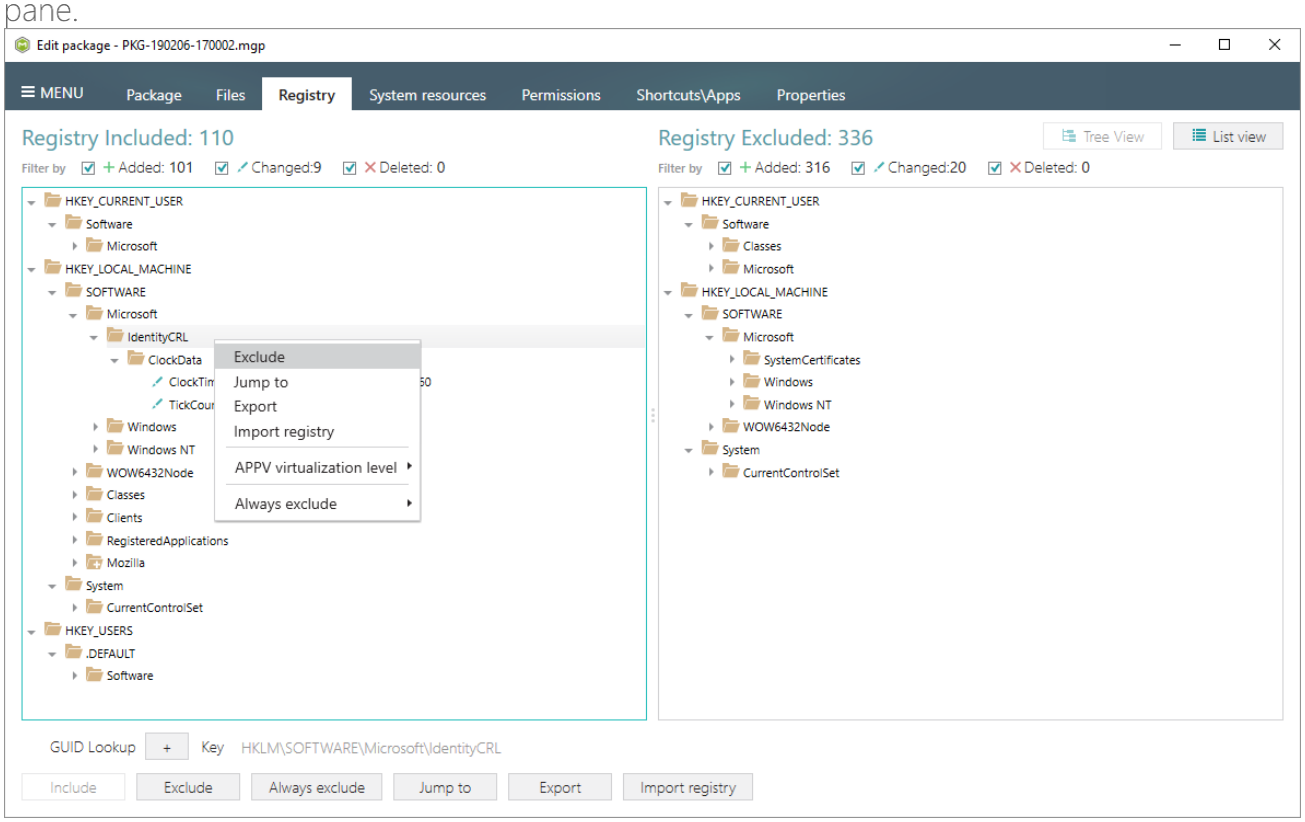


- Once the project is opened in the project editor, it is strongly recommended to review the Files, Registry, System resources tabs and exclude the accidentally captured resources, which do not belong to the application. Such accidentally captured resources are also called system noise and it could be Windows Updates, Search indexing, Antivirus software and so on.

- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.



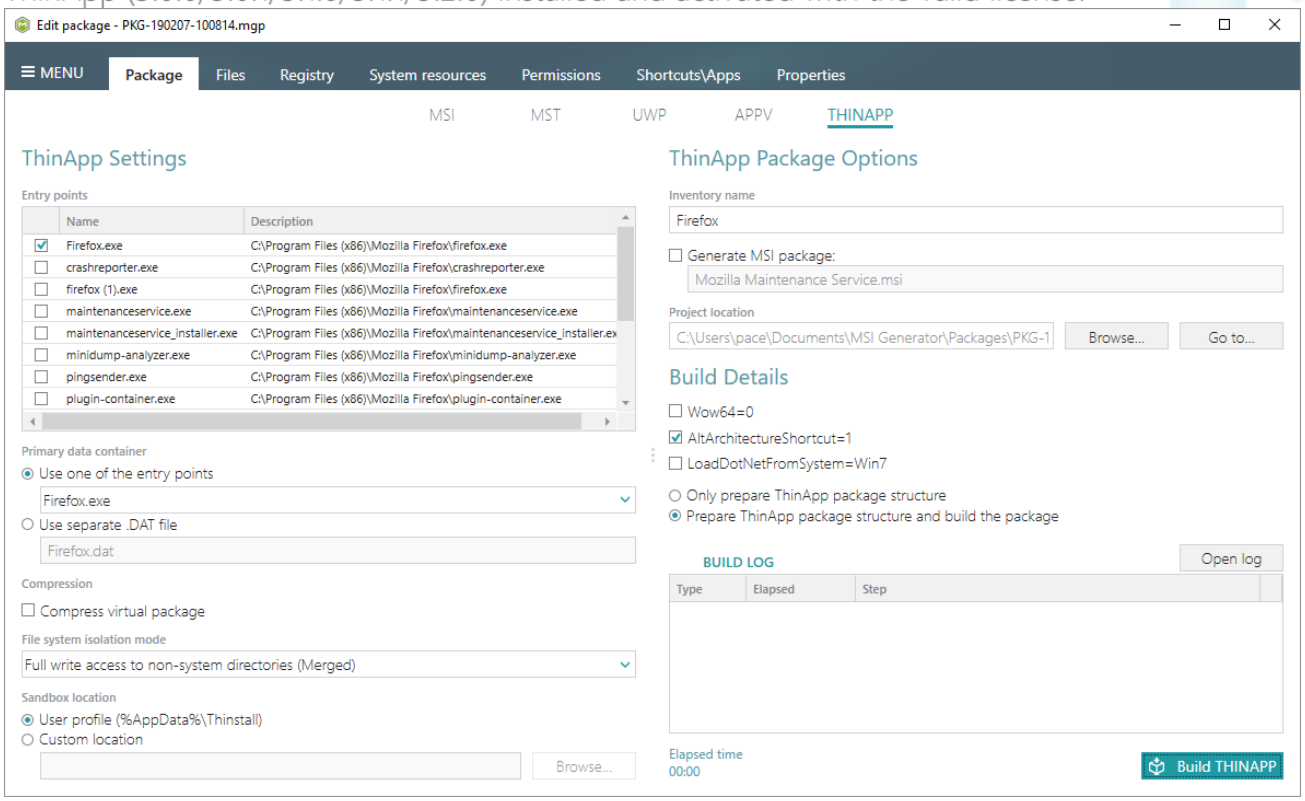
- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.



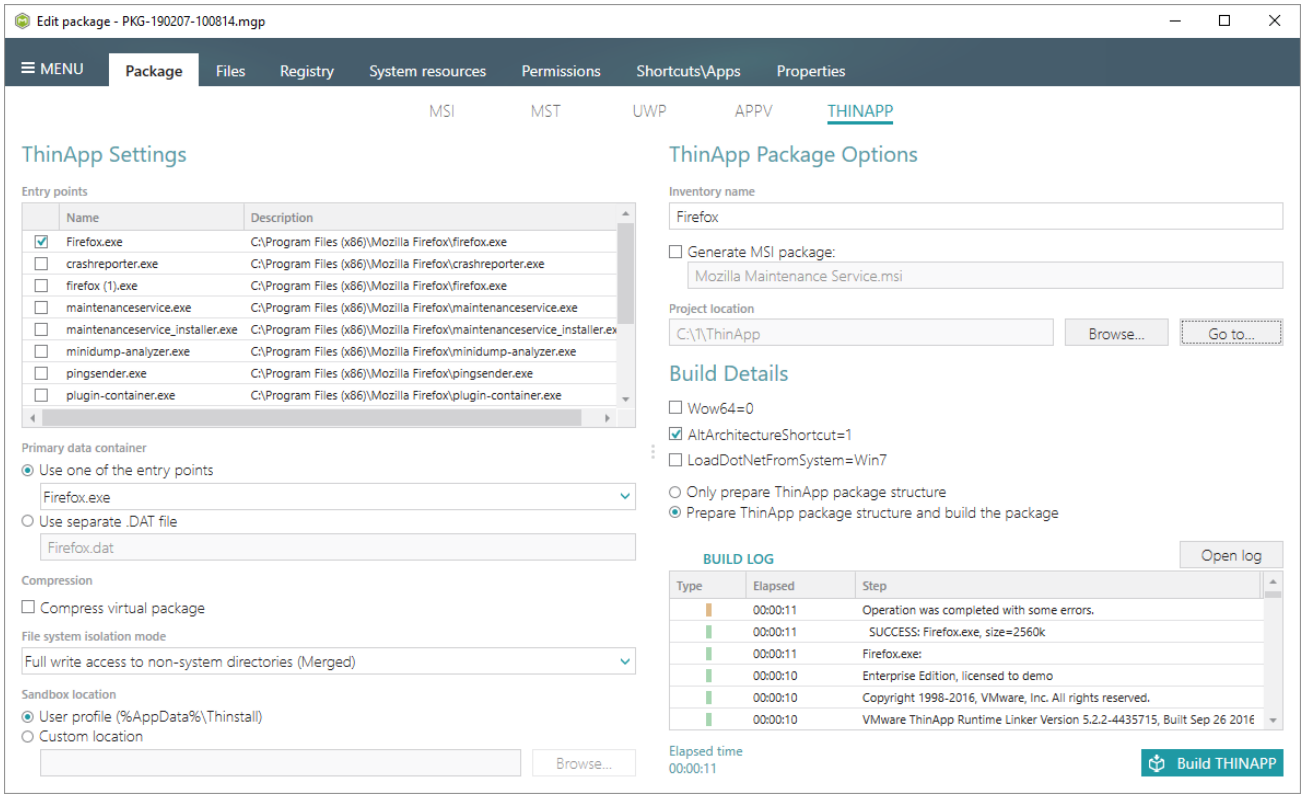
- For managing **THINAPP Settings** such as entry points, primary data container, compression, file system isolation mode, sandbox location and the like, see section 3.8.9

- Finaly, to generate a THINAPP package navigate to the **Package** -> **THINAPP** tab and click **Build THINAPP**. NOTE For building THINAPP packages, PACE Suite requires the VMware

ThinApp (5.0.0/5.0.1/5.1.0/5.1.1/5.2.0) installed and activated with the valid license.



- Once the package is built, click Go to..., located next to the Project location field, to open the package-containing folder in Windows Explorer. Find the generated package in the bin subfolder.



3.7 Edit MSI/MST

NOTE Adhere to the following Windows Installer Best Practices during editing your package.

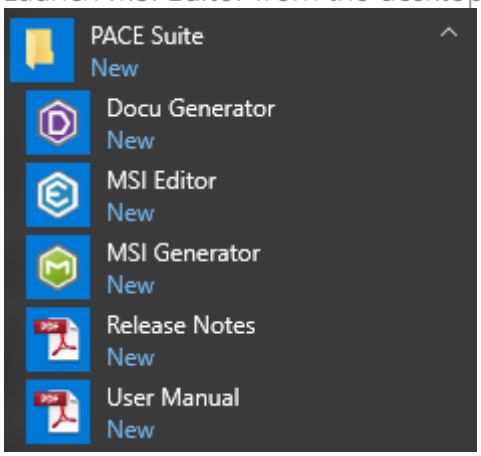
Choose a scenario that better suits your needs:

- For managing Application Details see section 3.7.2
  Manage product name, version, language and manufacturer of the application.
- For managing Package IDs see section 3.7.3
  Manage product, upgrade and package codes.
- For managing Files and Folders see section 3.7.4
  Review package files and folders, add new folders, import files, set permissions, register COM objects and .NET assemblies, set the INSTALLDIR folder and the like.
- For managing Shortcuts see section 3.7.5
  Review package shortcuts, add new or import existing shortcuts, manage their properties.
- For managing Registry see section 3.7.6
  Review package registry, add new or import from REG file registry keys and values, integrate ActiveSetup, set permissions and the like.
- For managing Custom Actions see section 3.7.7
  Manage custom and standard actions of the package, add new one to run scripts, EXE, etc.
- For managing Drivers see section 3.7.8
  Review the package device drivers, modify and add new ones.
- For managing Features and Components see section 3.7.9
  Review the feature-component structure and resources, linked with components. Add new features, components and manage their properties.
- For managing ARP Appearance see section 3.7.10
  Control how the package will be displayed in the App list (Add & Remove Programs).
- For managing Upgrades see section 3.7.11
  Manage the package upgrade settings to automatically upgrade previous MSI packages.
- For managing MSI Properties see section 3.7.12
  Manage package properties, defined in the Property table and those, which will be updated by the CustomAction, AppSearch, Control, ComboBox, CheckBox, RadioButton tables.
- For managing Summary Information see section 3.7.13
  Manage Summary Information of the opened MSI/MST.
- For managing MSI Tables see section 3.7.14
  Add, import, drop MSI tables.
- For managing Merge Modules, described in section 3.7.15
  Create new MSM or importing existing MSM to the package.

3.7.1 Pre-Condition

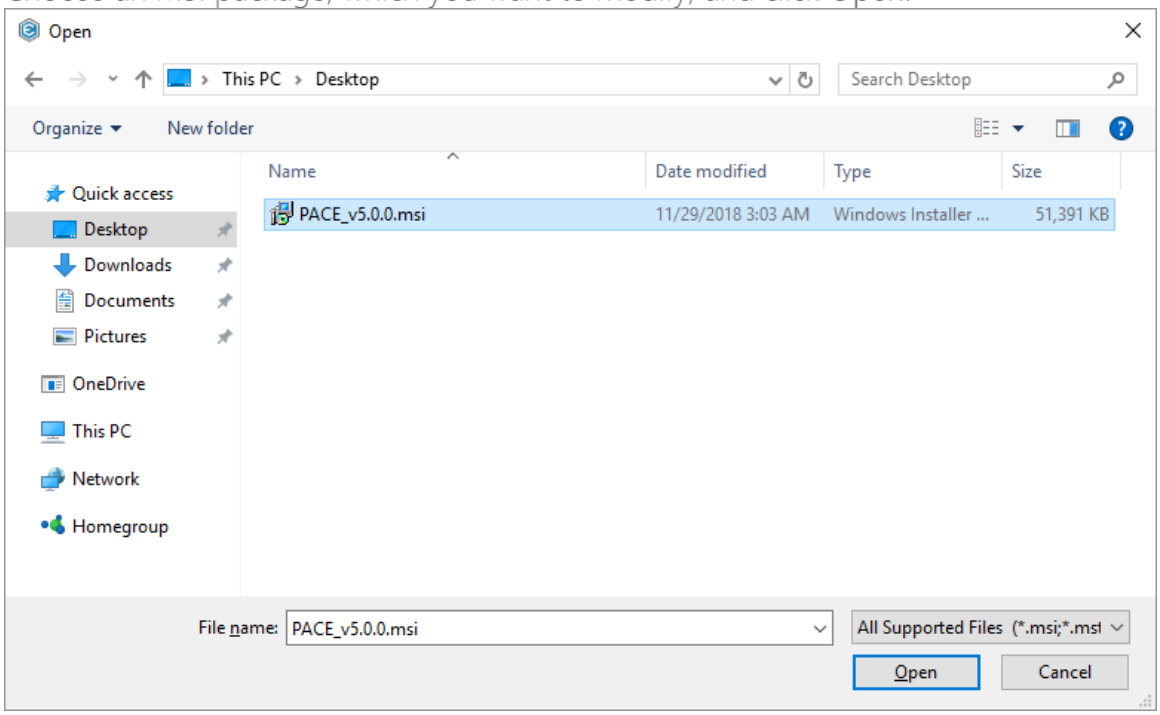
For editing MSI/MST open the file as described below:

- Launch MSI Editor from the desktop or the start menu shortcut.

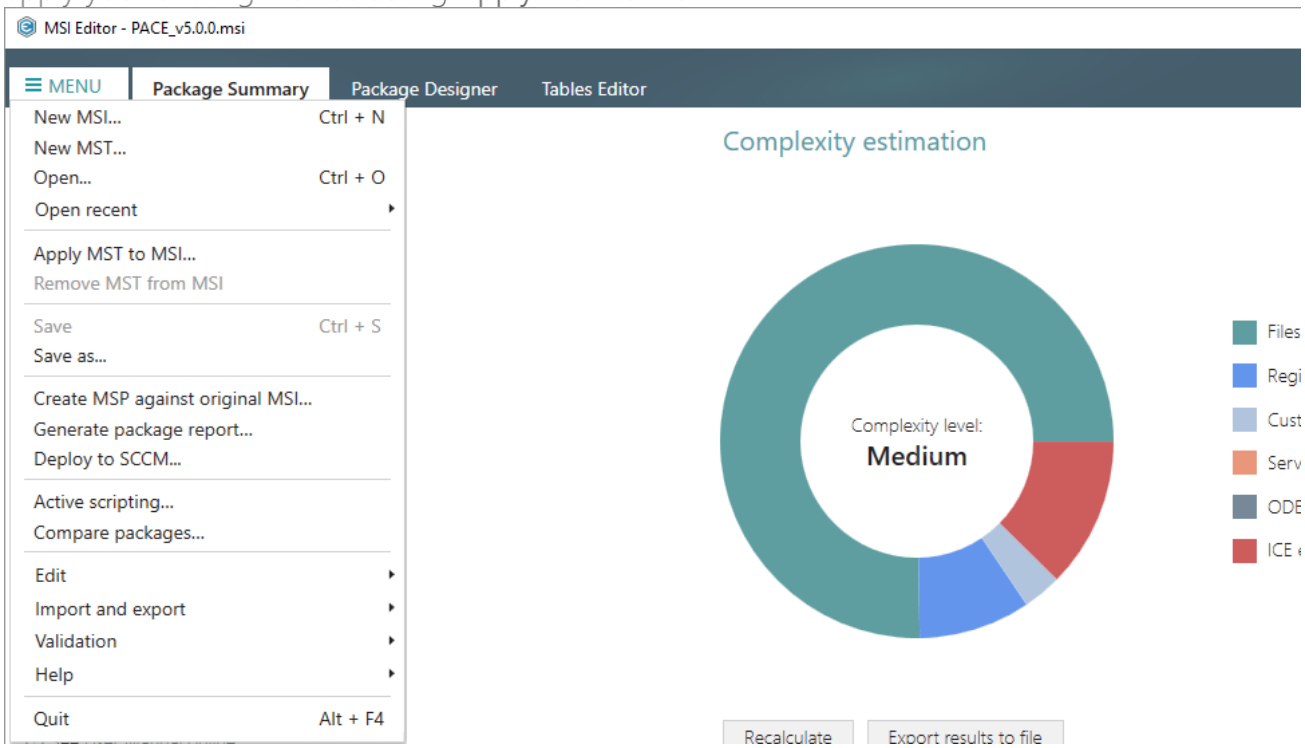


- Select Open... from the MENU.

- Choose an MSI package, which you want to modify, and click **Open**.



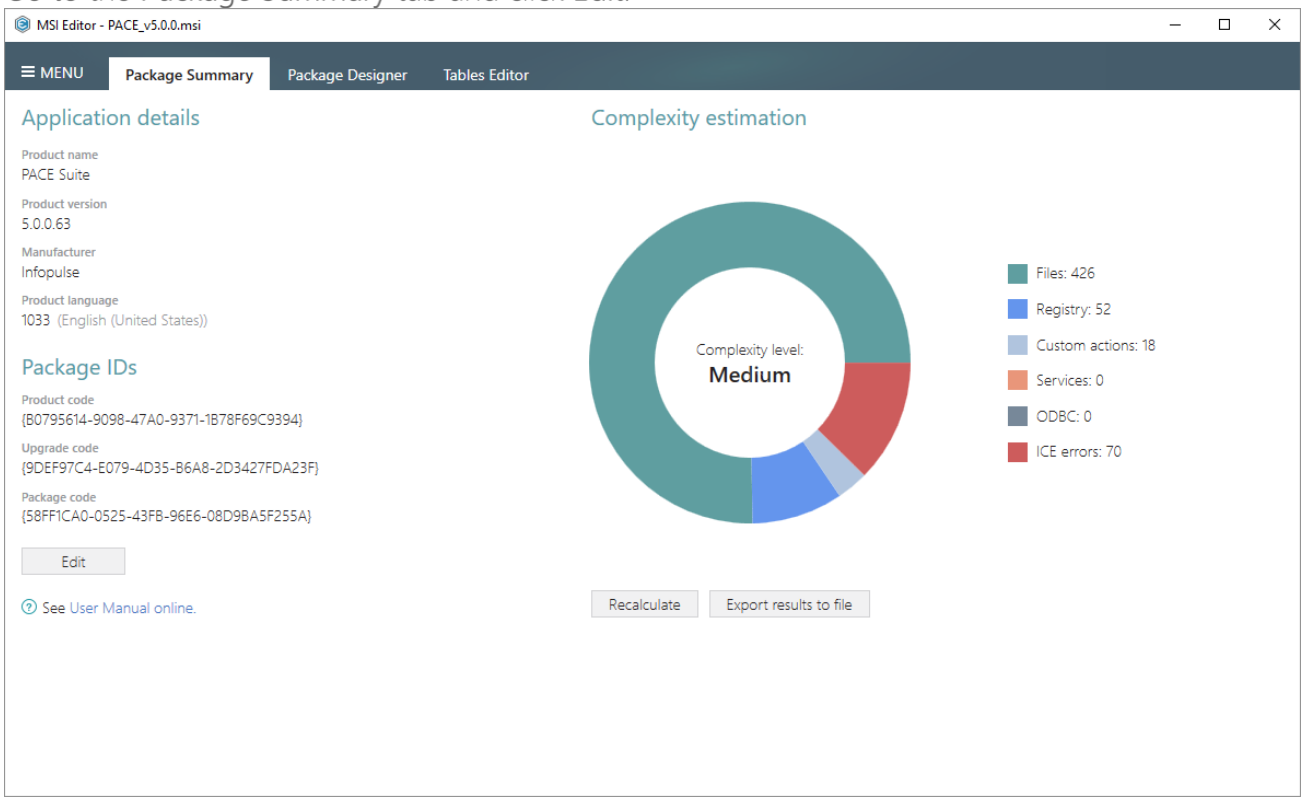
- If you want to modify MSI via MST, select **New MST** from the **MENU** to create a blank MST or apply your existing MST selecting **Apply MST to MSI**.



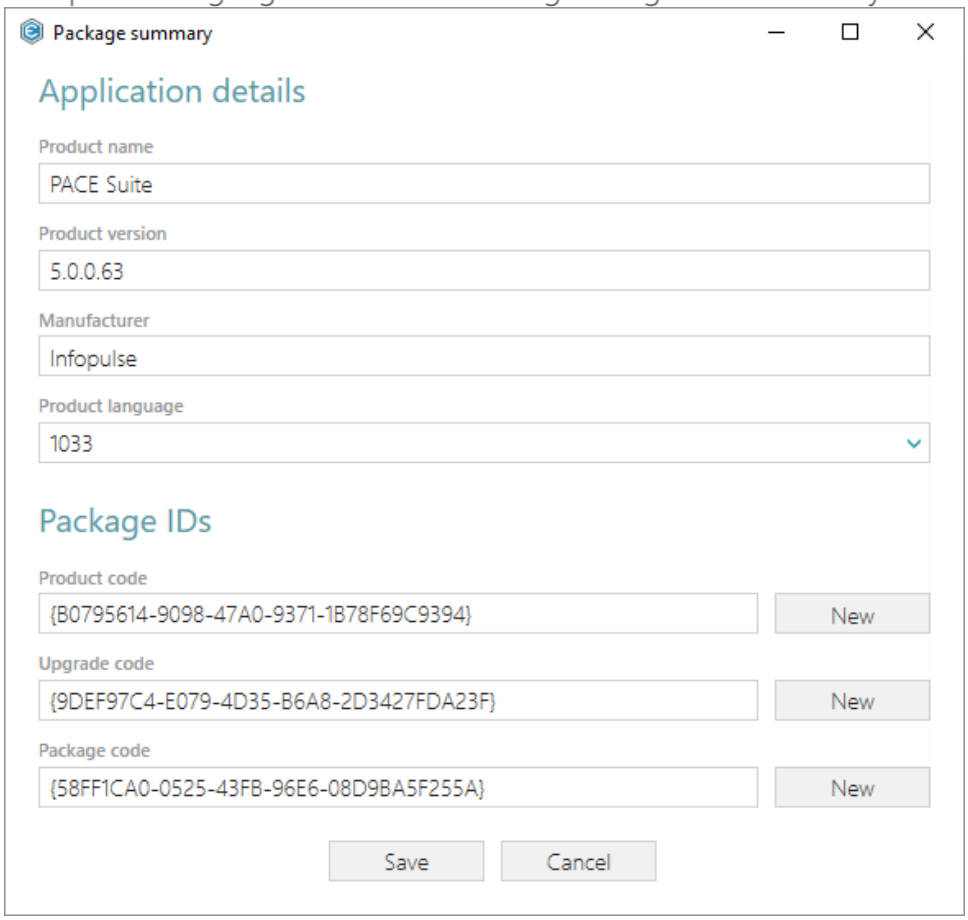
3.7.2 Application Details

Manage product name, version, language and manufacturer of the application.

- Go to the Package Summary tab and click Edit.



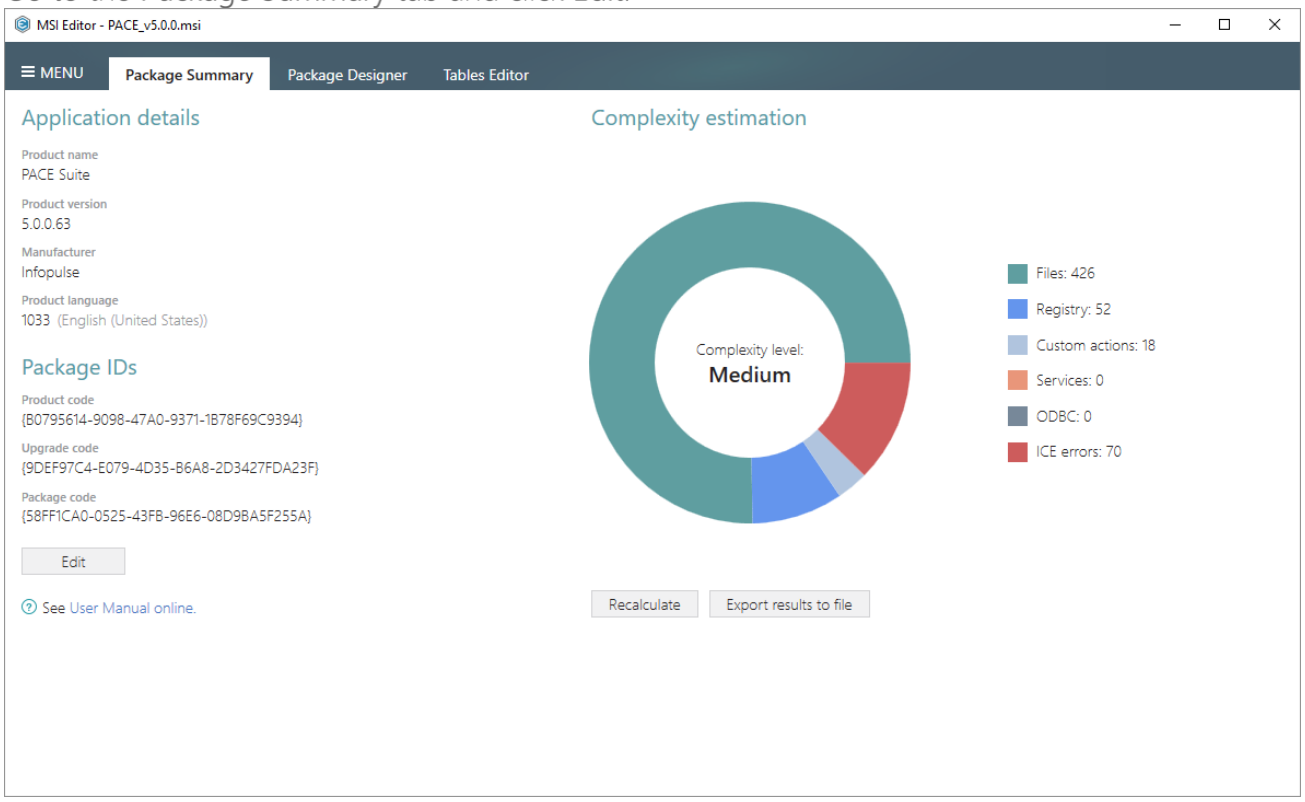
- Update the necessary values of the Application details section and click Save.
  NOTE If you specify a new product language, append this language ID to a value of the Compatible languages field of the Package Designer -> Summary Information tab.



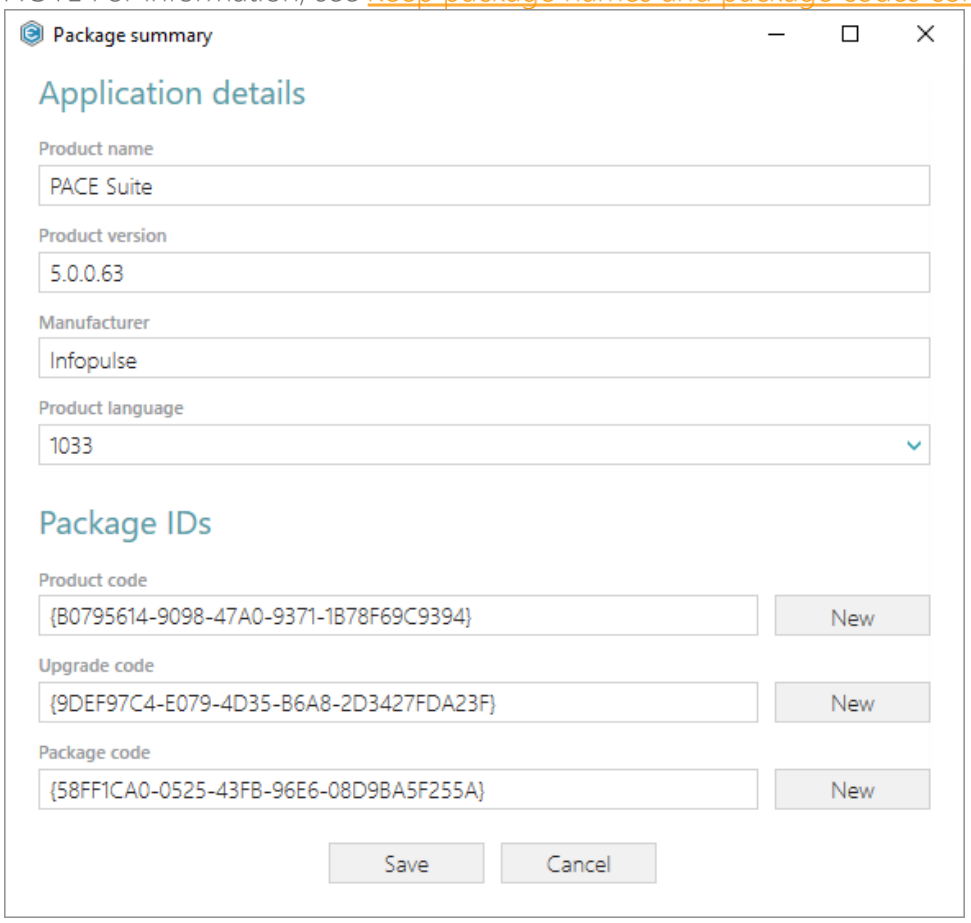
3.7.3 Package IDs

Manage product, upgrade and package codes.

- Go to the Package Summary tab and click Edit.



- Update the necessary values of the Package IDs section and click Save.
NOTE For information, see Keep package names and package codes consistent.



3.7.4 Files and Folders

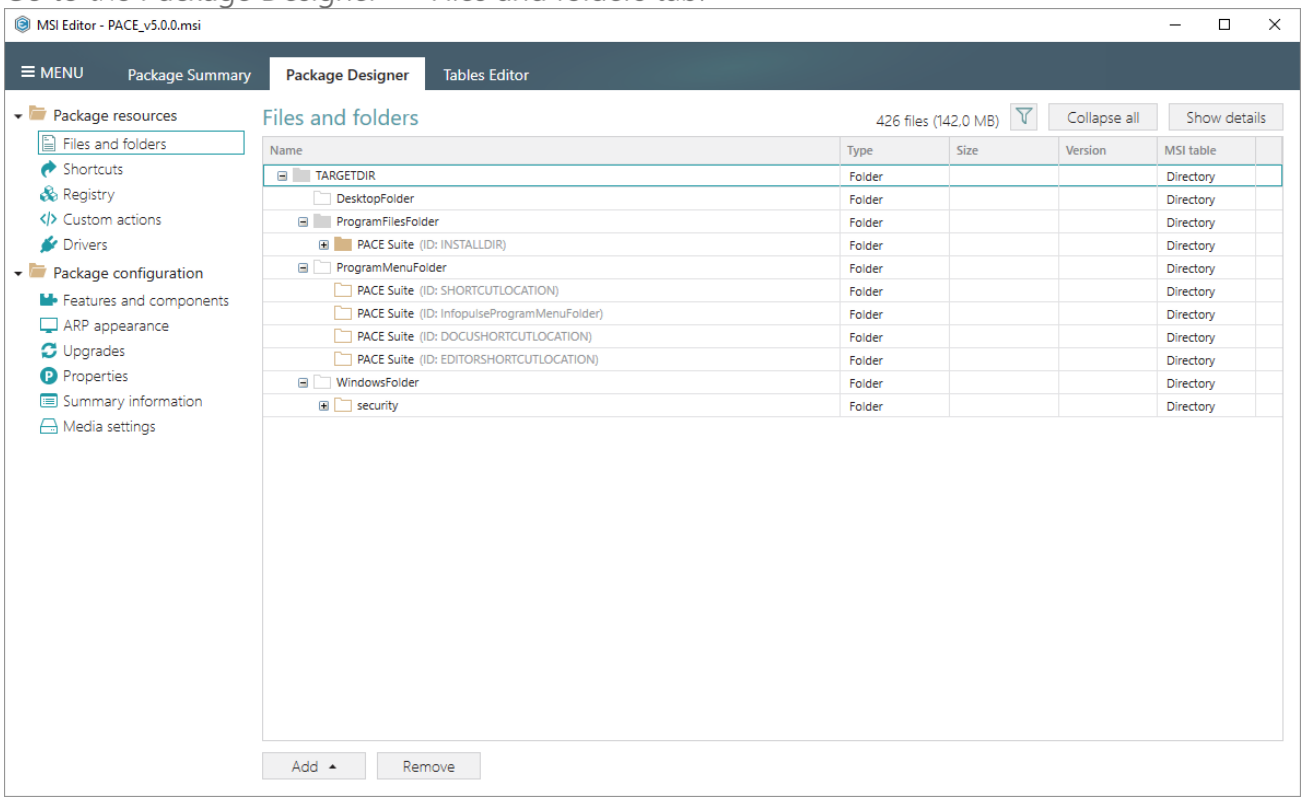
Choose a scenario that better suits your needs:

- Add System Folder, described in section 3.7.4.1
  Add a system (standard) folder to the package.
- Add Subfolder, described in section 3.7.4.2
  Add a subfolder (user-defined folder) to the package.
- Import Files and COM, described in section 3.7.4.3
  Import files along with their, extracted on-the-fly, COM information.
- Change Compression Type, described in section 3.7.4.4
  Change the compression type for all package files by rebuilding the whole package.
- Set Permissions, described in section 3.7.4.5
  Set custom permissions for a file or folder.
- Remove Permissions, described in section 3.7.4.6
  Remove the captured or manually set custom permissions from a file or folder.
- Move Files, Folders, described in section 3.7.4.7
  Move a file or a folder from one folder to another.
- Set INSTALLDIR, described in section 3.7.4.8
  Set a subfolder (user-defined folder) of the package as the INSTALLDIR.
- Control Empty Folder Creation, described in section 3.7.4.9
  Force the empty folder creation on install.
- Control File/Folder Removal, described in section 3.7.4.10
  Force folder and its files removal on install or uninstall.
  NOTE By default, MSI removes only those files, which it installs. This means that files, created during the application work, will remain on the system.
- Edit INI File, described in section 3.7.4.11
  Edit a name, destination location and content of the INI file, saved to the IniFile table.
- Edit File Properties, described in section 3.7.4.12
  Edit a name, identifier, destination location, component and many other attributes of the file, saved to the File table.
- Edit Folder Properties, described in section 3.7.4.13
  Edit a folder target and source names, identifier, and its parent folder.

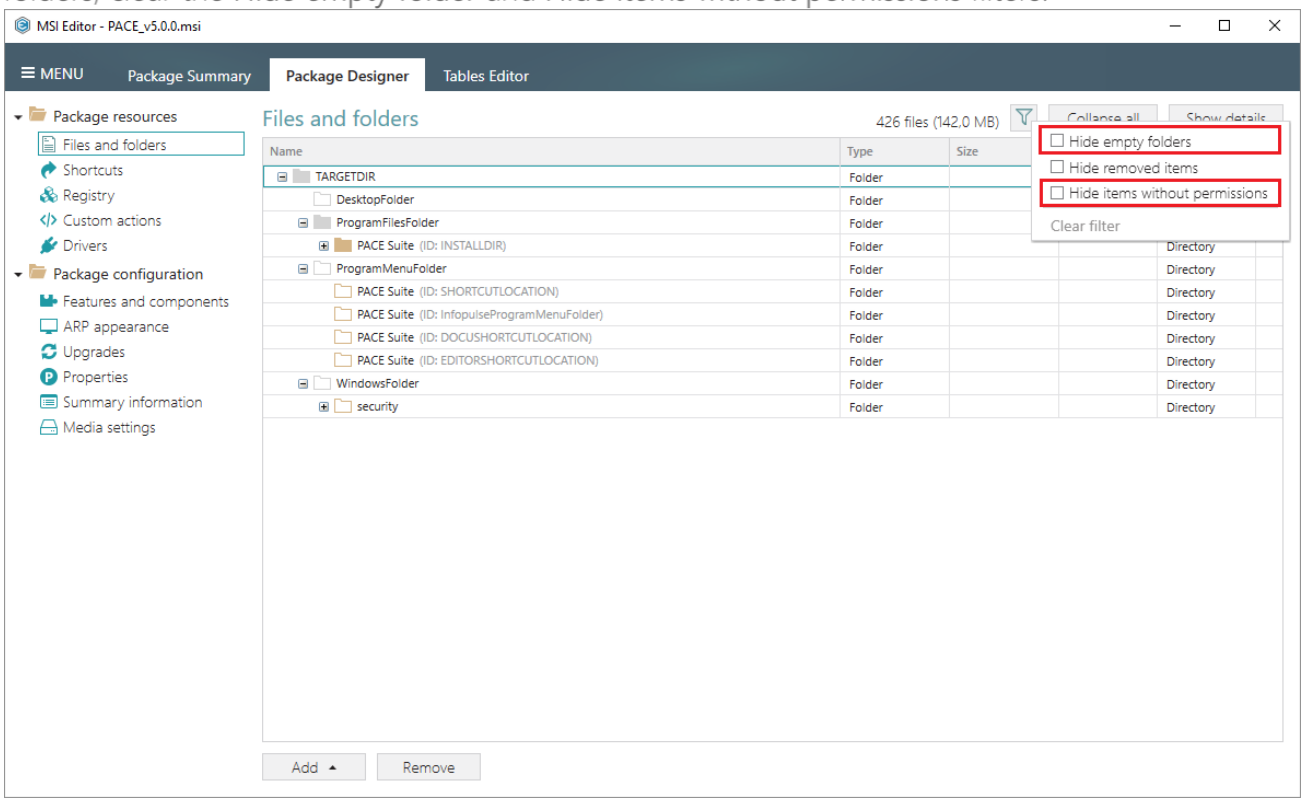
3.7.4.1 Add System Folder

Add a system (standard) folder from the list into the package.

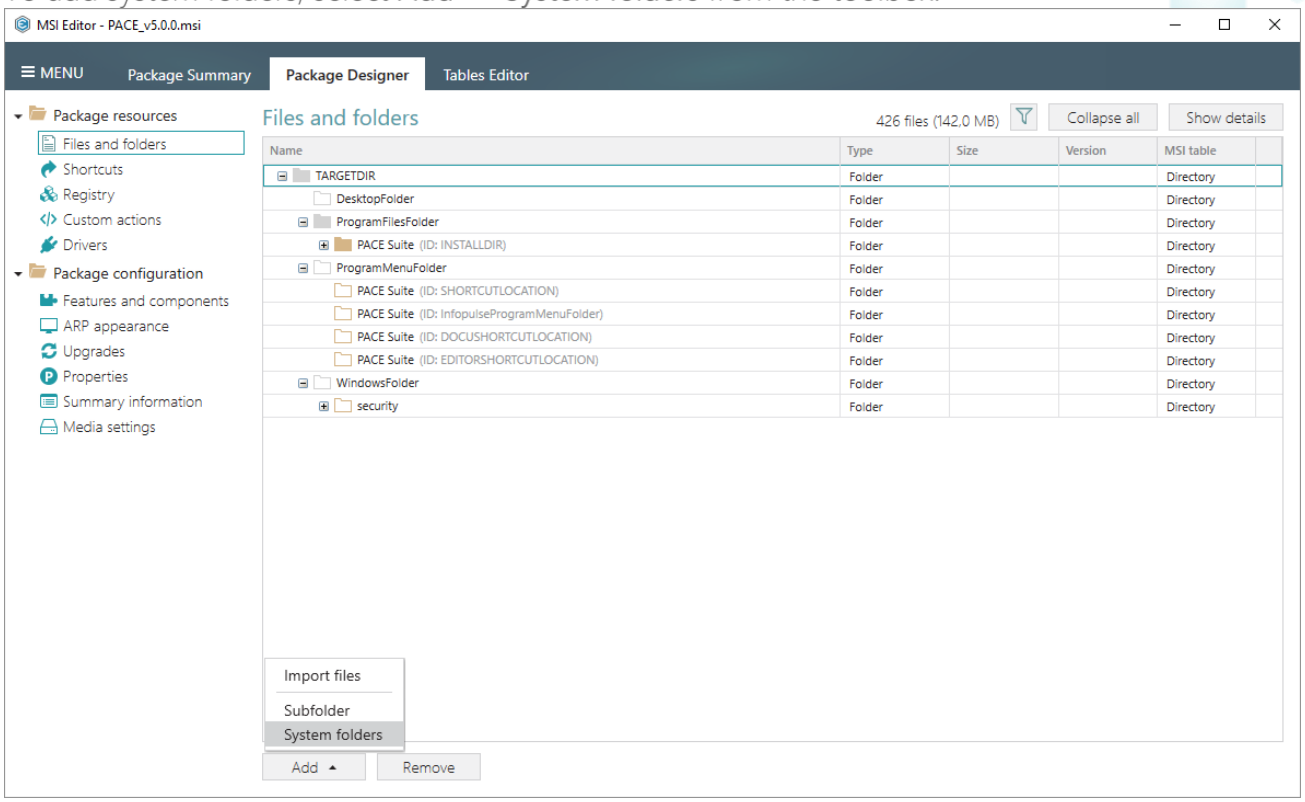
- Go to the Package Designer -> Files and folders tab.



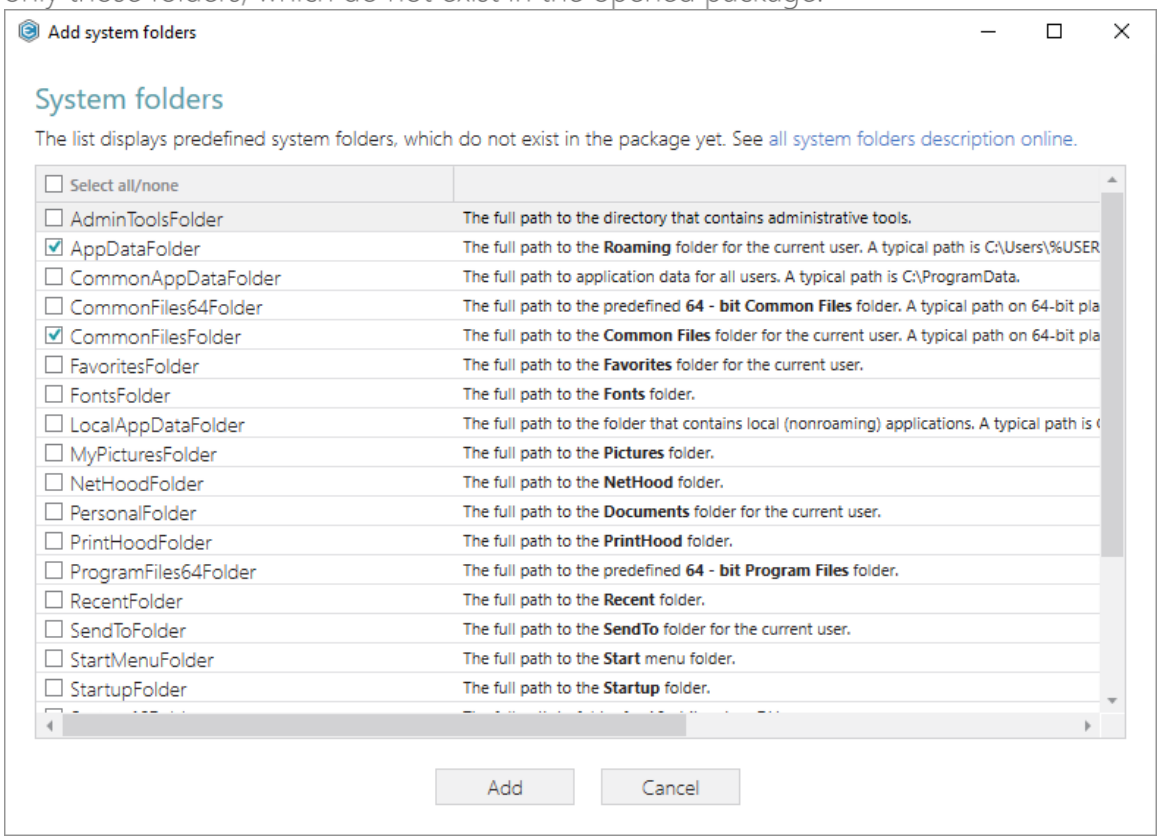
- Ensure that the necessary system folder does not exist in the package. To display all package folders, clear the Hide empty folder and Hide items without permissions filters.



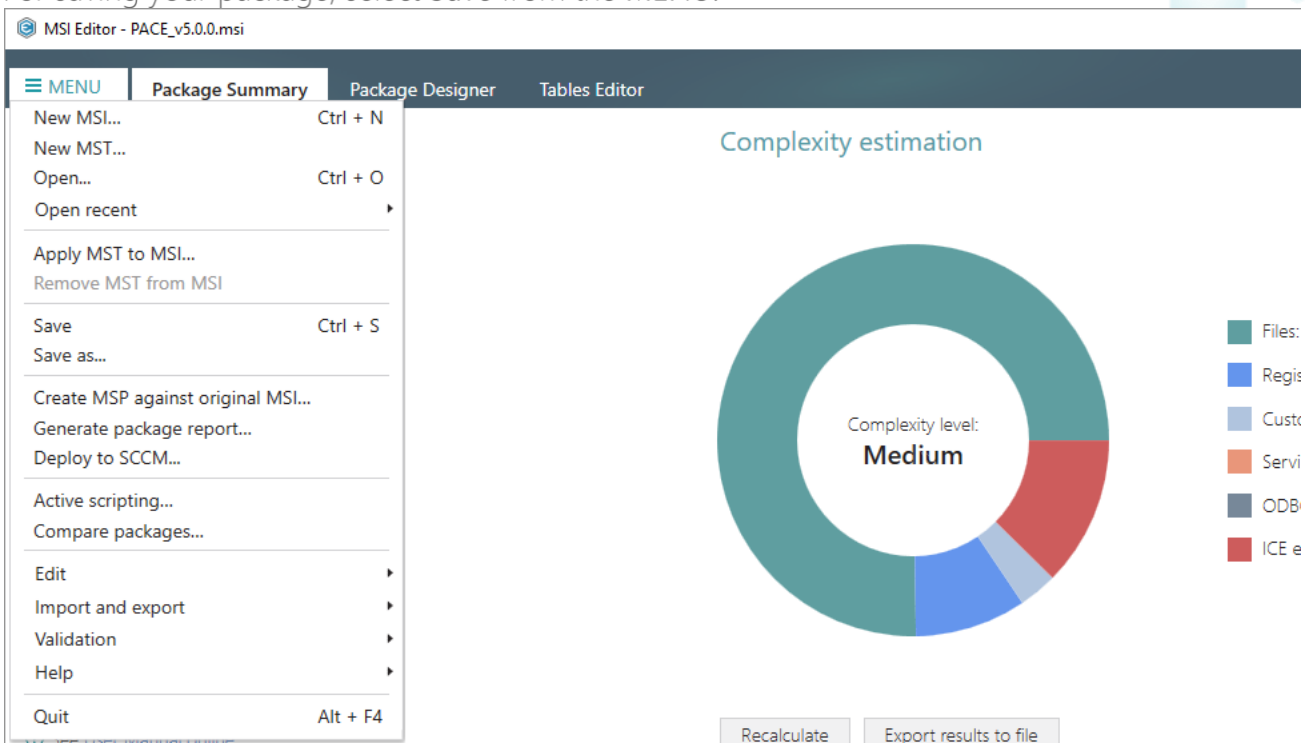
- To add system folders, select **Add** -> **System folders** from the toolbox.



- Choose system folders, which you want to add, and then click **Add**. Note that this list displays only those folders, which do not exist in the opened package.



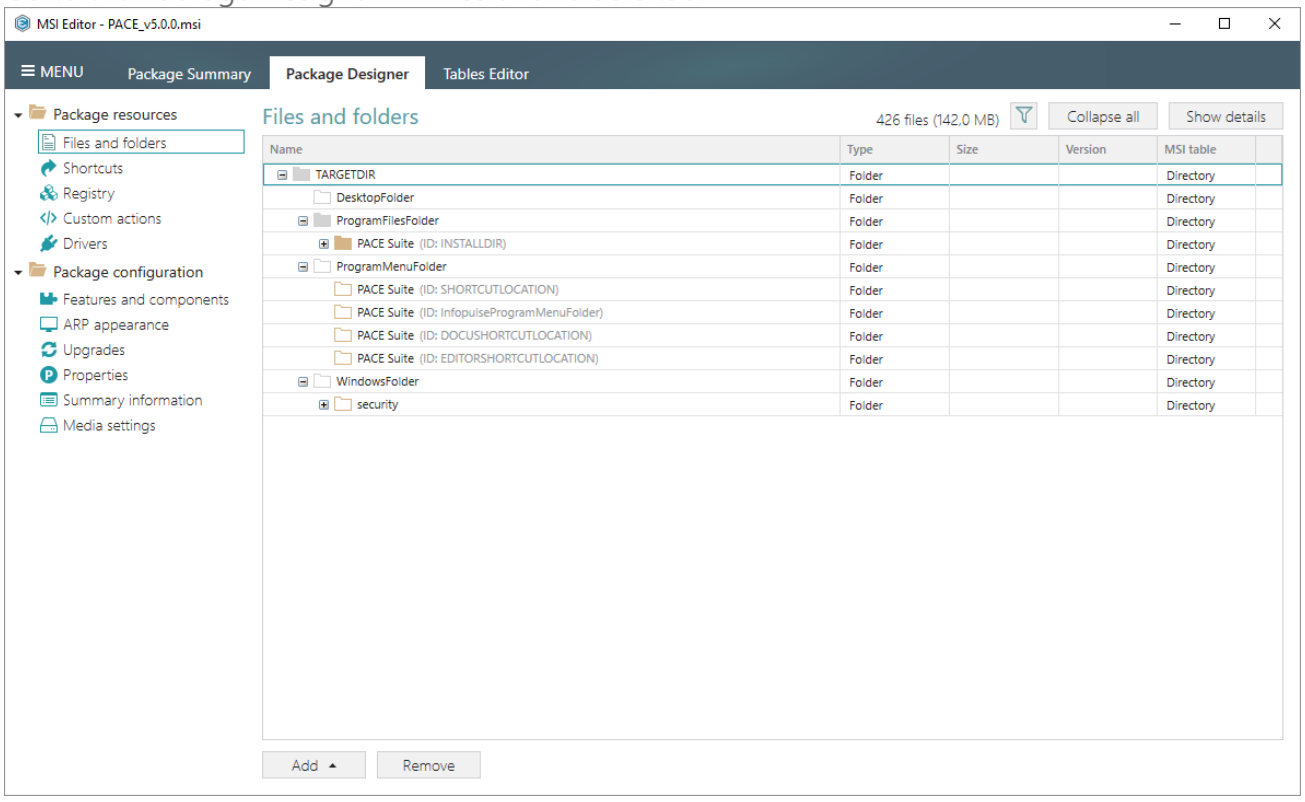
- For saving your package, select Save from the MENU.



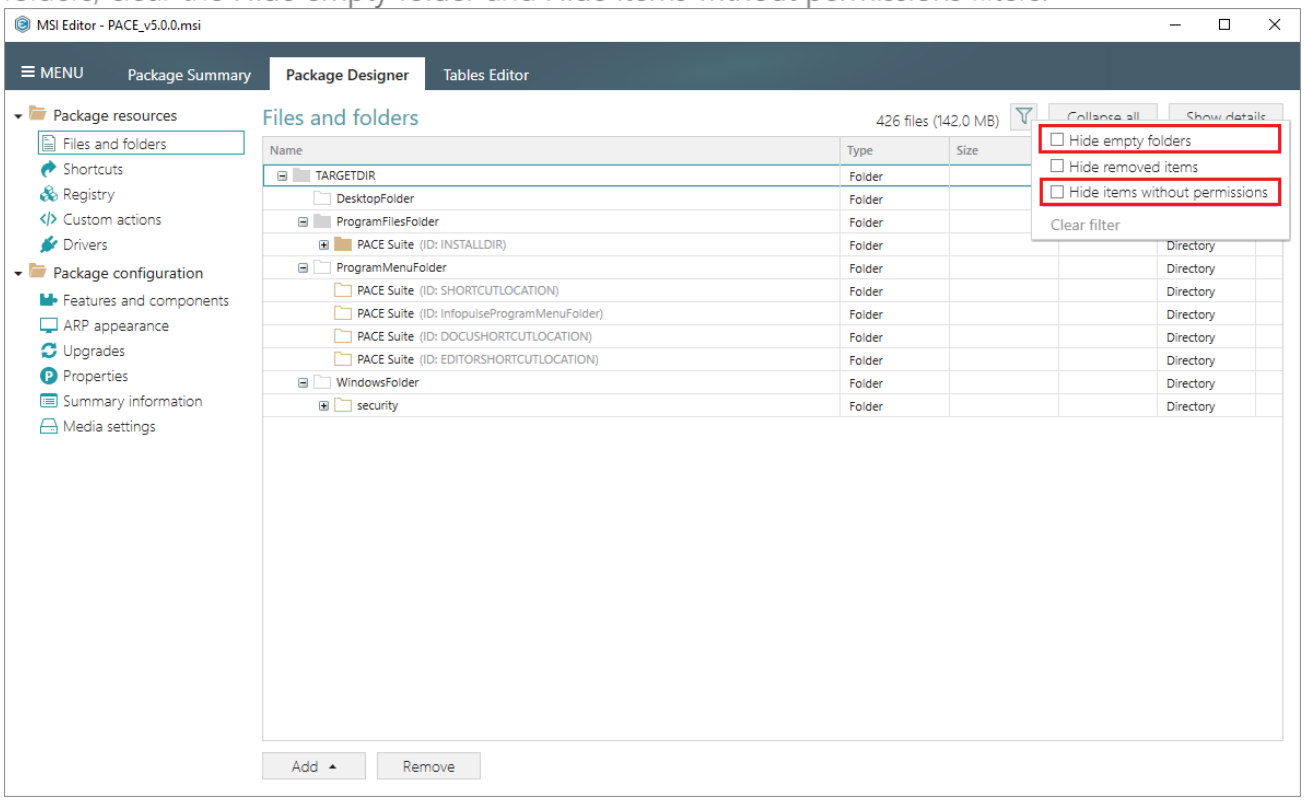
3.7.4.2 Add Subfolder

Add a subfolder (user-defined folder) to the package.

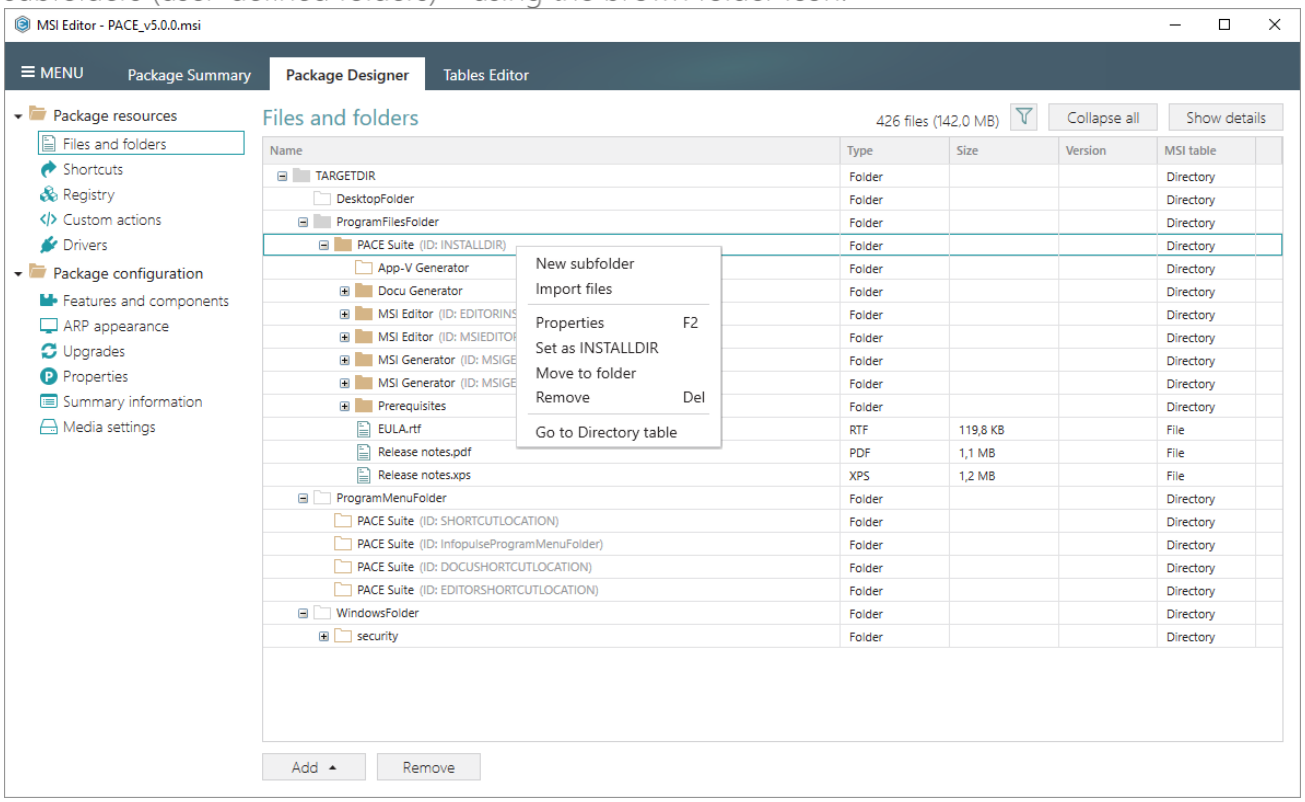
- Go to the Package Designer -> Files and folders tab.



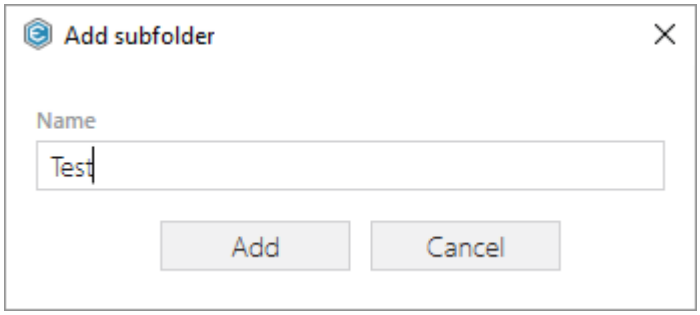
- Ensure that the necessary subfolder does not exist in the package. To display all package folders, clear the **Hide empty folder** and **Hide items without permissions** filters.



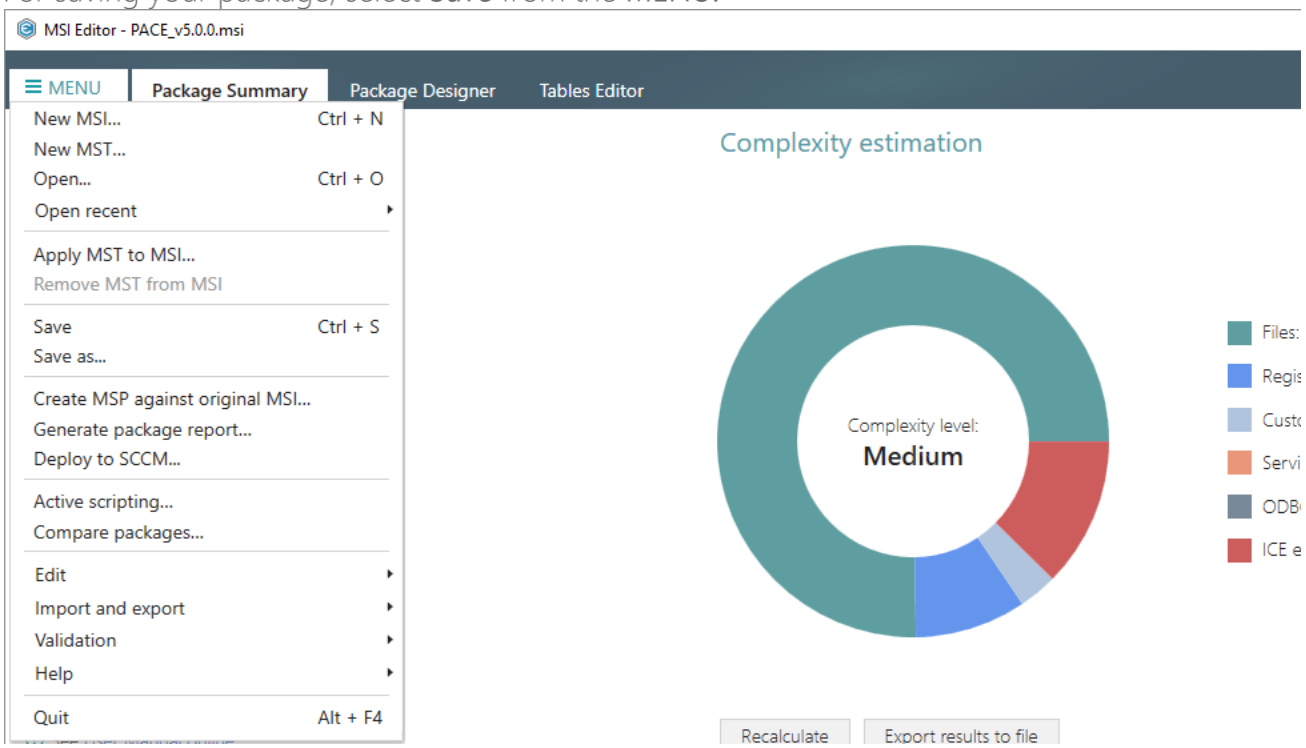
- Select **New subfolder** from the context menu of a folder, where you want to create a new subfolder. Note that system folders are displayed using the grey folder icon and the subfolders (user-defined folders) – using the brown folder icon.



- Enter a folder name and click Add.



- For saving your package, select Save from the MENU.

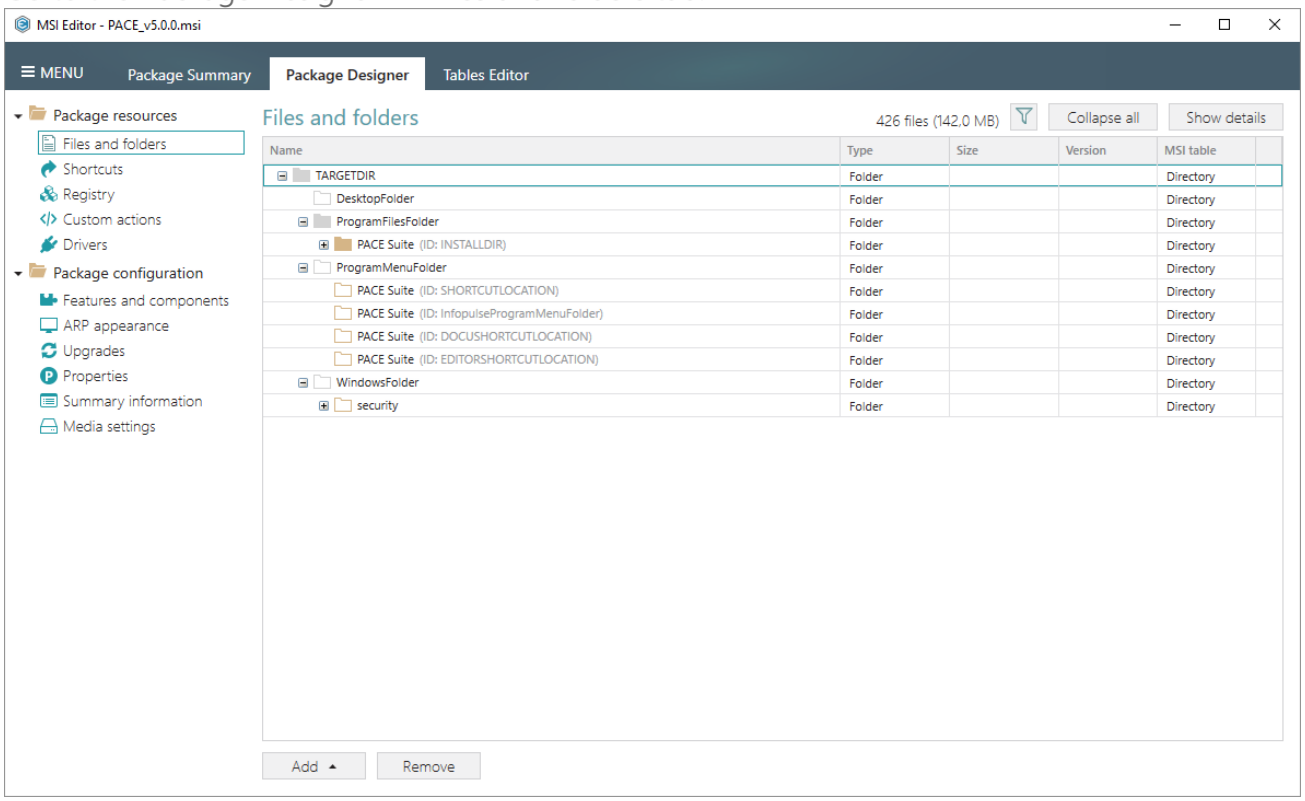


3.7.4.3 Import Files and COM

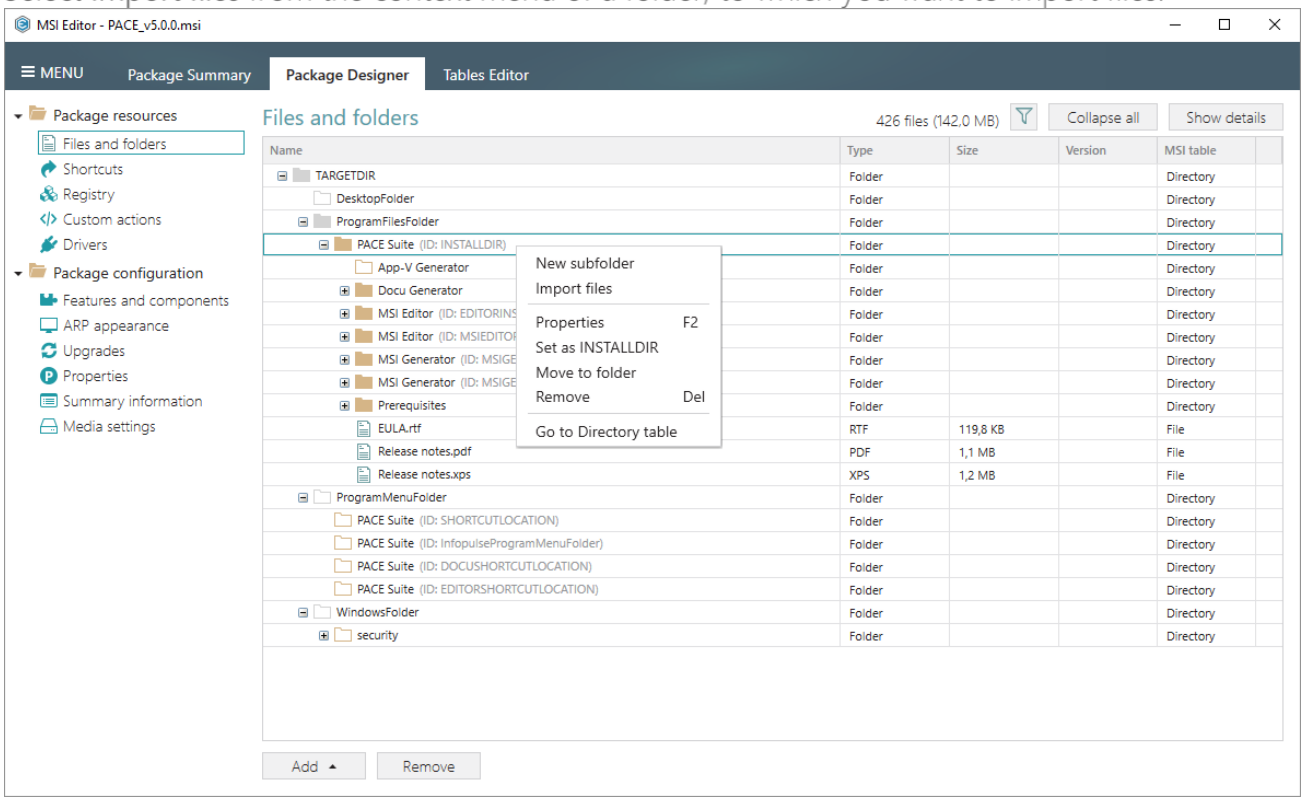
Import files along with their, extracted on-the-fly, COM information.

NOTE MSI Editor extracts both 32-bit and 64-bit COM information and .NET assemblies from the selected (*.DLL, *.OCX, *.LTB) files.

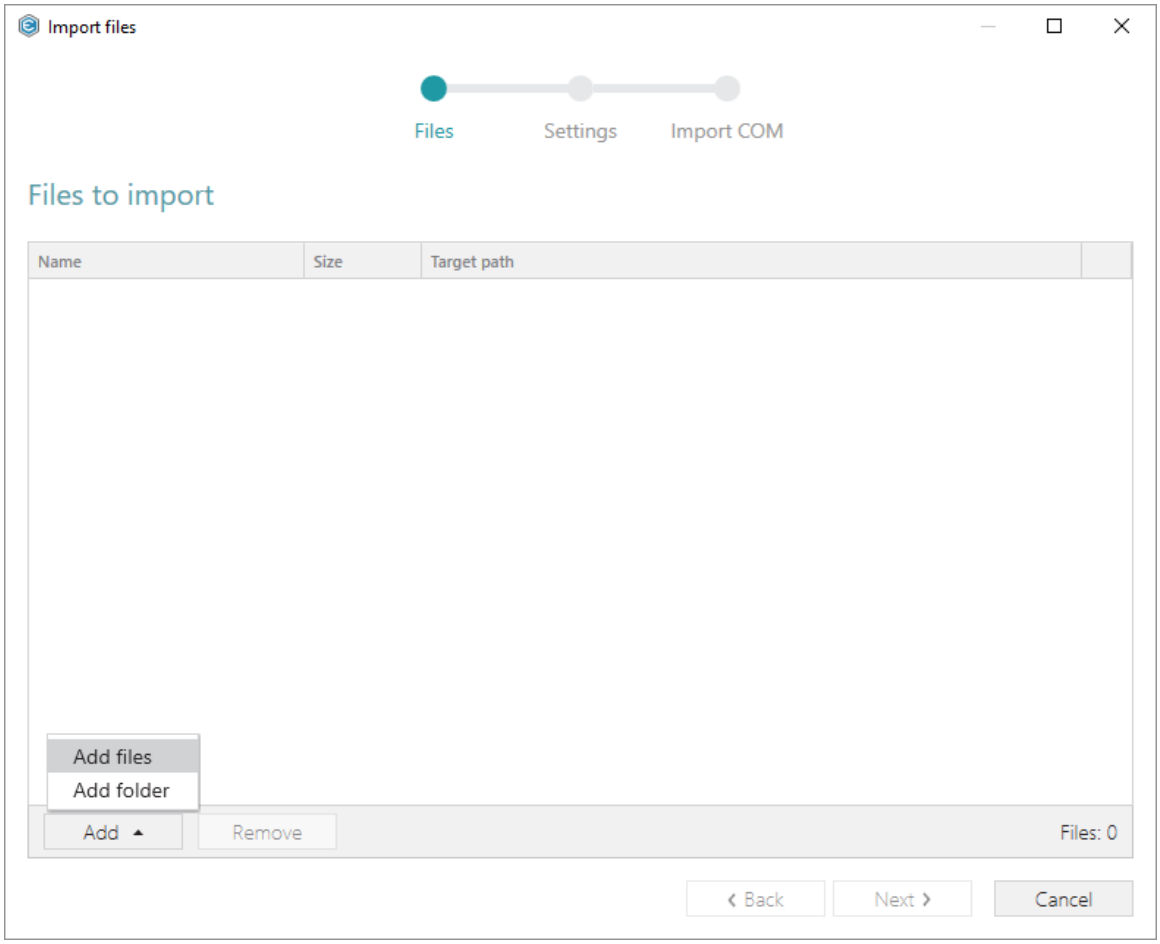
- Go to the Package Designer -> Files and folders tab.



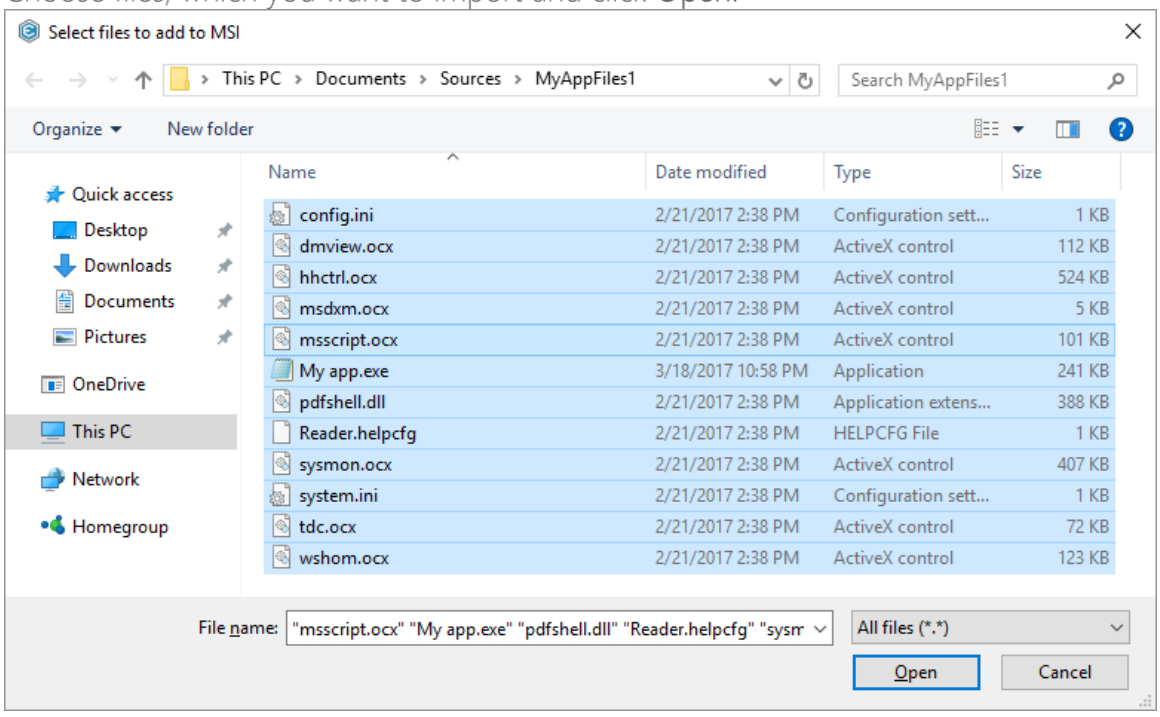
- Select Import files from the context menu of a folder, to which you want to import files.



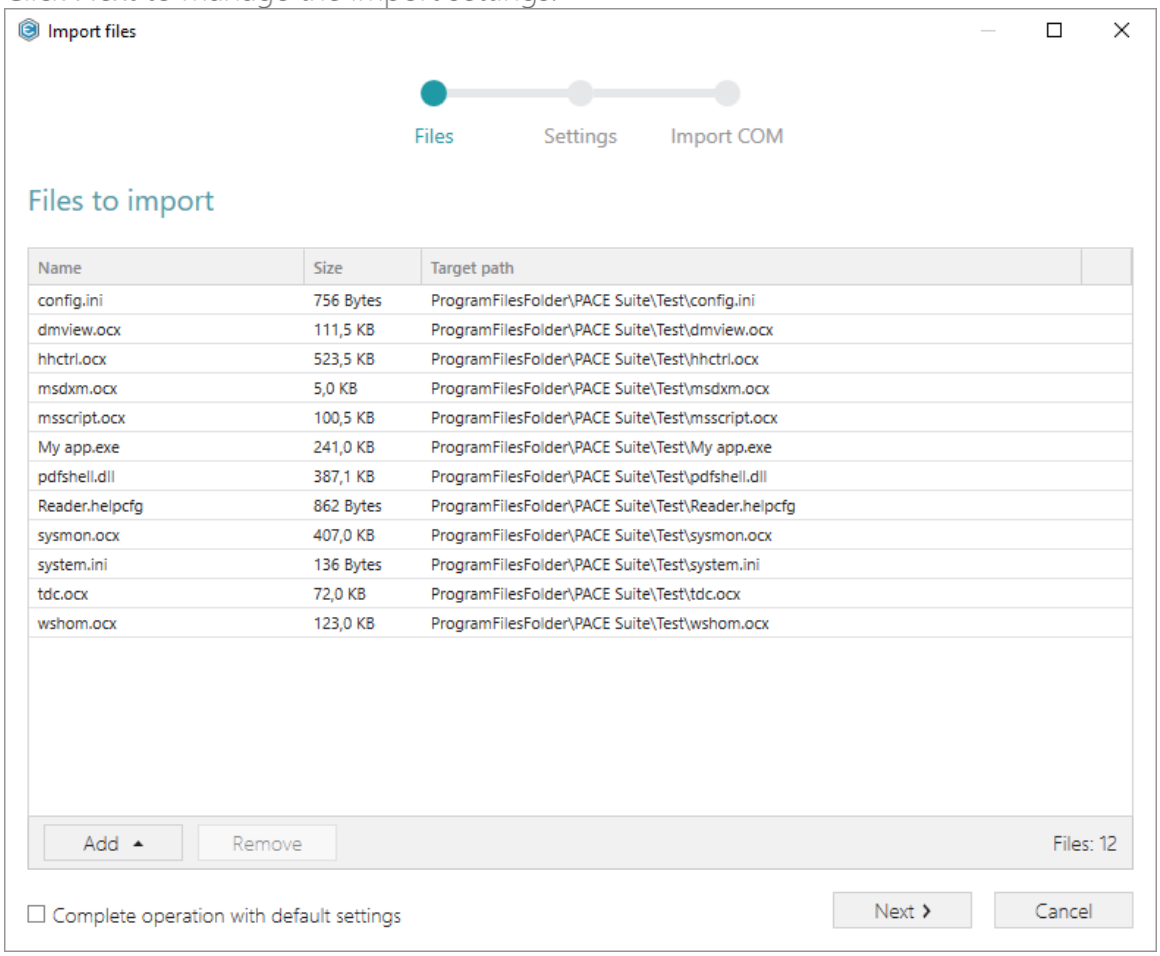
- Select Add -> Add files to choose files in a folder or Add -> Add folder to choose a folder with subfolders and files.



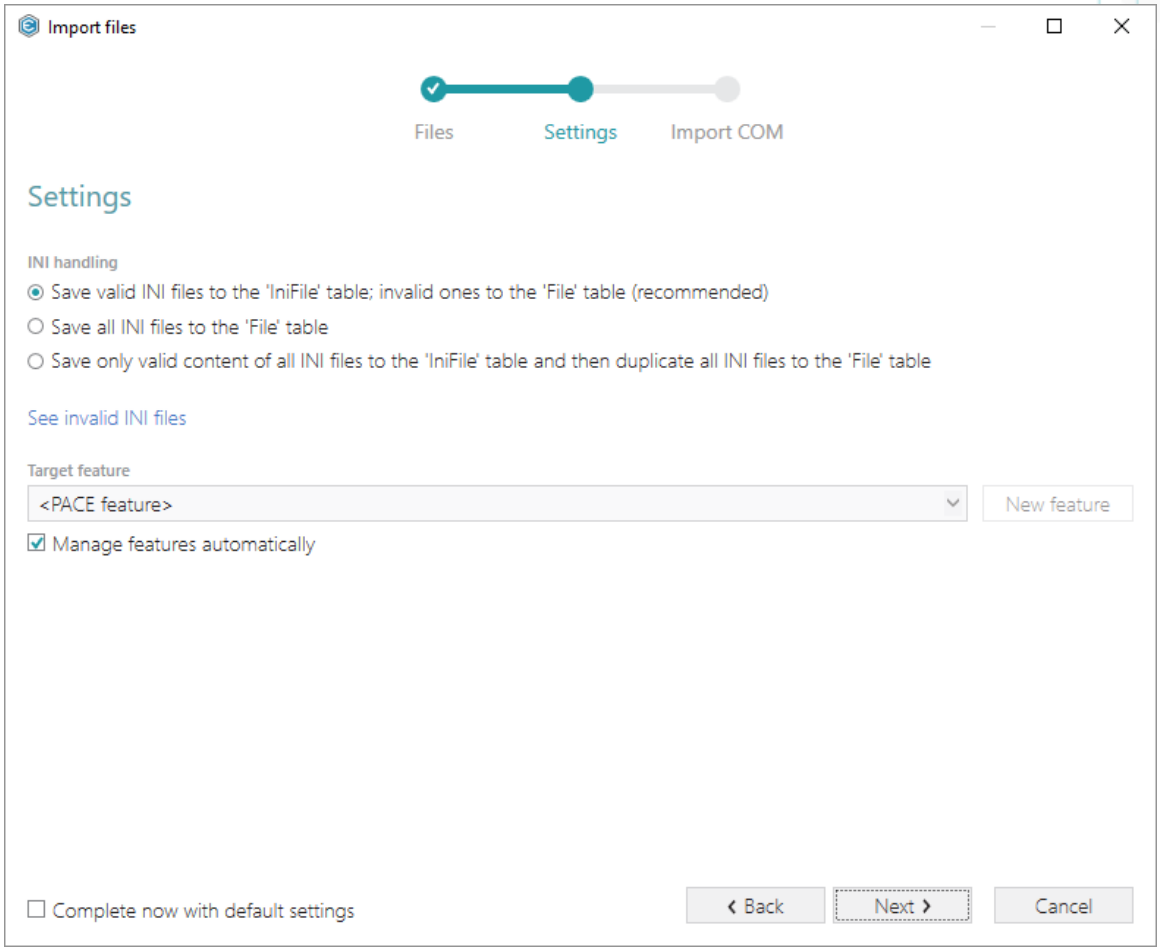
- Choose files, which you want to import and click Open.



- Click Next to manage the import settings.



- Select the INI file handling options, target feature settings and click Next. Find the detailed description of these settings in table below.



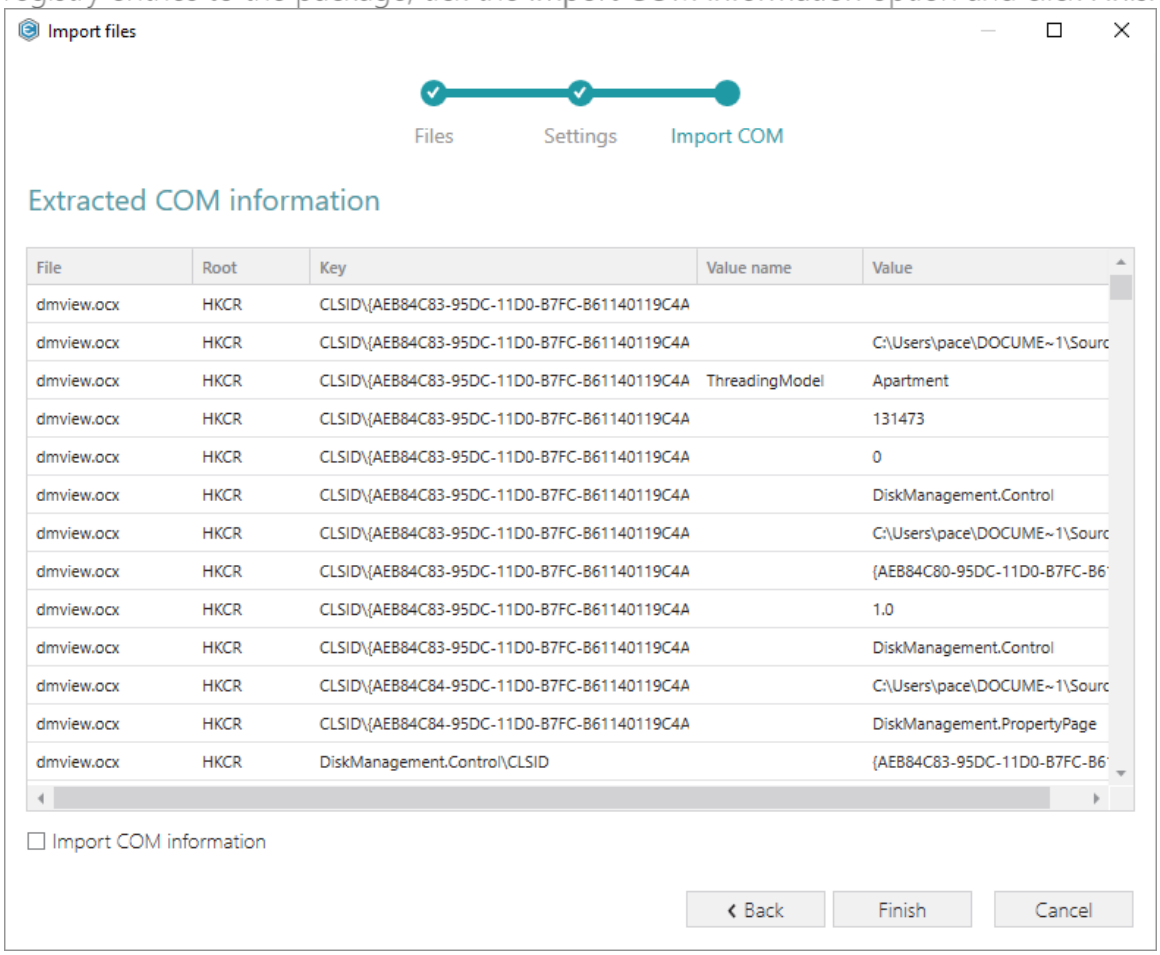
INI handling

Save valid INI files to the 'IniFile' table; invalid ones to the 'File' table (recommended)	INI files, which comply with the INI file format, will be saved to the 'IniFile' table. All other INI files, which contain unsupported data, will be saved to the 'File' table as binary files.
Save all INI files to the 'File' table	All INI files will be saved only to the 'File' table as binary files.
Save only valid content of all INI files to the 'IniFile' table and then duplicate all INI files to the 'File' table	A valid content of all INI files will be saved to the 'IniFile' table, and then all INI files will be saved to the 'File' table as binary files. This allows installing INI files to the target system keeping their original structure with unsupported data, and update the necessary INI file values.

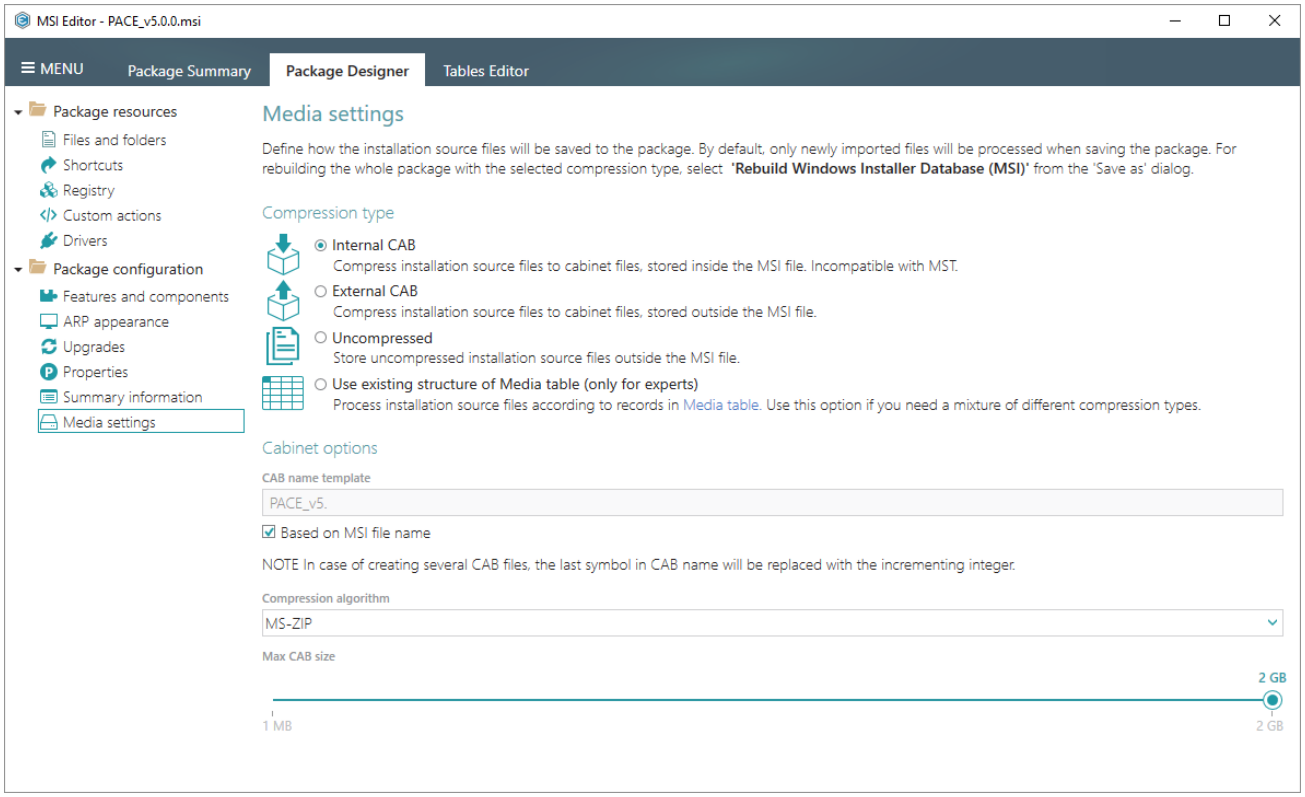
Target feature

Manage features automatically	The specific PACE features will be created (or used if they exist) for the components, containing the imported files. Feature structure is based on the packaging best practices and depends on the destination location of files.
<a feature, selected from the list>	Components with files will be assigned to a feature, selected from the list.

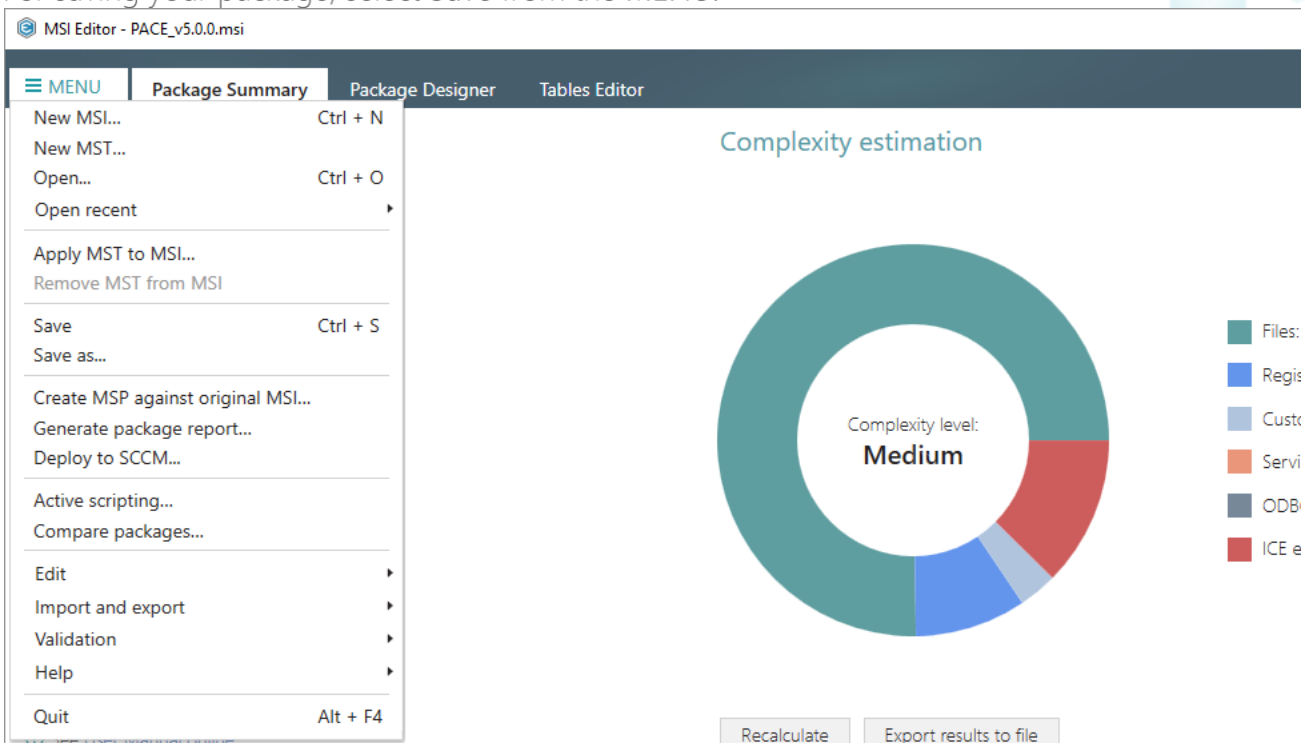
- If the selected files contain COM information, it will be displayed at this step. To include these registry entries to the package, tick the Import COM information option and click Finish.



- Go to the Package Designer -> Media settings tab and choose the necessary compression type for the imported files. For the compressed files you can manage cabinet options such as cabinet (CAB) name, compression algorithm and maximum CAB size. Cabinets will be spit by the selected maximum size.



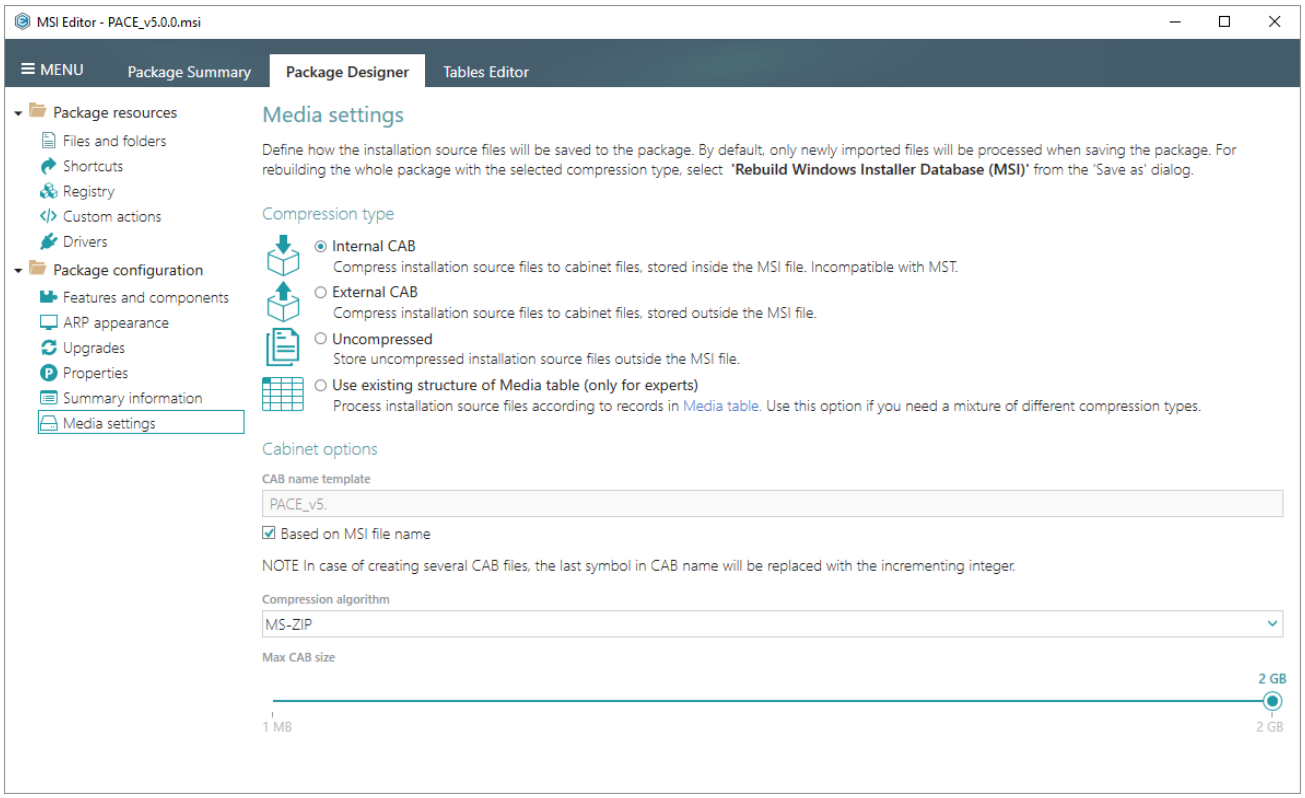
- For saving your package, select Save from the MENU.



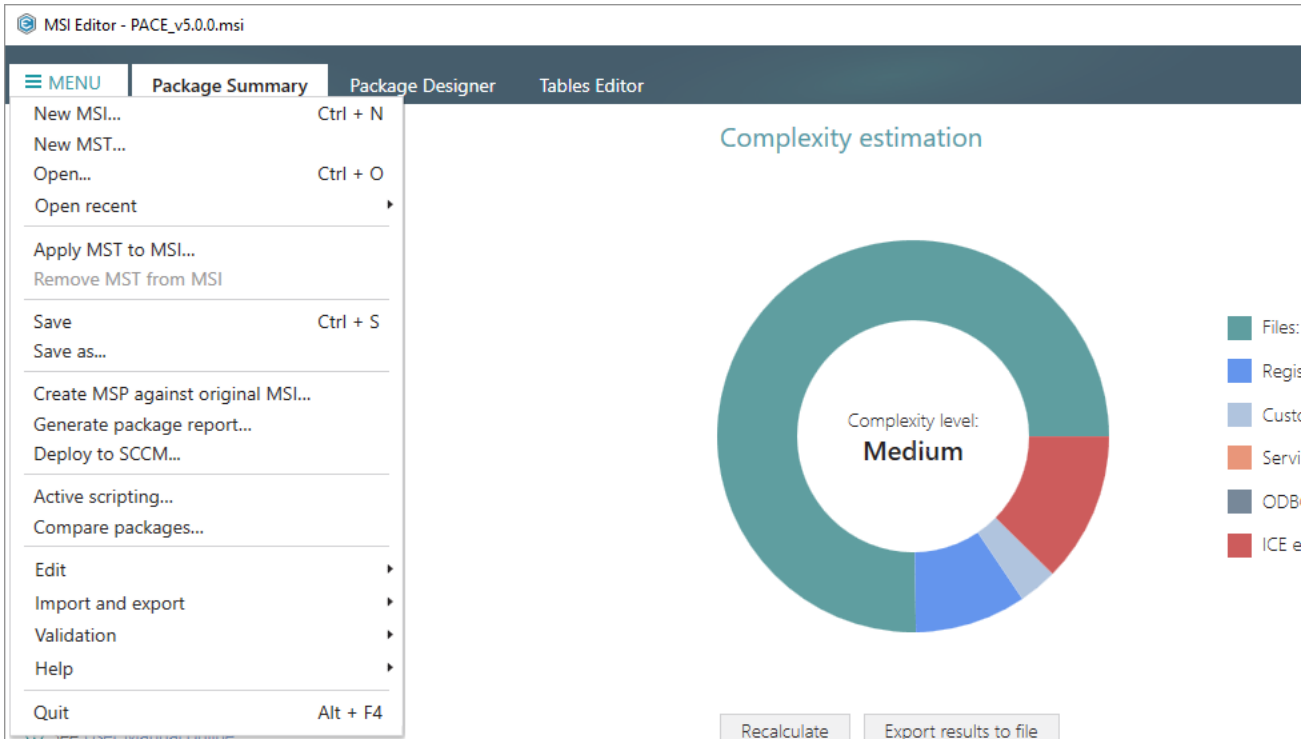
3.7.4.4 Change Compression Type

Change the compression type for all package files by rebuilding the whole package. The rebuild package will be saved as a new separate MSI file.

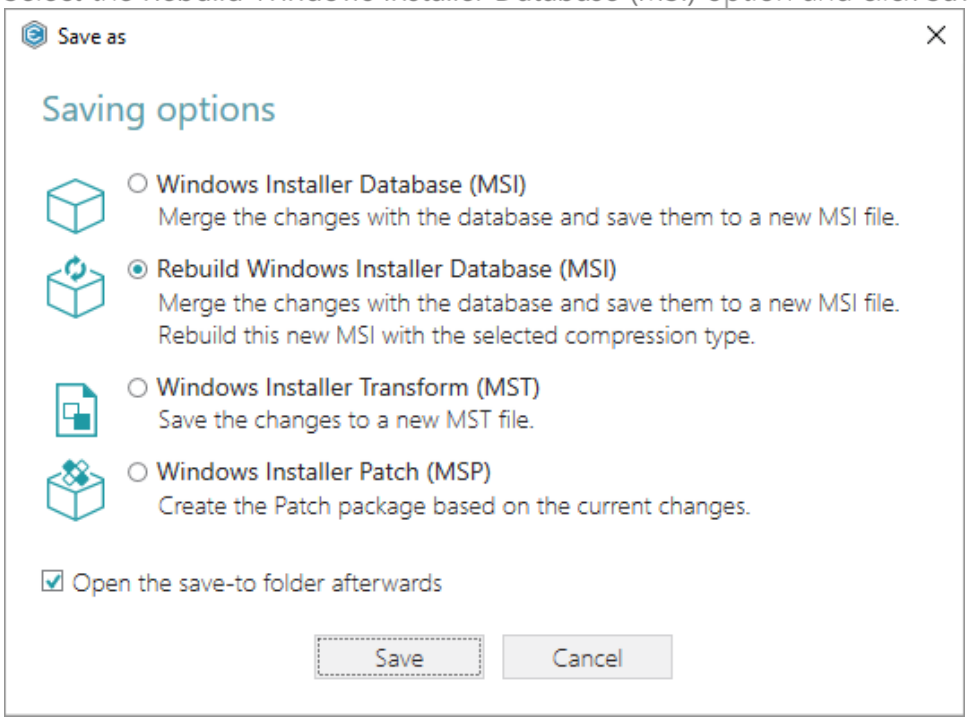
- Go to the Package Designer -> Media settings tab and select a new compression type for all package files. Also, for compressed files you manage cabinet options such as cabinet (CAB) name, compression algorithm and maximum CAB size. Cabinets will be spit by the selected maximum size.



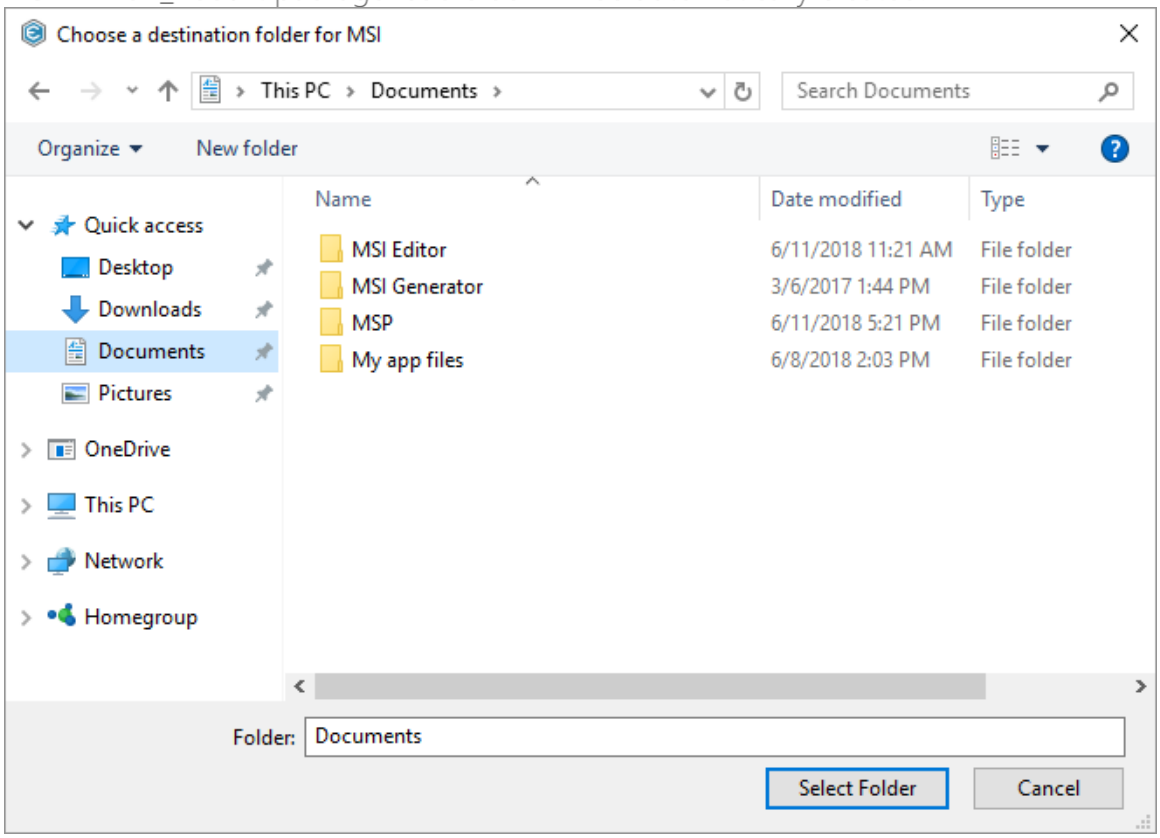
- Select Save as from MENU.



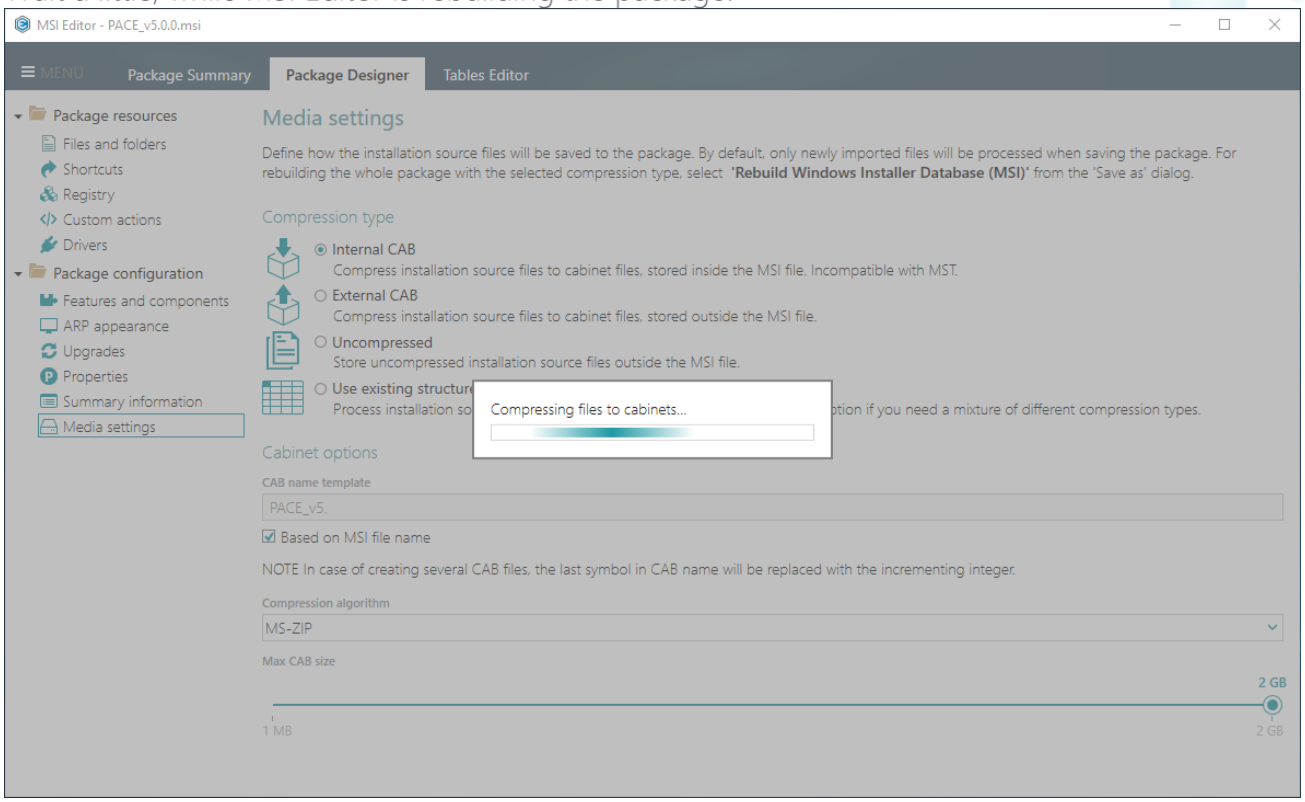
- Select the Rebuild Windows Installer Database (MSI) option and click Save.



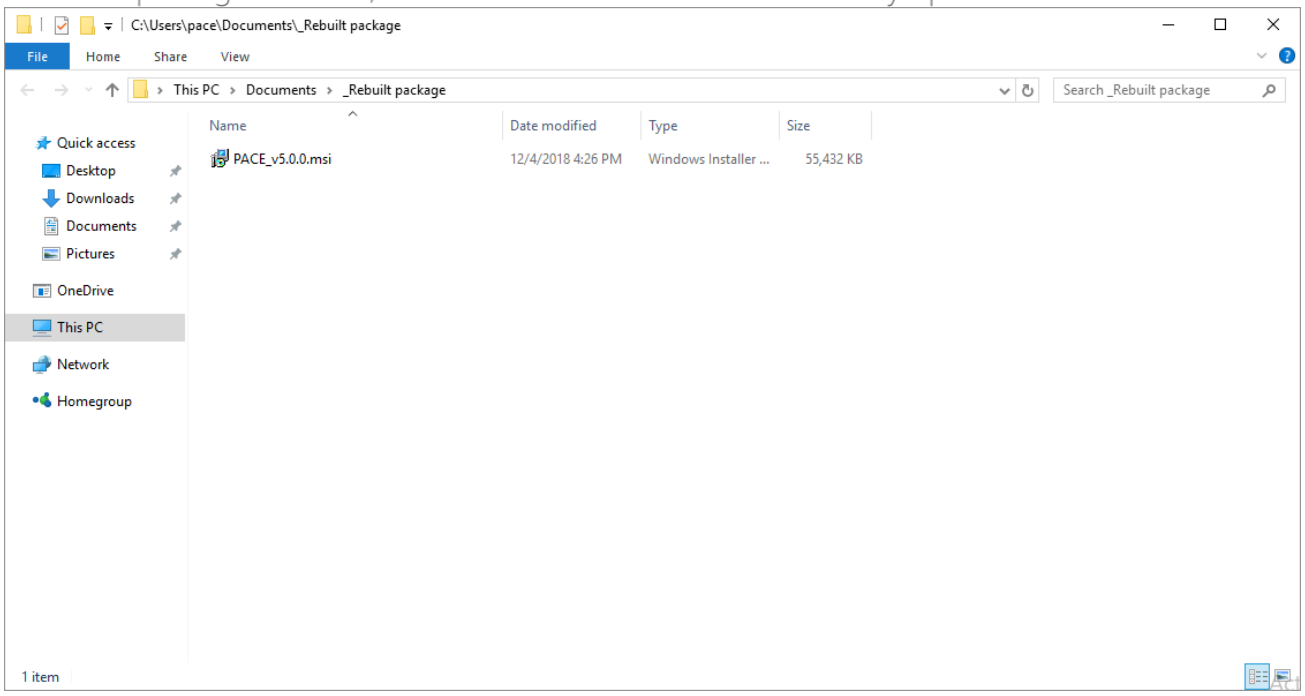
- Choose a folder, to which the rebuilt MSI will be saved, and click Select Folder.
NOTE The "_Rebuilt package" subfolder will be automatically created.



- Wait a little, while MSI Editor is rebuilding the package.



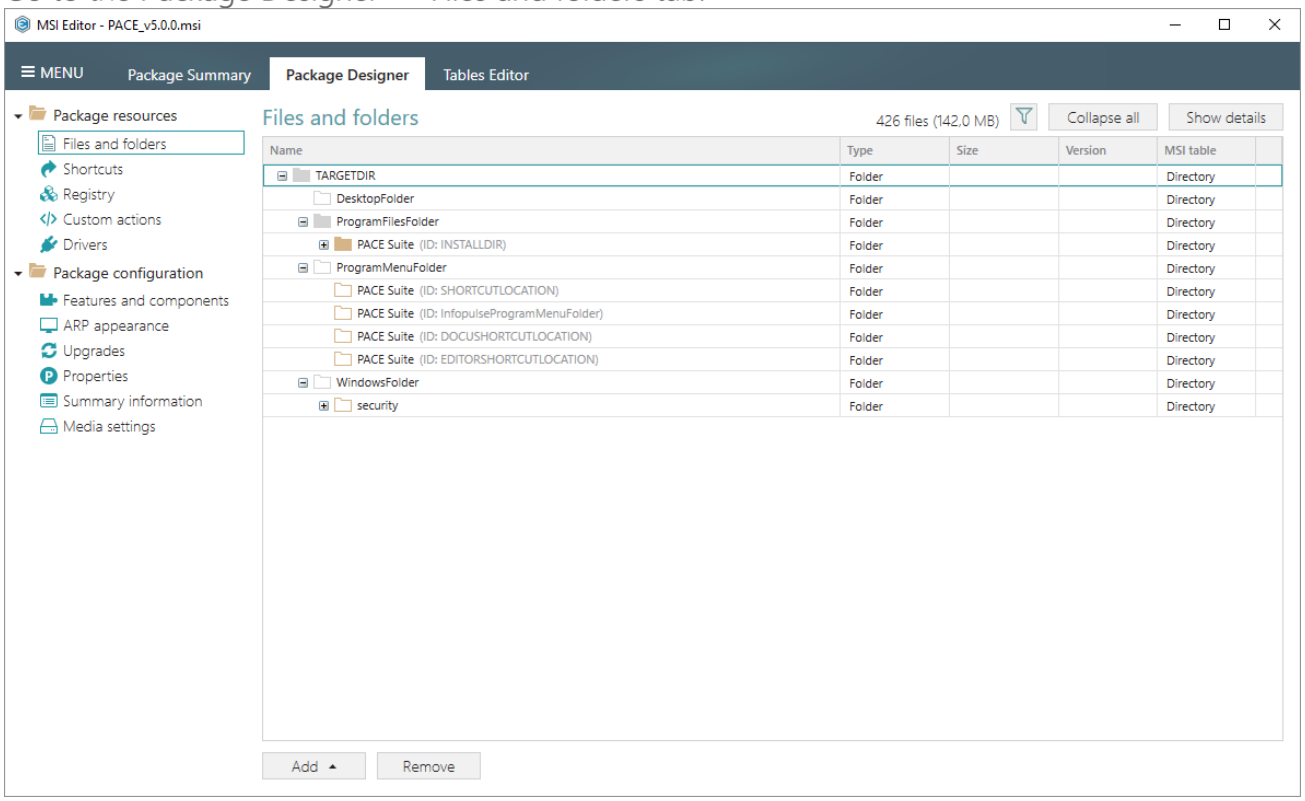
- Once the package is rebuilt, the save-to folder will be automatically opened.



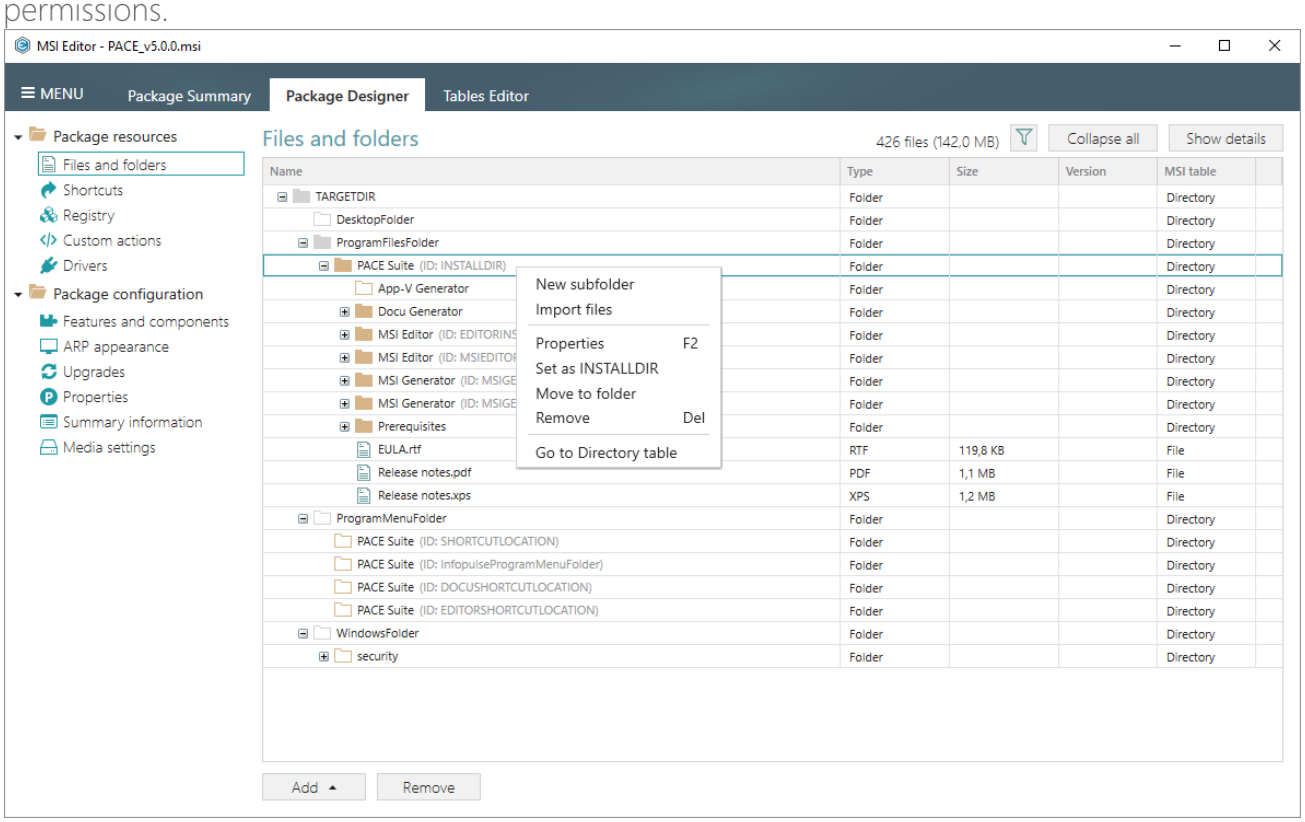
3.7.4.5 Set Permissions

Change permissions for a file or folder.

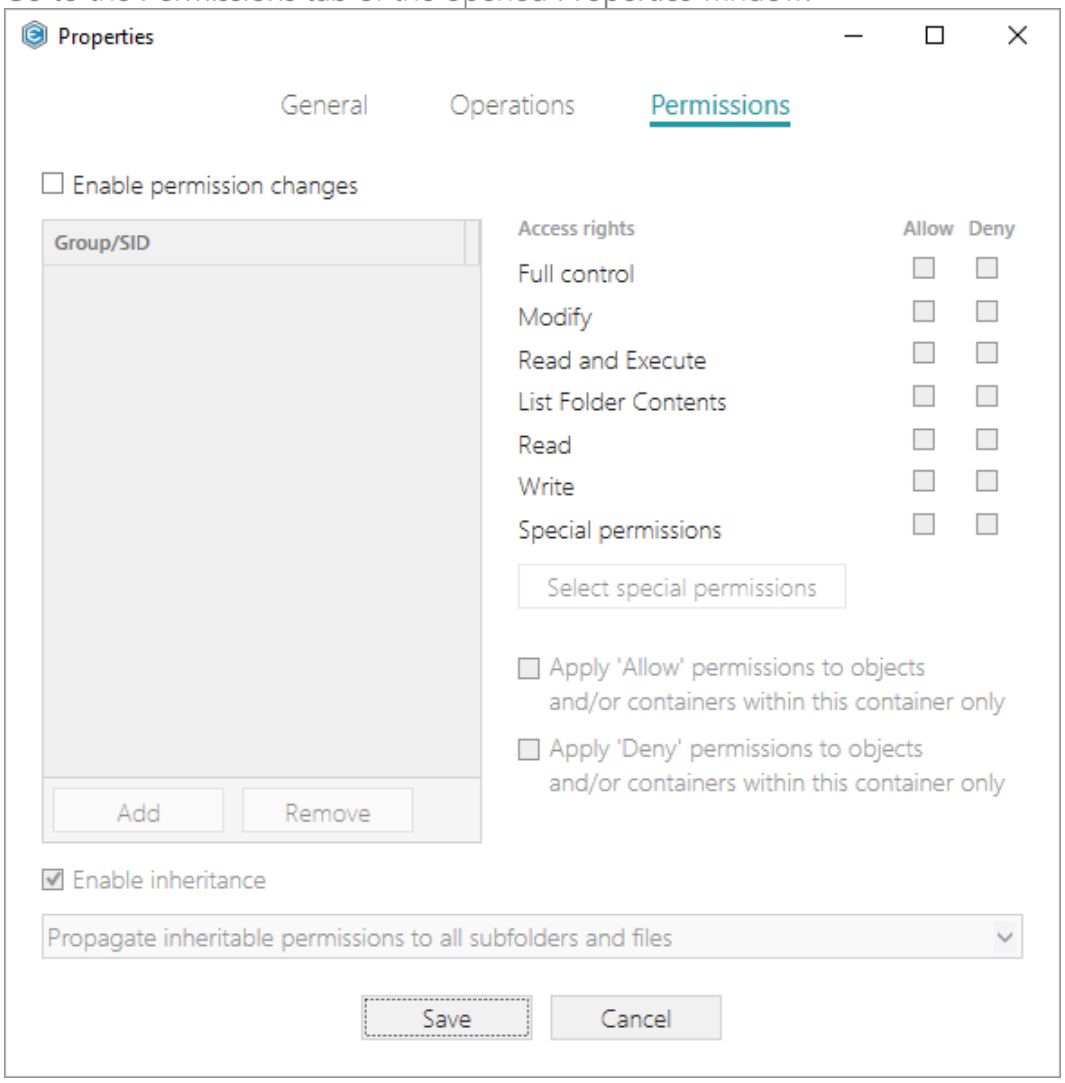
- Go to the Package Designer -> Files and folders tab.



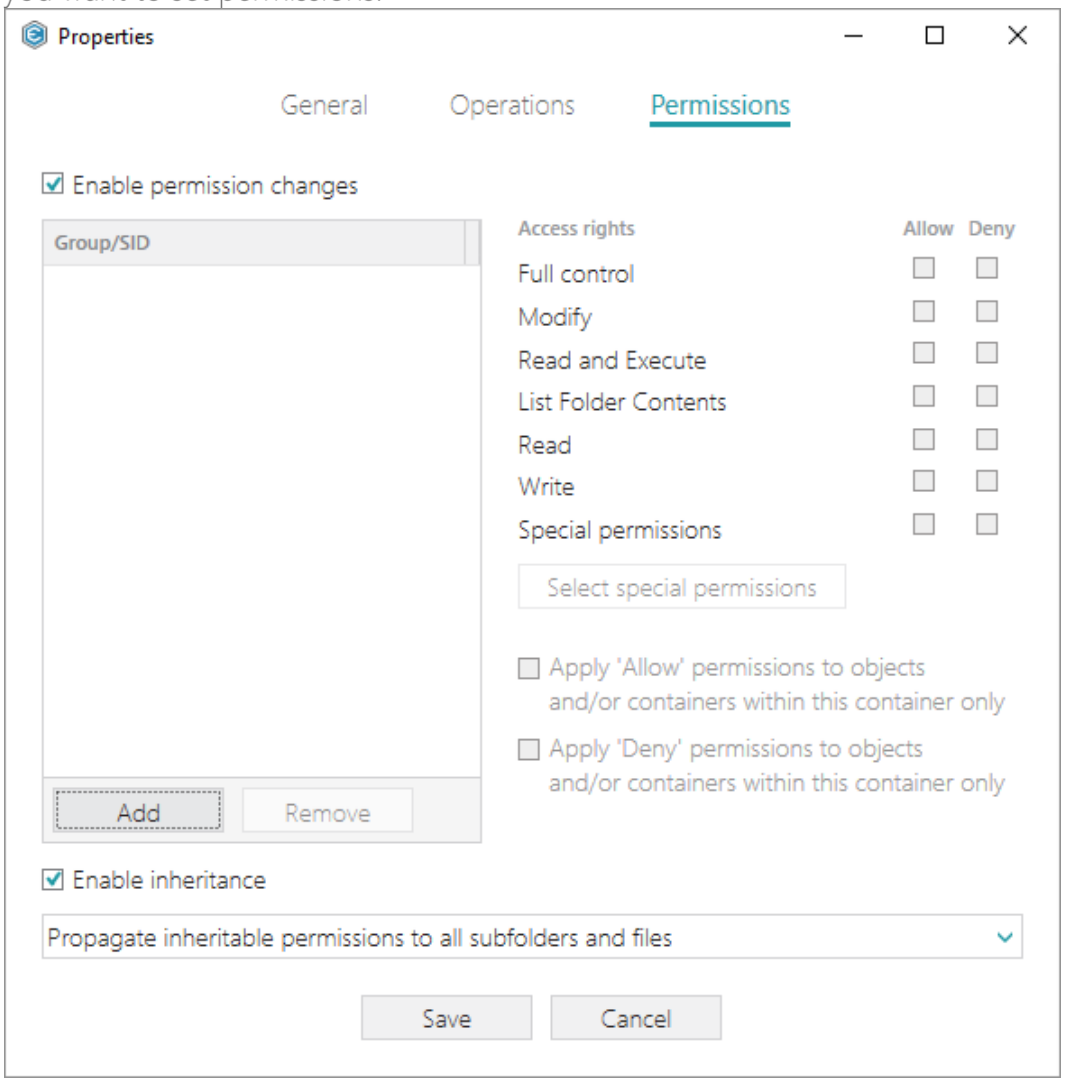
- Select Properties from the context menu of a file or folder, for which you want to set custom permissions.



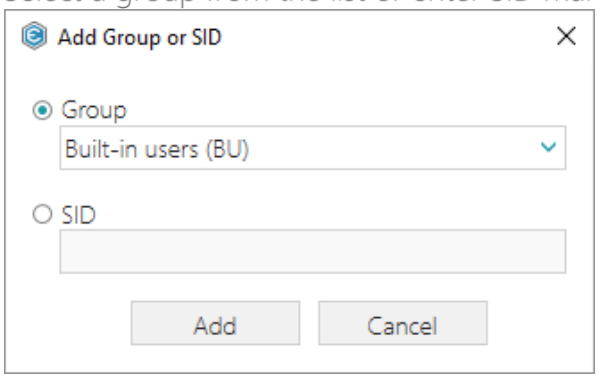
- Go to the Permissions tab of the opened Properties window.



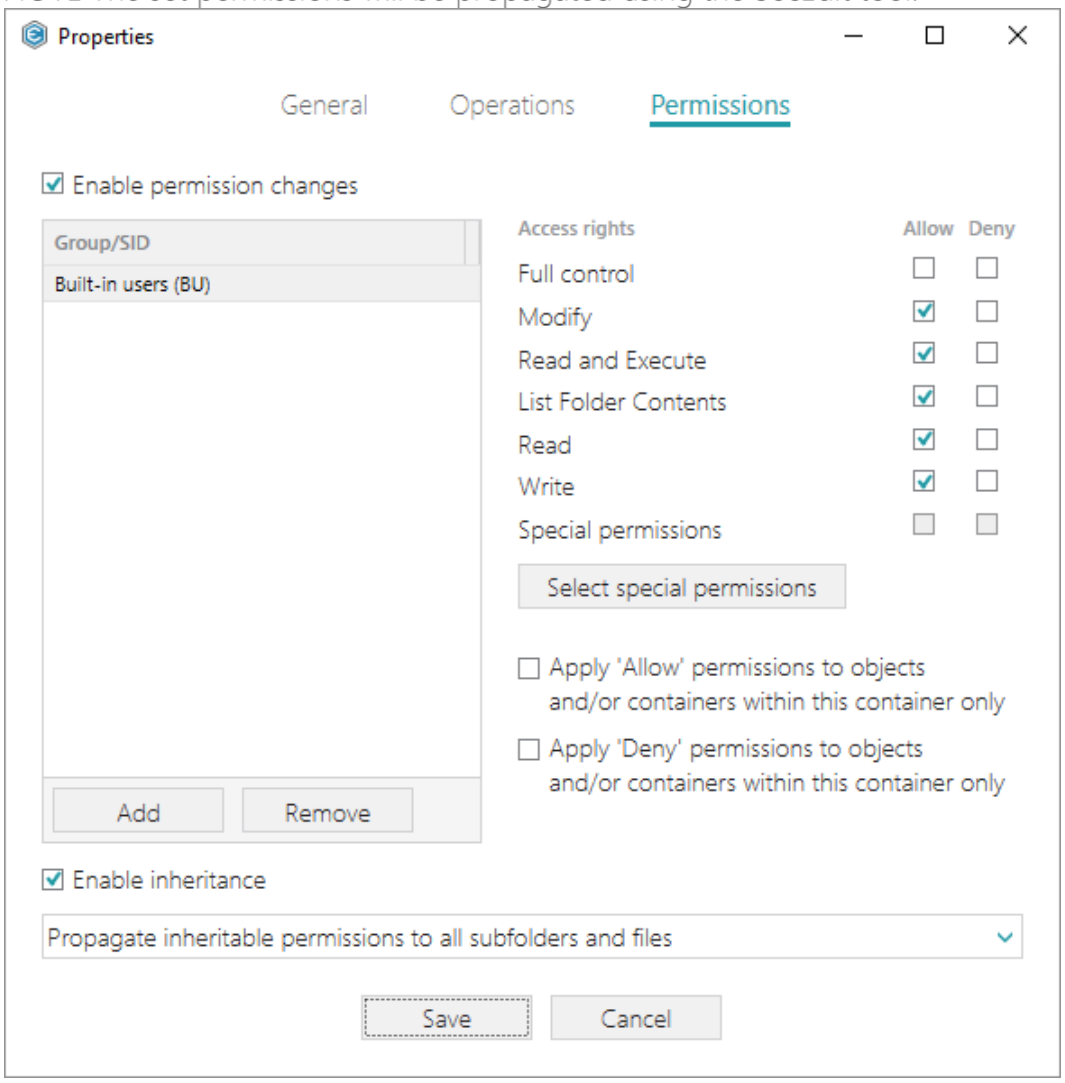
- Tick the **Enable permission changes** checkbox and click **Add** to add a group or SID, for which you want to set permissions.



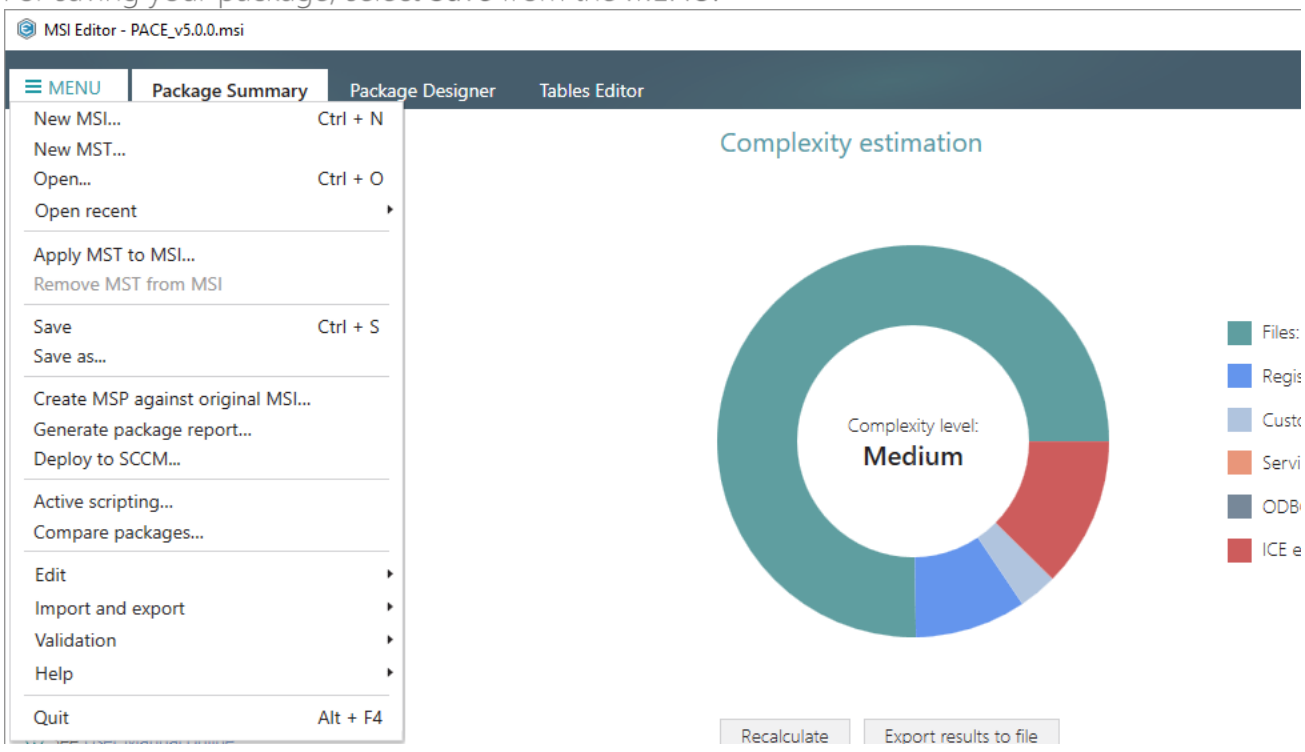
- Select a group from the list or enter SID manually and then click **Add**.



- Select the necessary access rights for the added group or SID and click Save.
NOTE The set permissions will be propagated using the SecEdit tool.



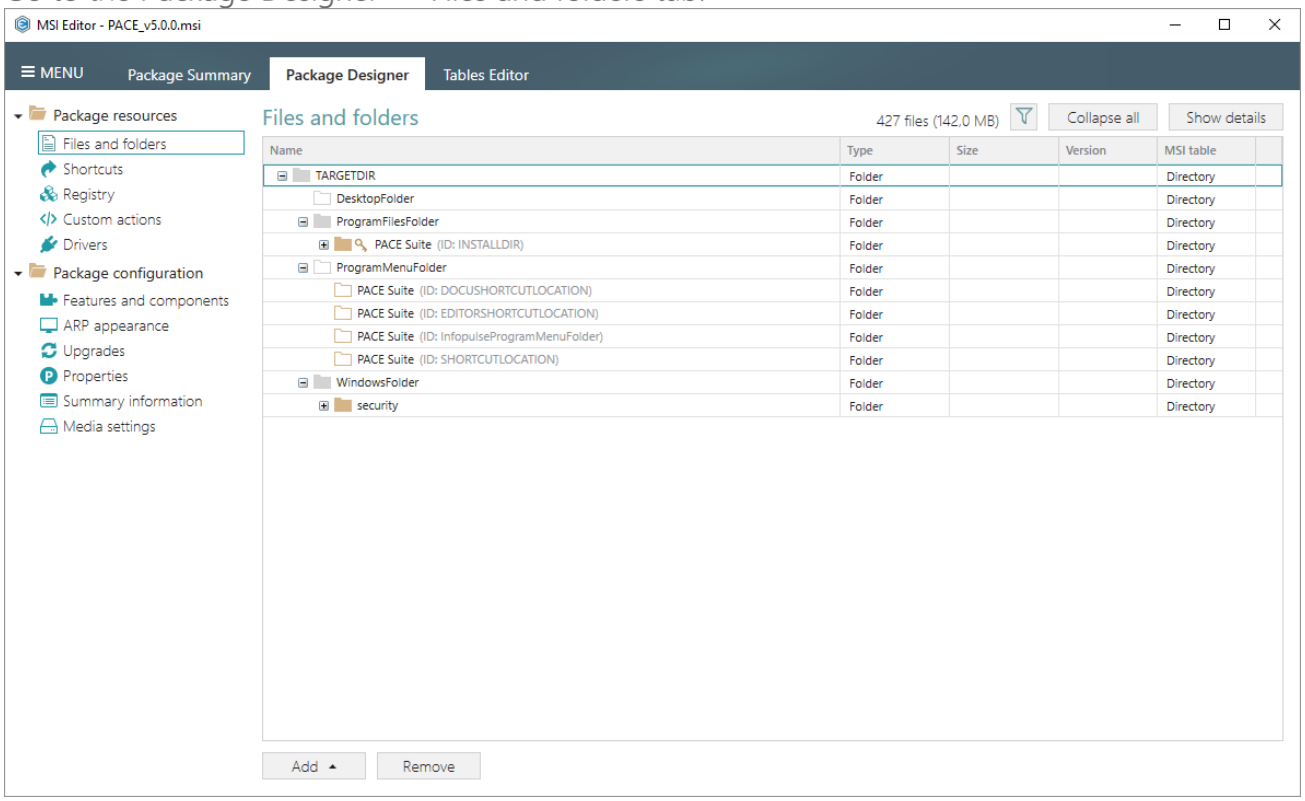
- For saving your package, select Save from the MENU.



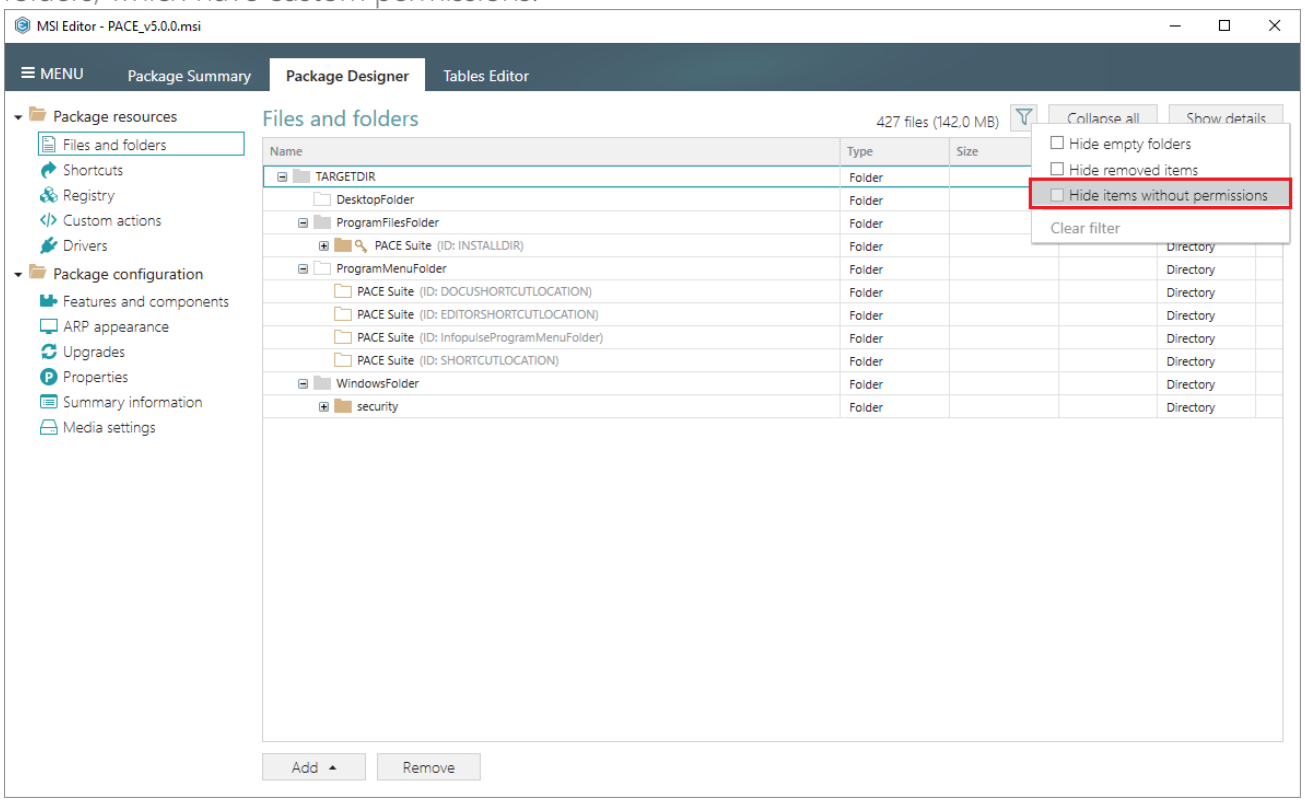
3.7.4.6 Remove Permissions

Remove the captured or manually set custom permissions from a file or folder.

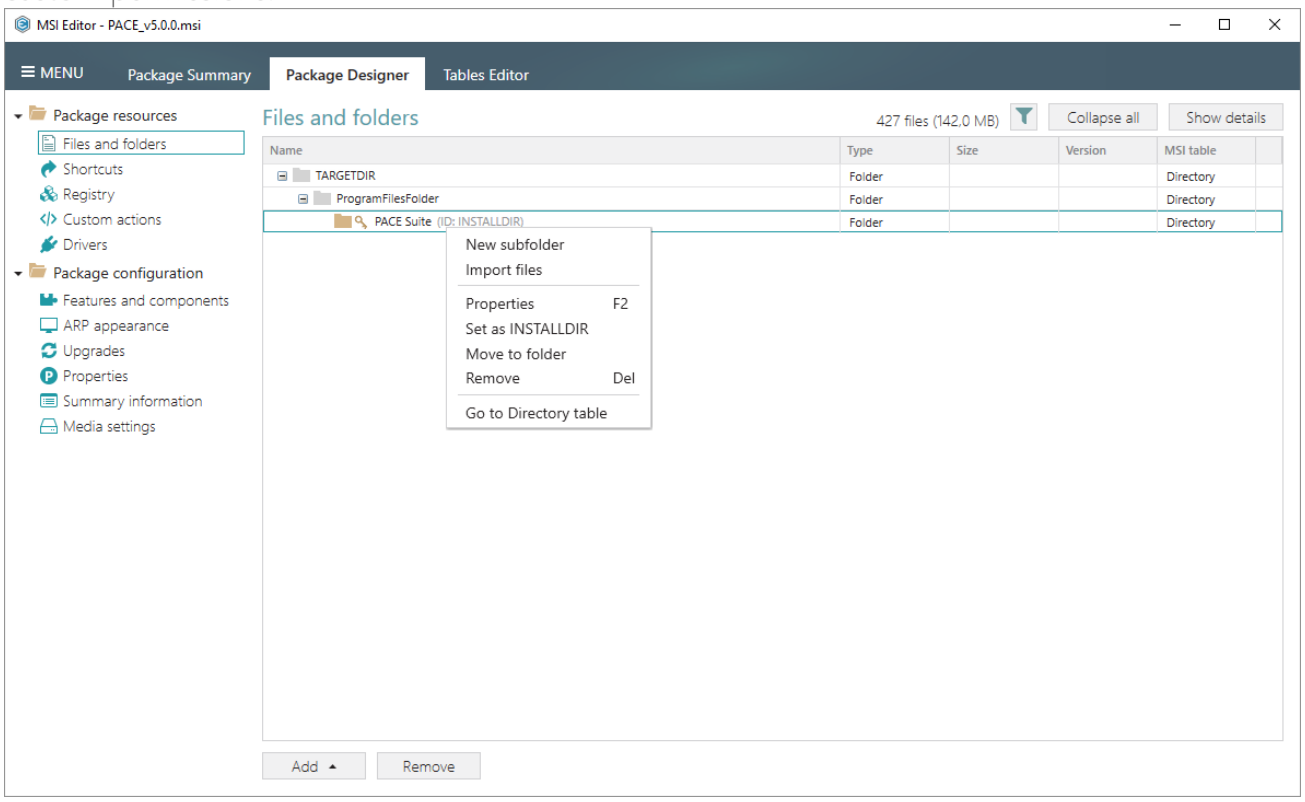
- Go to the Package Designer -> Files and folders tab.



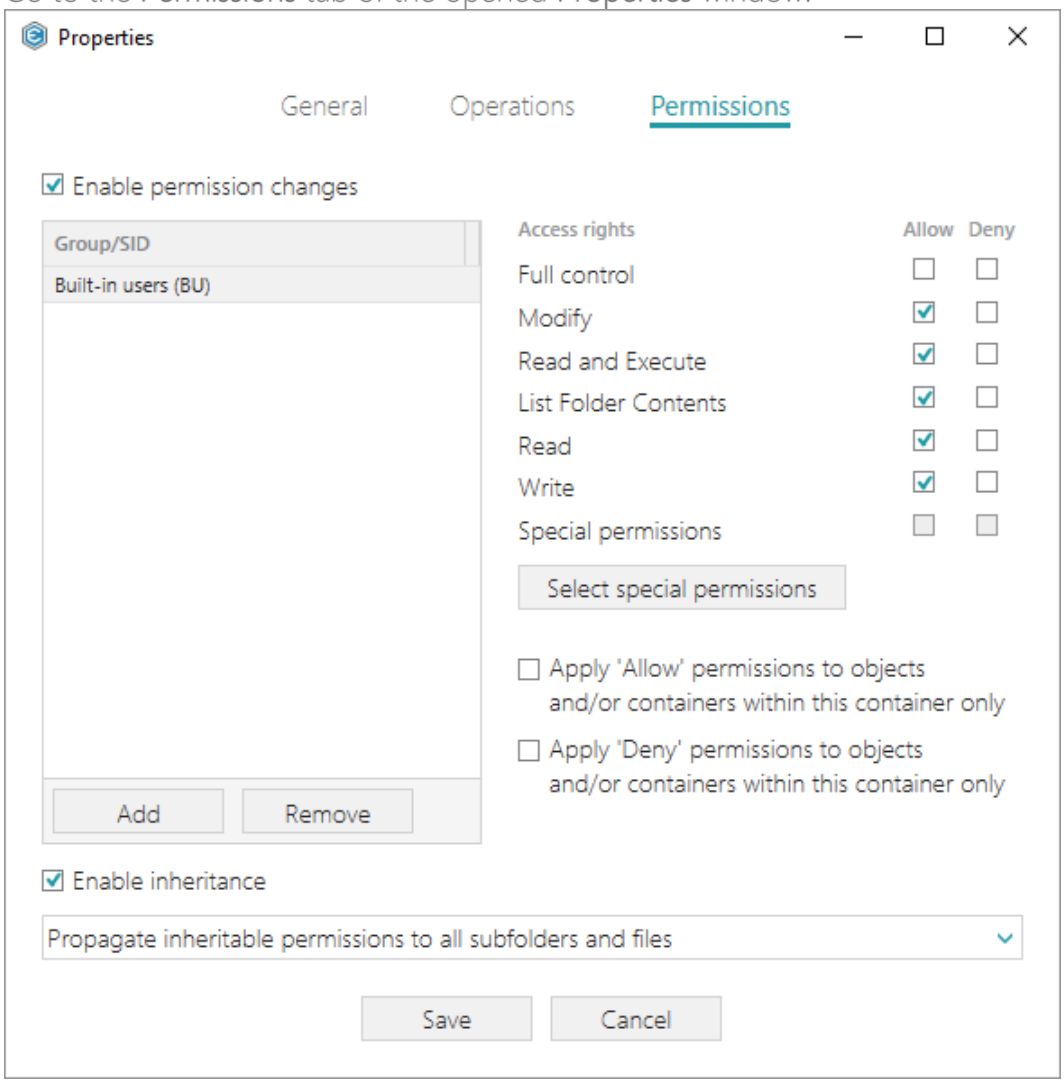
- *(Optional step)* Apply the Hide items without permissions filter to display only those files and folders, which have custom permissions.



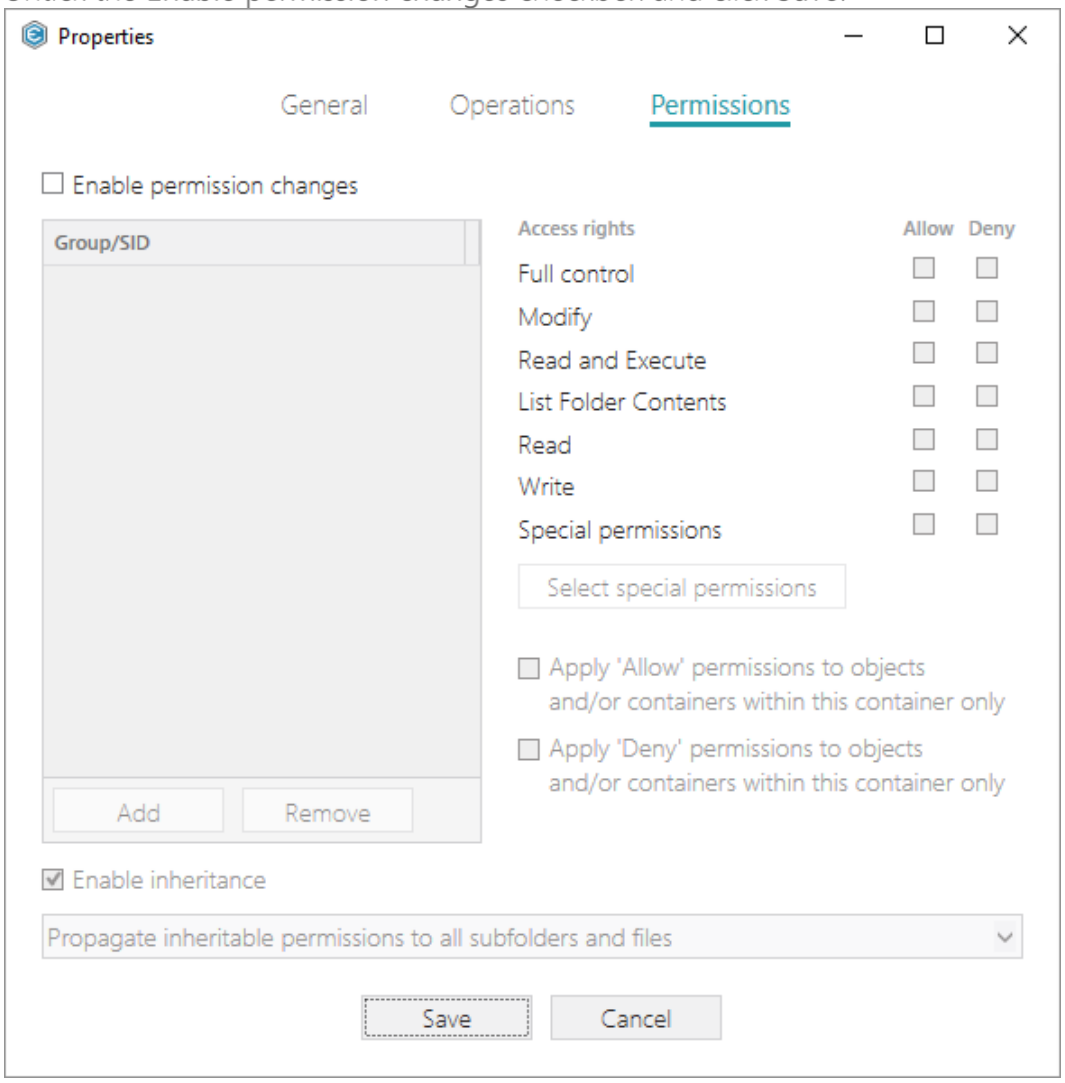
- Select **Properties** from the context menu of a file or folder, for which you want to remove custom permissions.



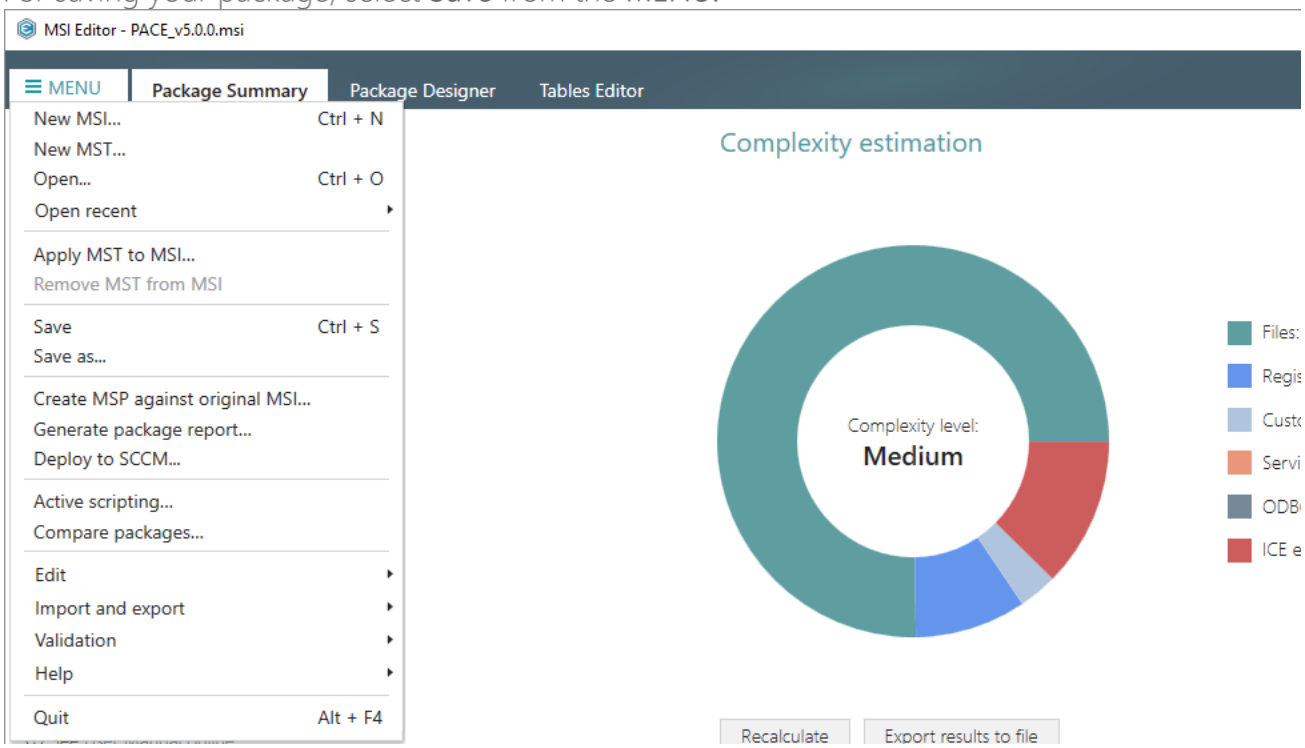
- Go to the **Permissions** tab of the opened **Properties** window.



- Untick the Enable permission changes checkbox and click Save.



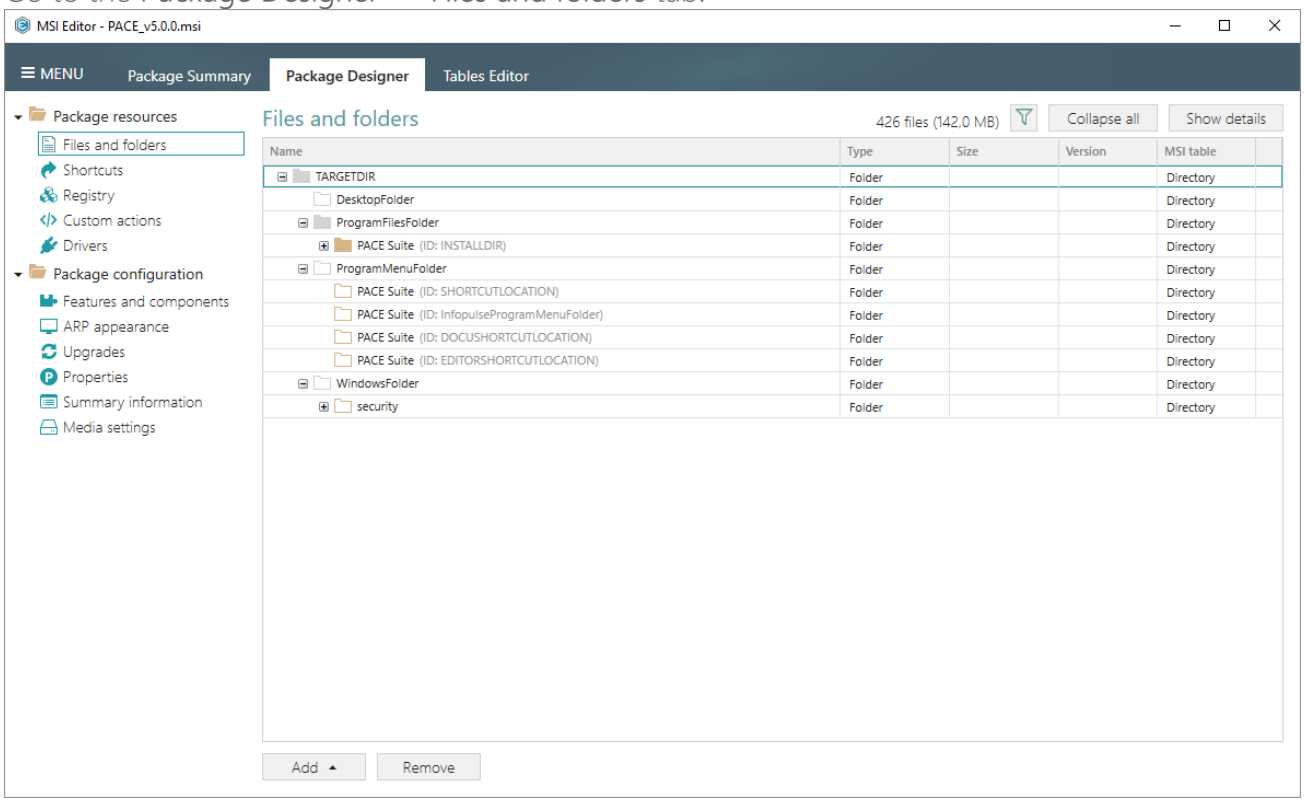
- For saving your package, select Save from the MENU.



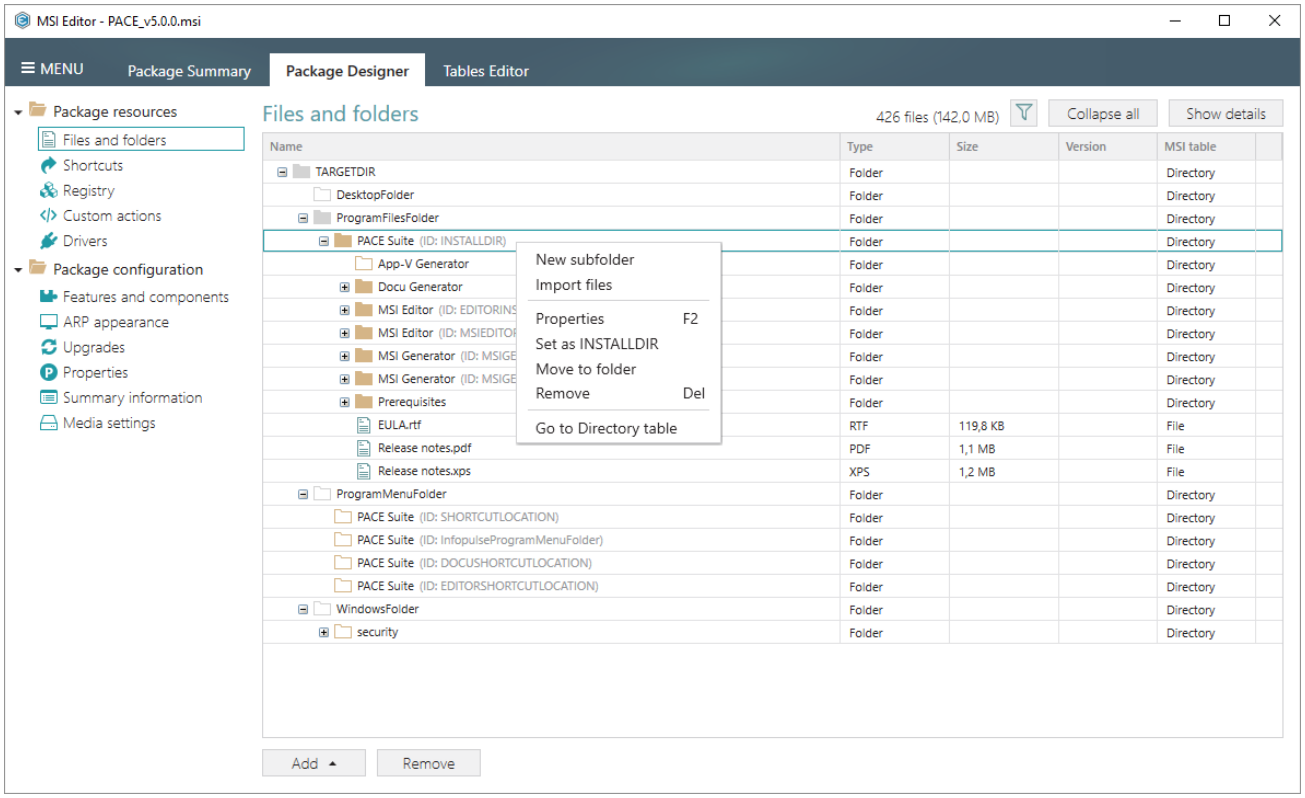
3.7.4.7 Move Files, Folders

Move a file or a folder from one folder to another. Note that moving a file of the multi-file component will result in creating a new individual component for the moved file.

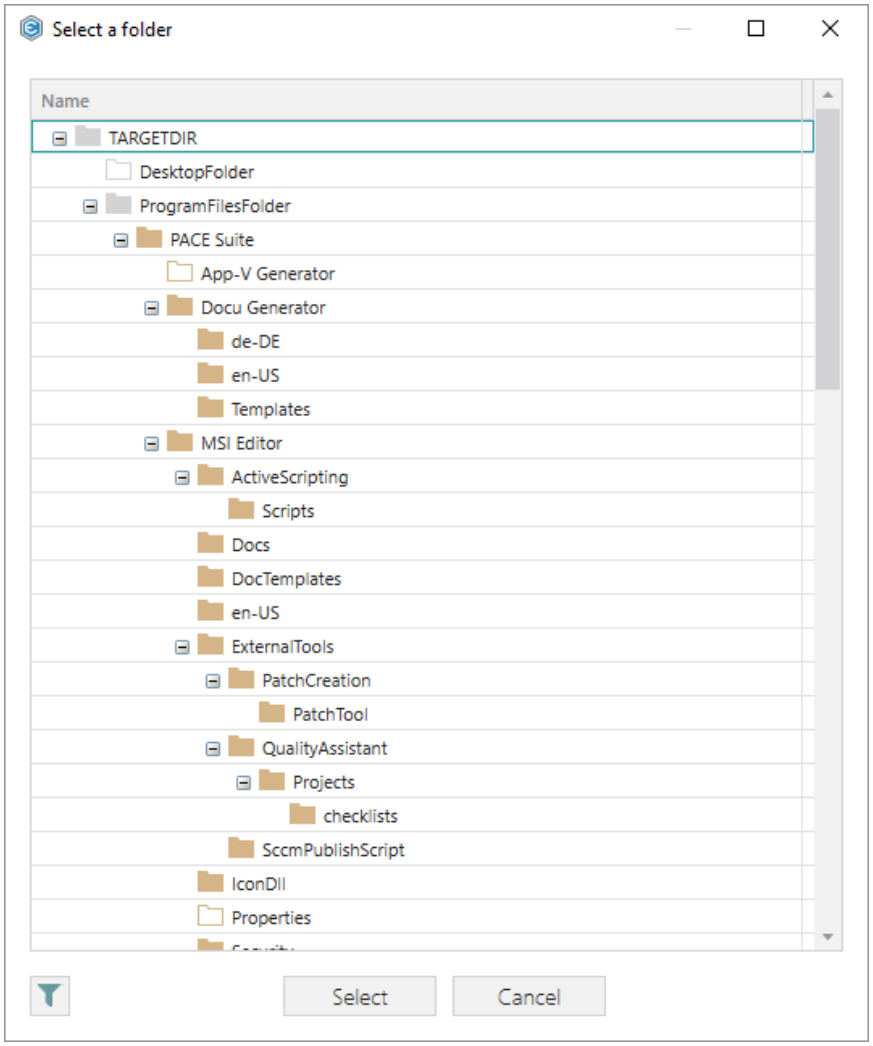
- Go to the Package Designer -> Files and folders tab.



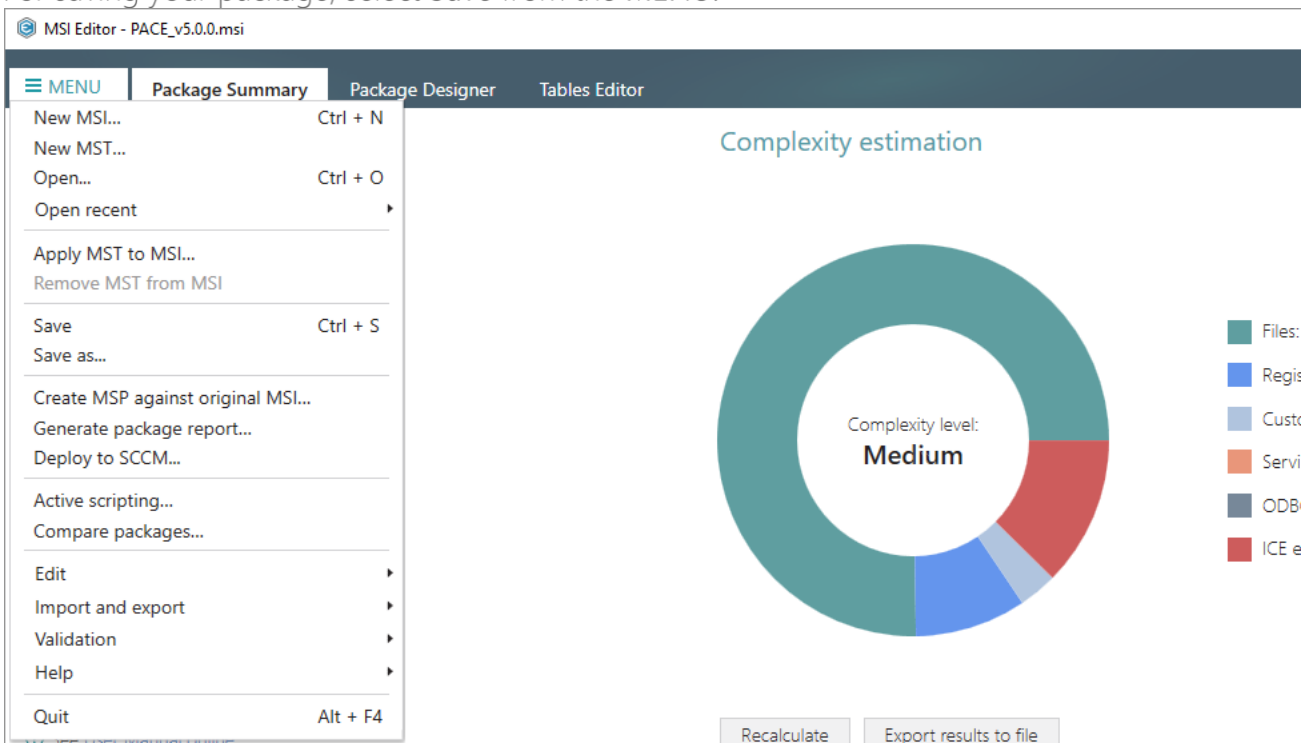
- Select Move to folder from the context menu of a file or folder, which you want to move to another folder.



- Select a new destination location from the folders tree and click Select.



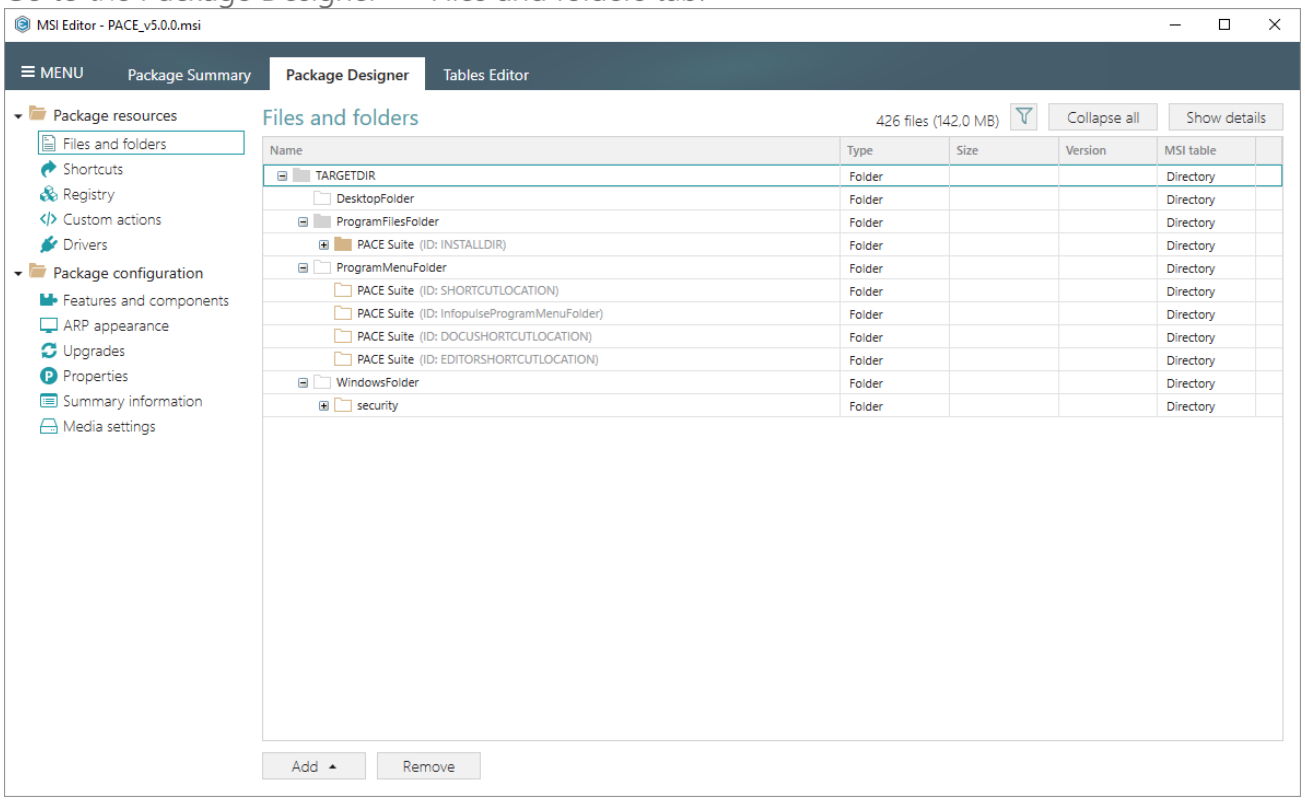
- For saving your package, select Save from the MENU.



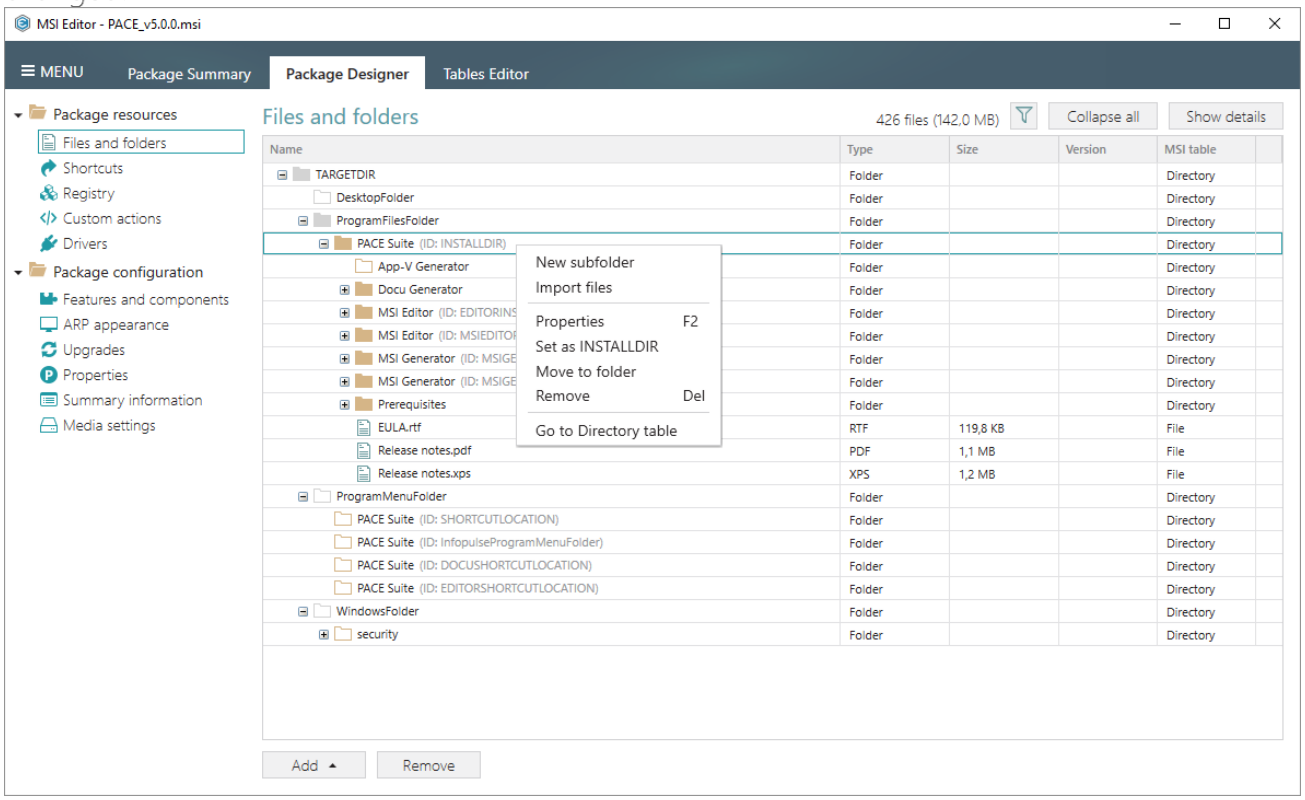
3.7.4.8 Set INSTALLDIR

Set a subfolder of the package as the INSTALLDIR. This means that an identifier of the selected folder will be changed to the 'INSTALLDIR' value. Usually, the INSTALLDIR property represents the main product installation folder, and you can re-define this path via the command line or a Custom Action.

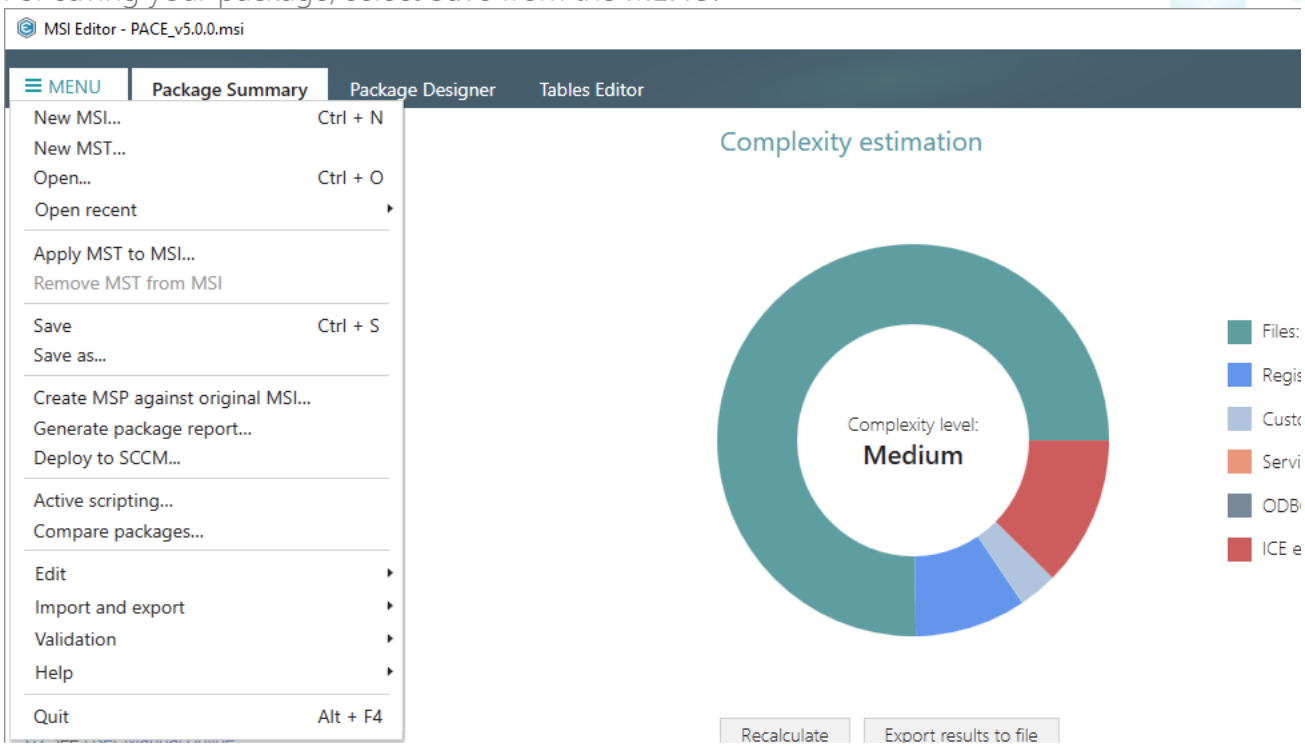
- Go to the Package Designer -> Files and folders tab.



- Select Set as INSTALLDIR from the context menu of a folder, identifier of which you want to change to 'INSTALLDIR'. NOTE Identifiers of system folders are fixed and could be not changed.



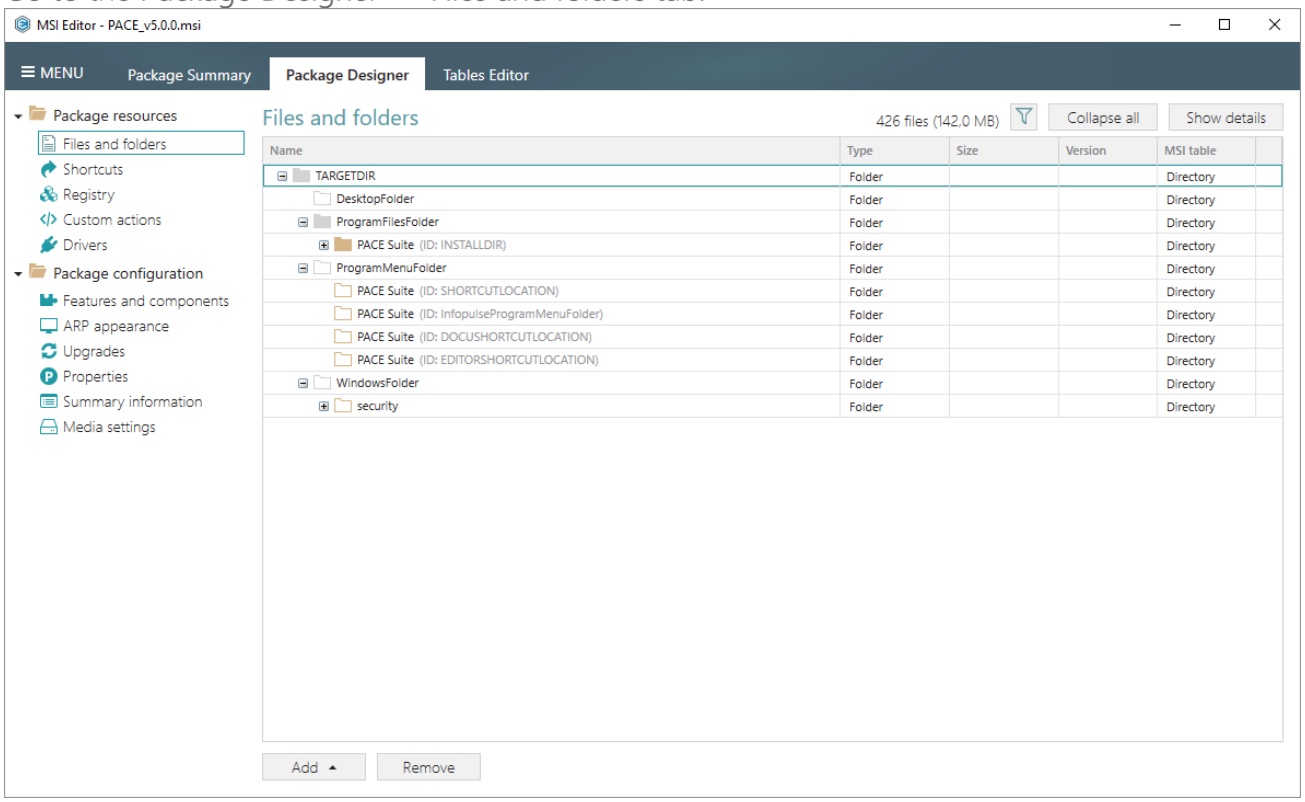
- For saving your package, select Save from the MENU.



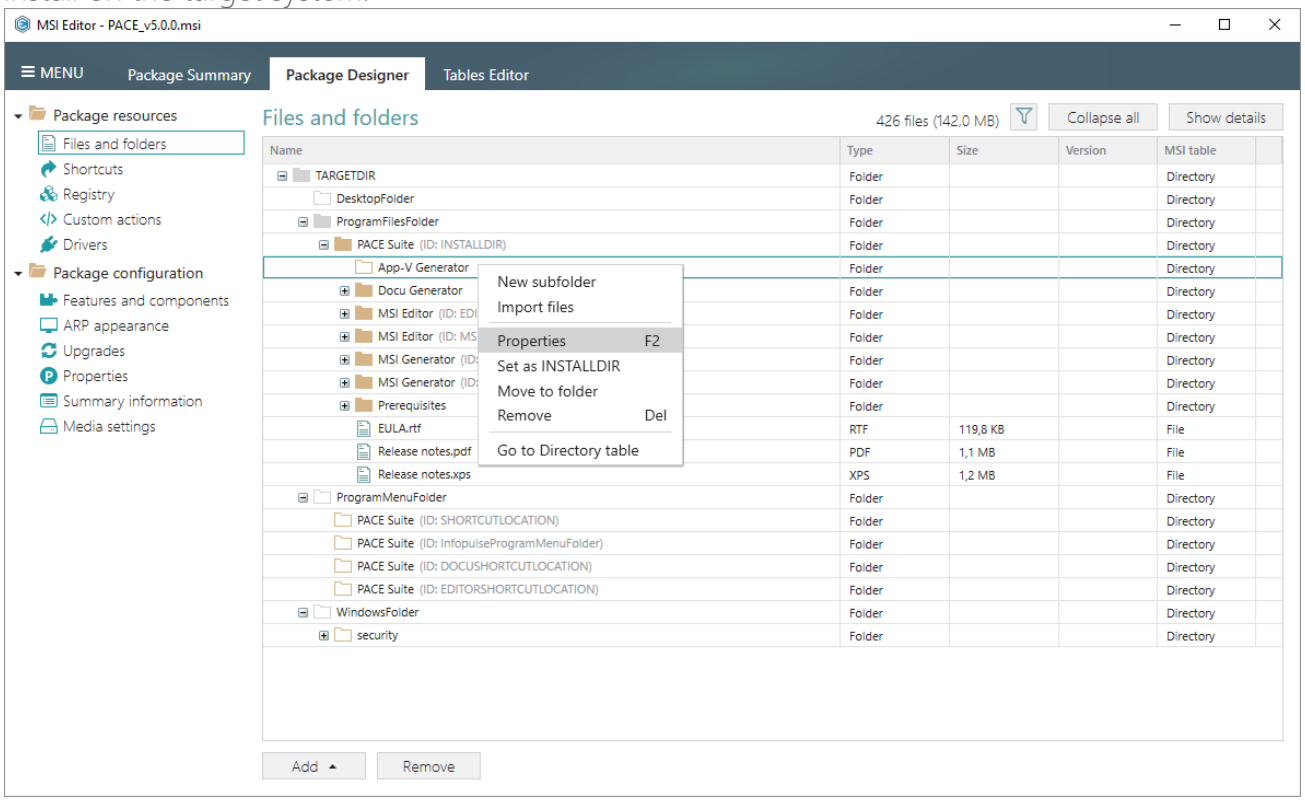
3.7.4.9 Control Empty Folder Creation

Force the empty folder creation on install. By default, Windows Installer creates only folders, which contain files.

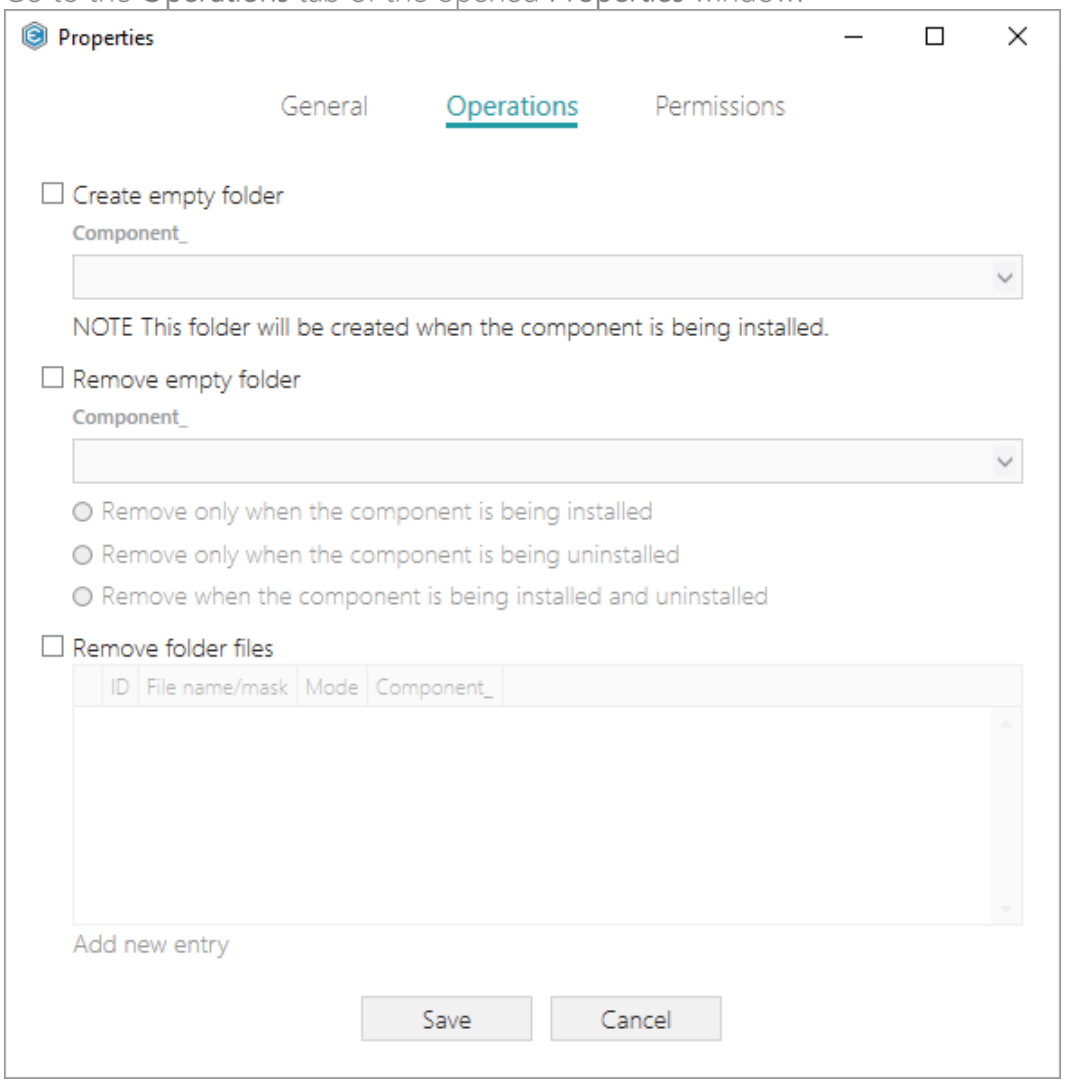
- Go to the Package Designer -> Files and folders tab.



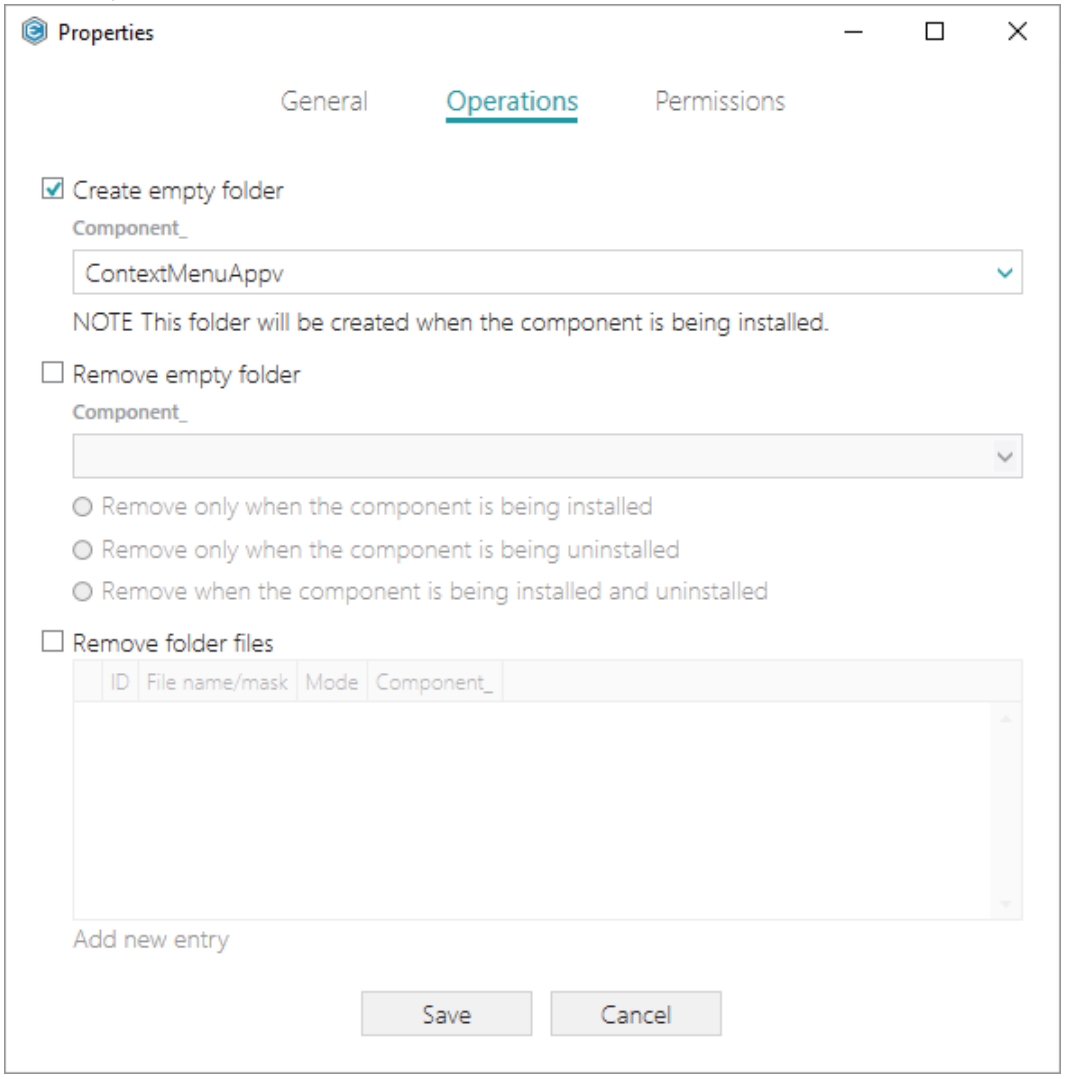
- Select **Properties** from the context menu of an empty folder, which you want to create on install on the target system.



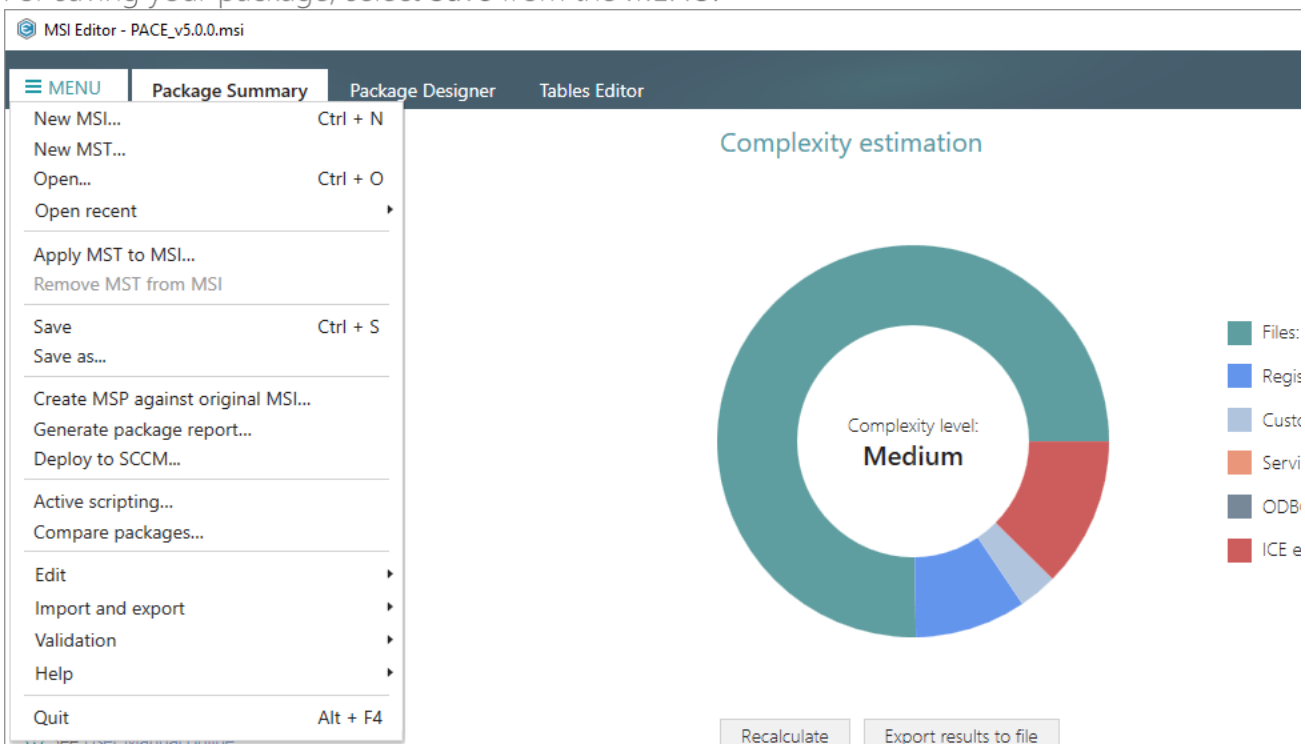
- Go to the **Operations** tab of the opened **Properties** window.



- Tick the **Create empty folder** option and select a component, which will control the folder creation, and click **Save**.



- For saving your package, select **Save** from the **MENU**.

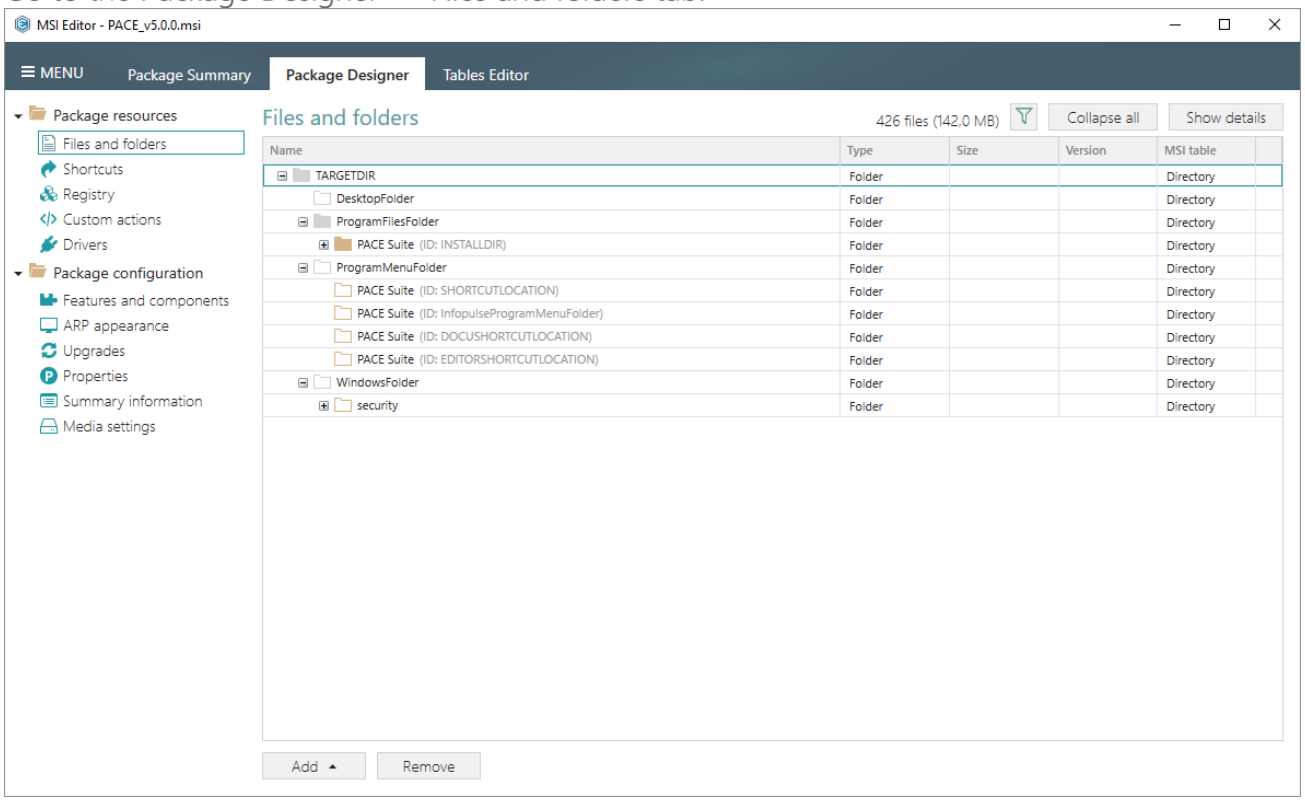


3.7.4.10 Control File/Folder Removal

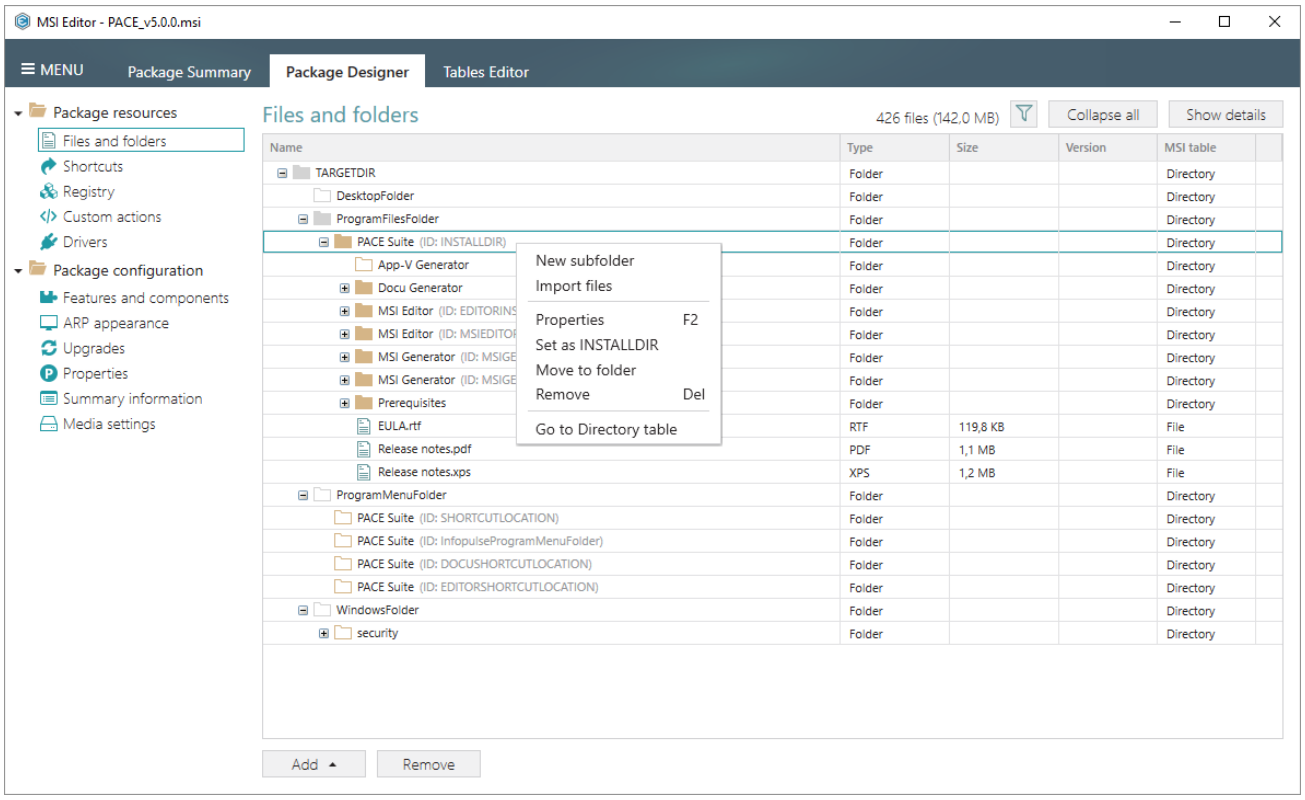
Force folder and its files removal on install or uninstall.

NOTE By default, MSI removes only those files, which it installs. This means that files, created during the application work, will remain on the system.

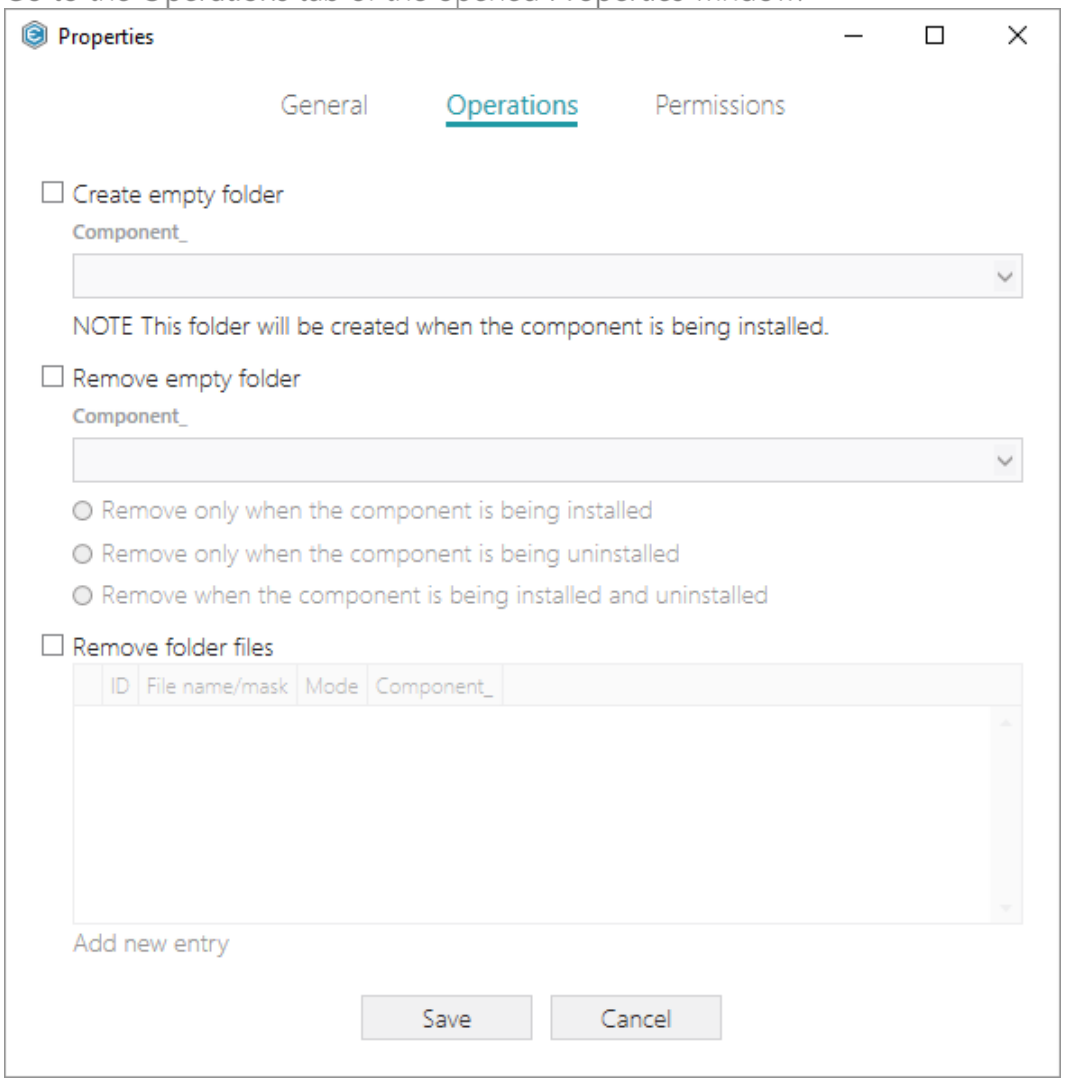
- Go to the Package Designer -> Files and folders tab.



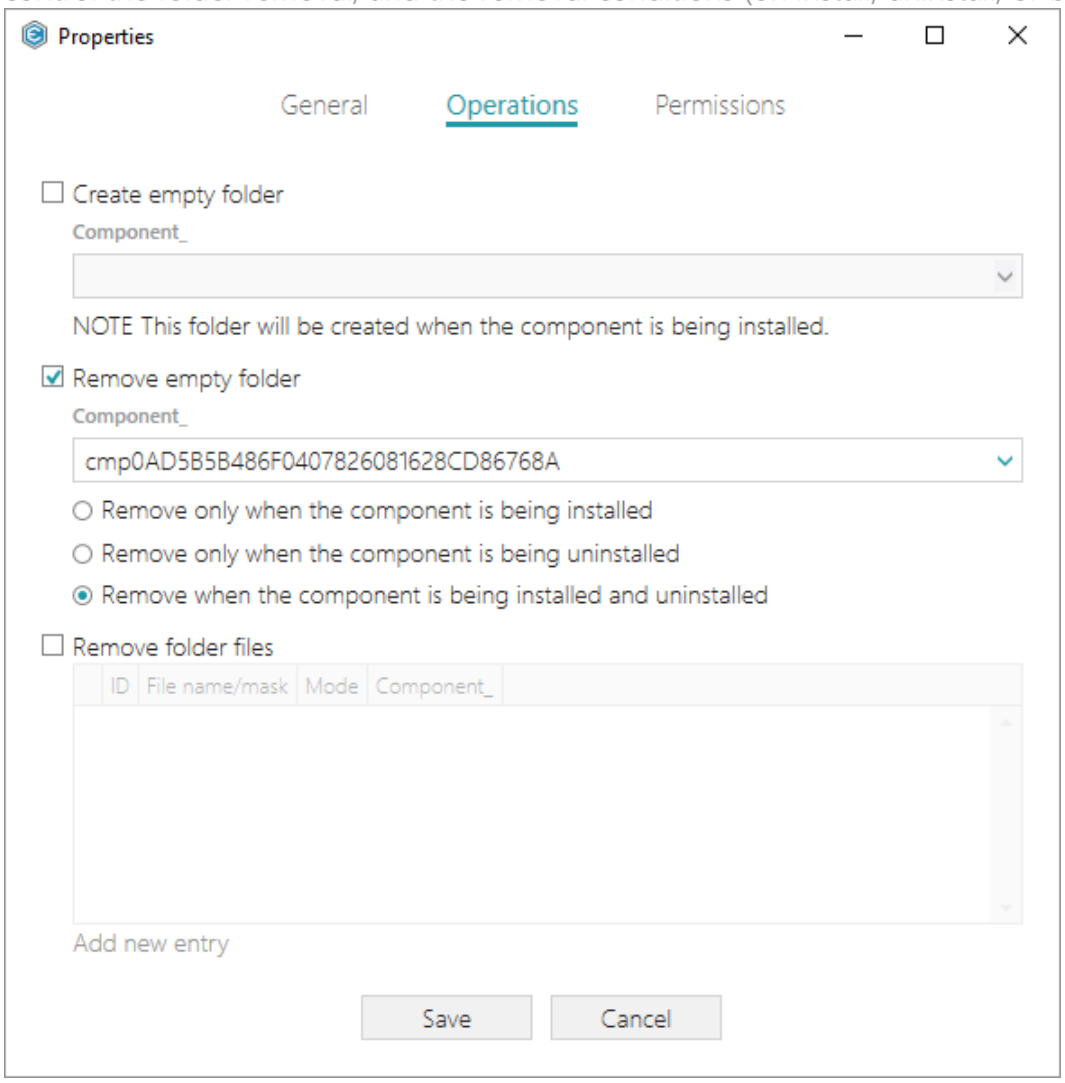
- Select Properties from the context menu of a folder, files of which or folder itself you want to remove on install or on uninstall.



- Go to the Operations tab of the opened Properties window.

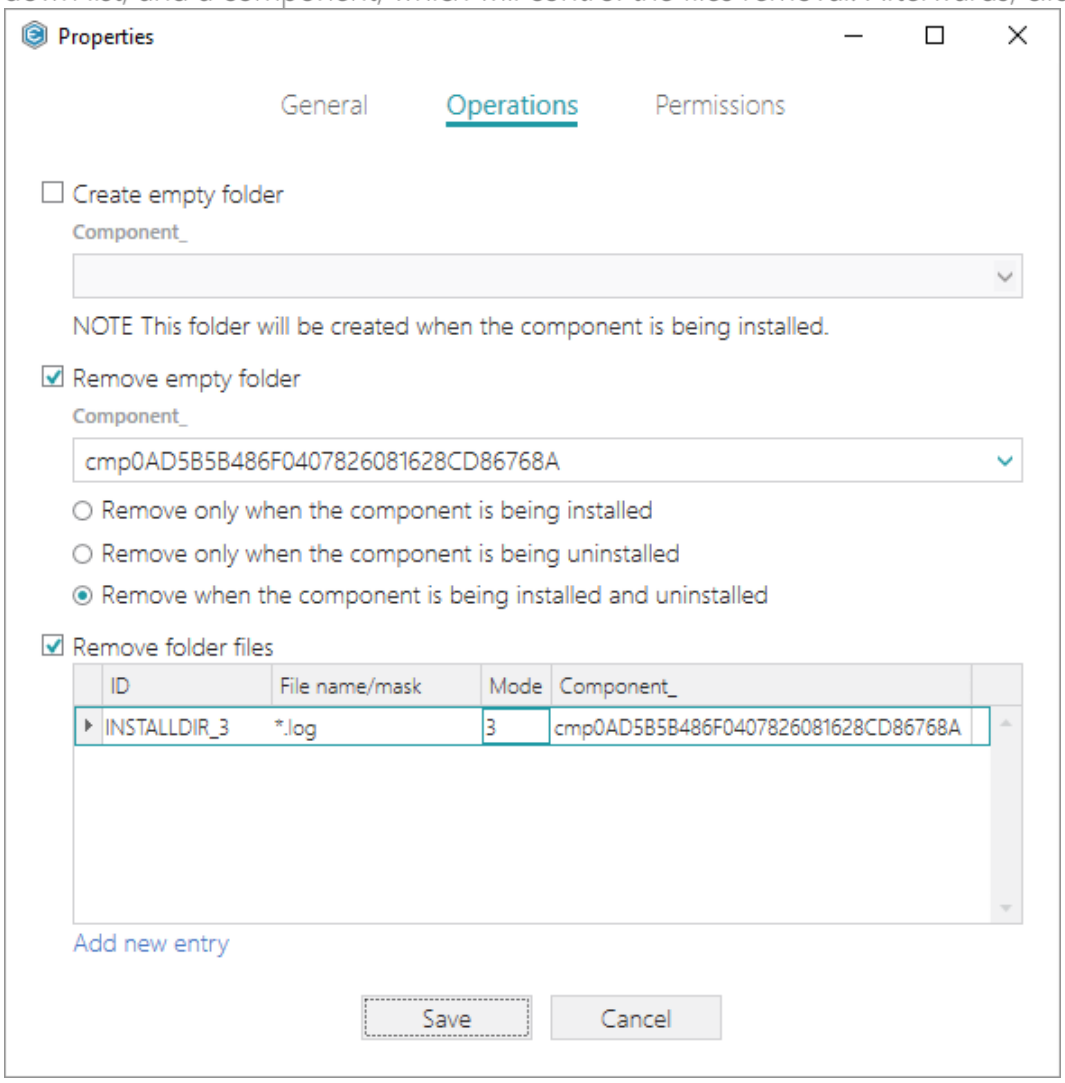


- To remove a folder, tick the **Remove empty folder** option, select a component, which will control the folder removal, and the removal conditions (on install, uninstall, or both).

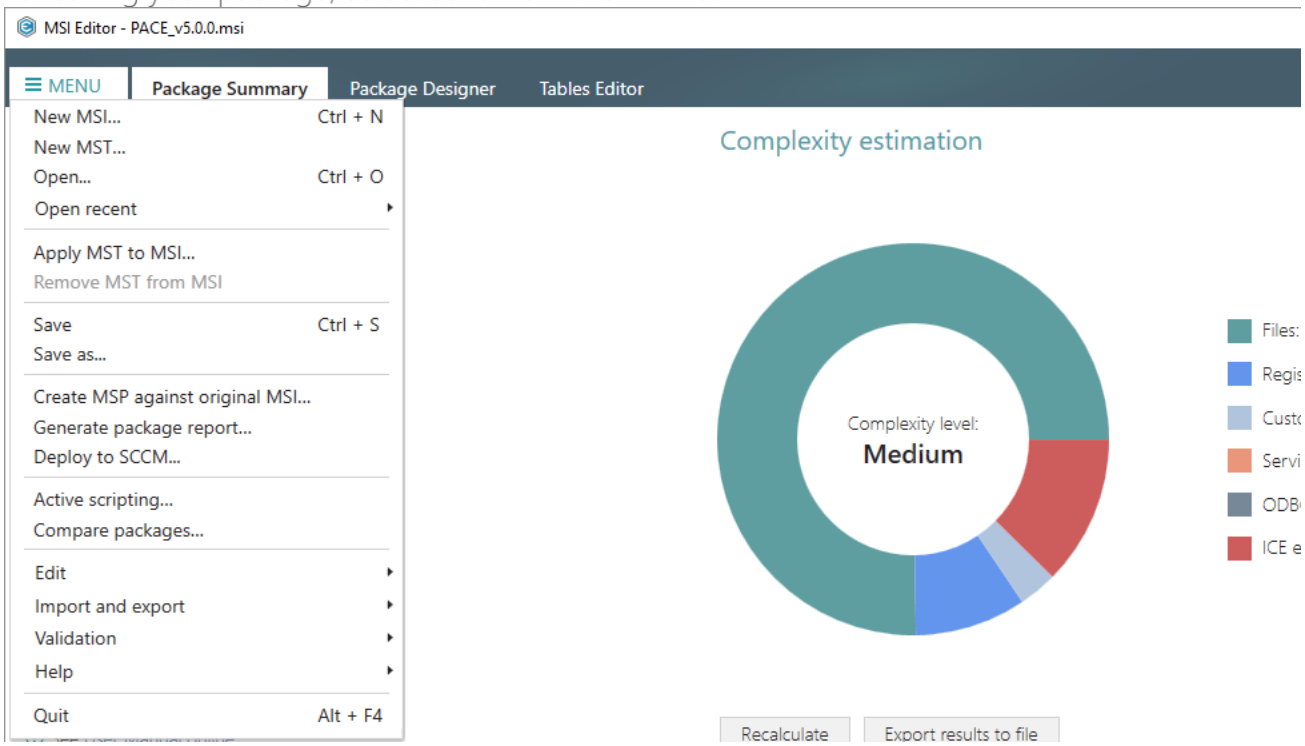


- To remove files of this folder tick the **Remove folder files** option and click **Add new entry**. Edit row cells, highlighted in red, to specify a complete file name or with wildcards to remove either a specific file or set of files, then choose removal conditions (Mode) from the drop-

down list, and a component, which will control the files removal. Afterwards, click Save.



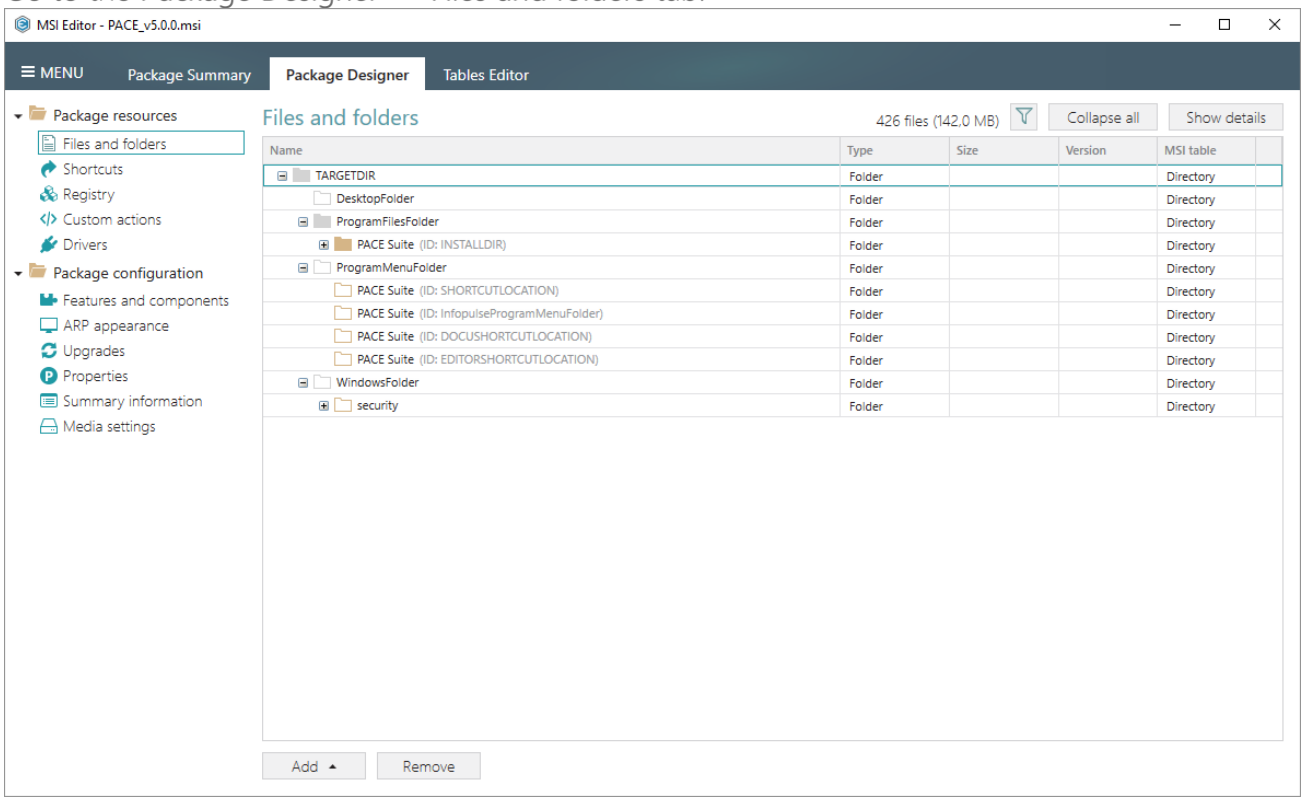
- For saving your package, select Save from the MENU.



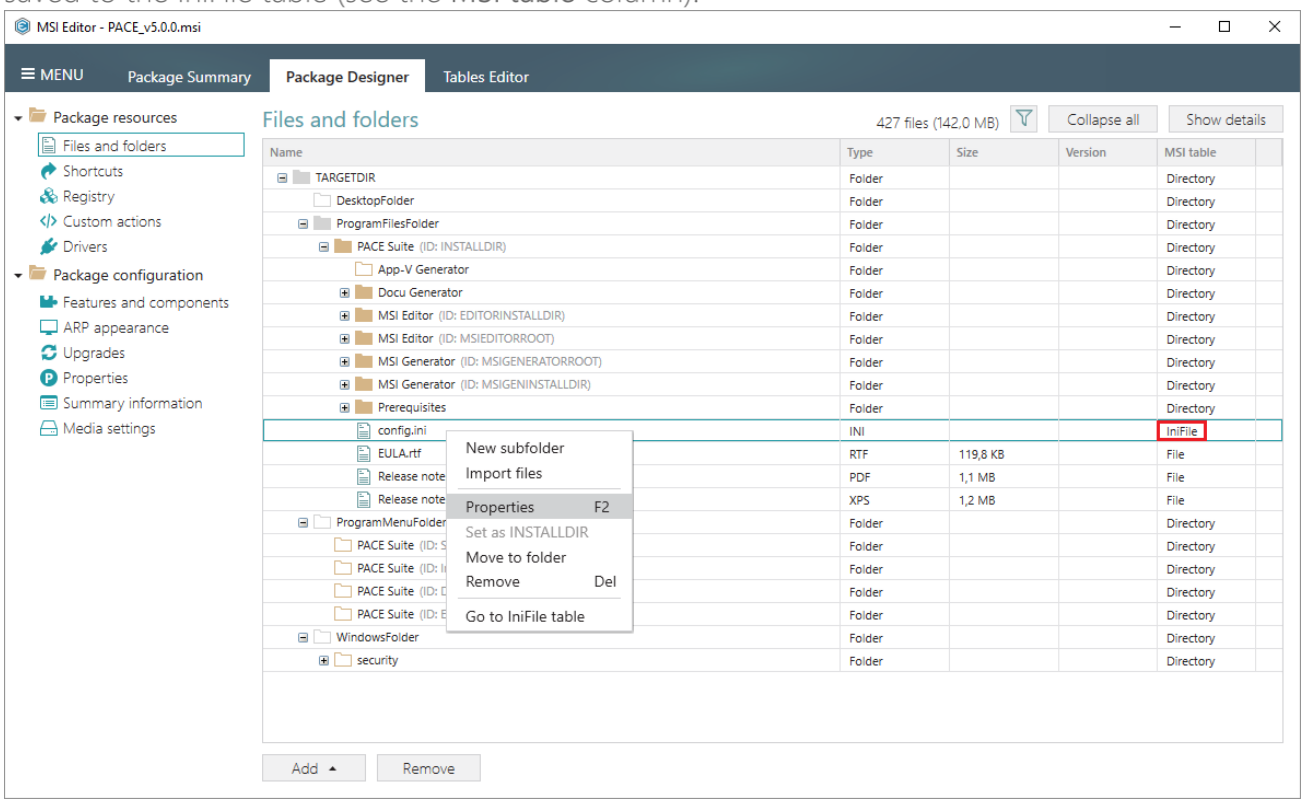
3.7.4.11 Edit INI File

Edit a name, destination location and content of the INI file, saved to the IniFile table. Note that some INI files could be saved to the File table as binary files and their content could not be modified using this approach.

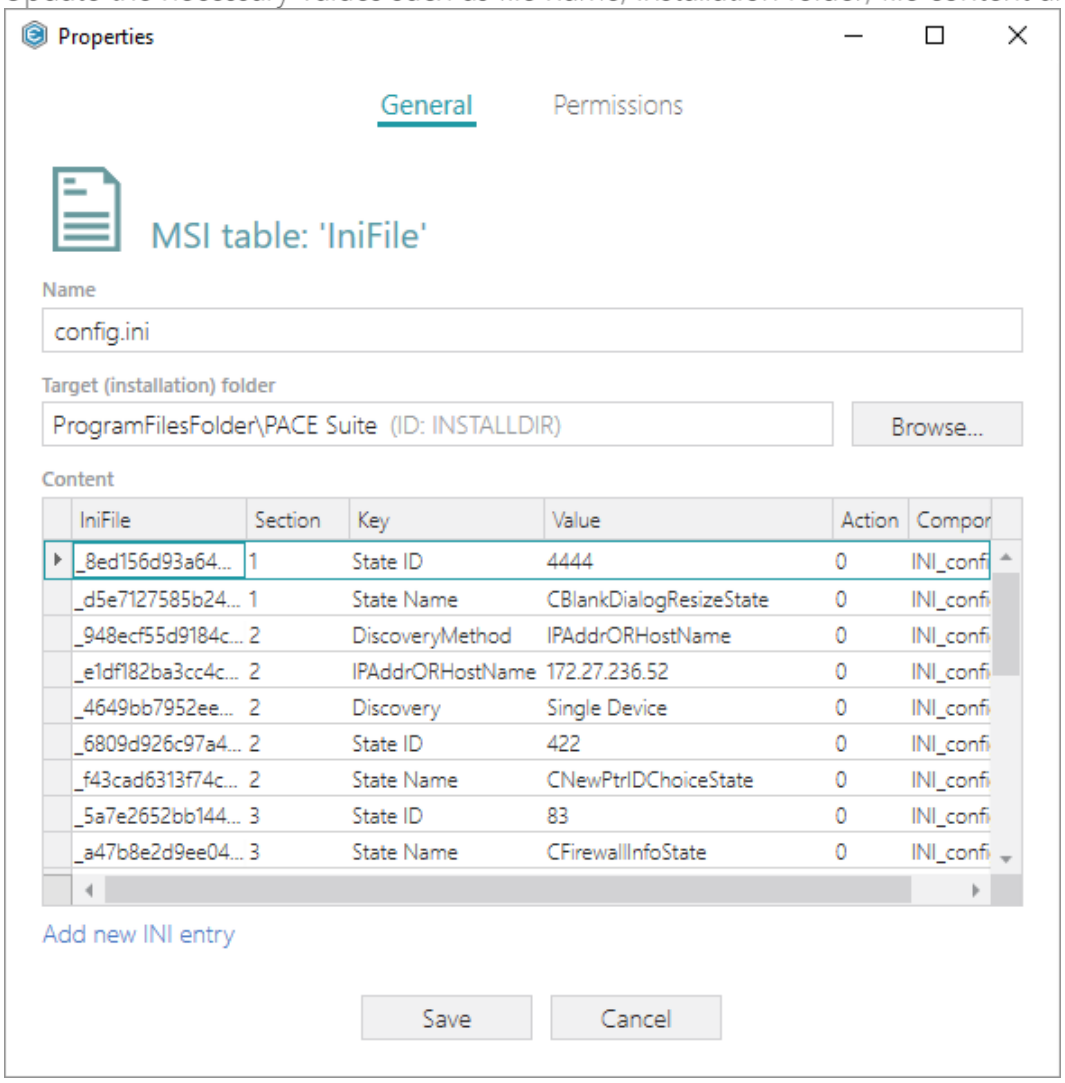
- Go to the **Package Designer** -> **Files and folders** tab.



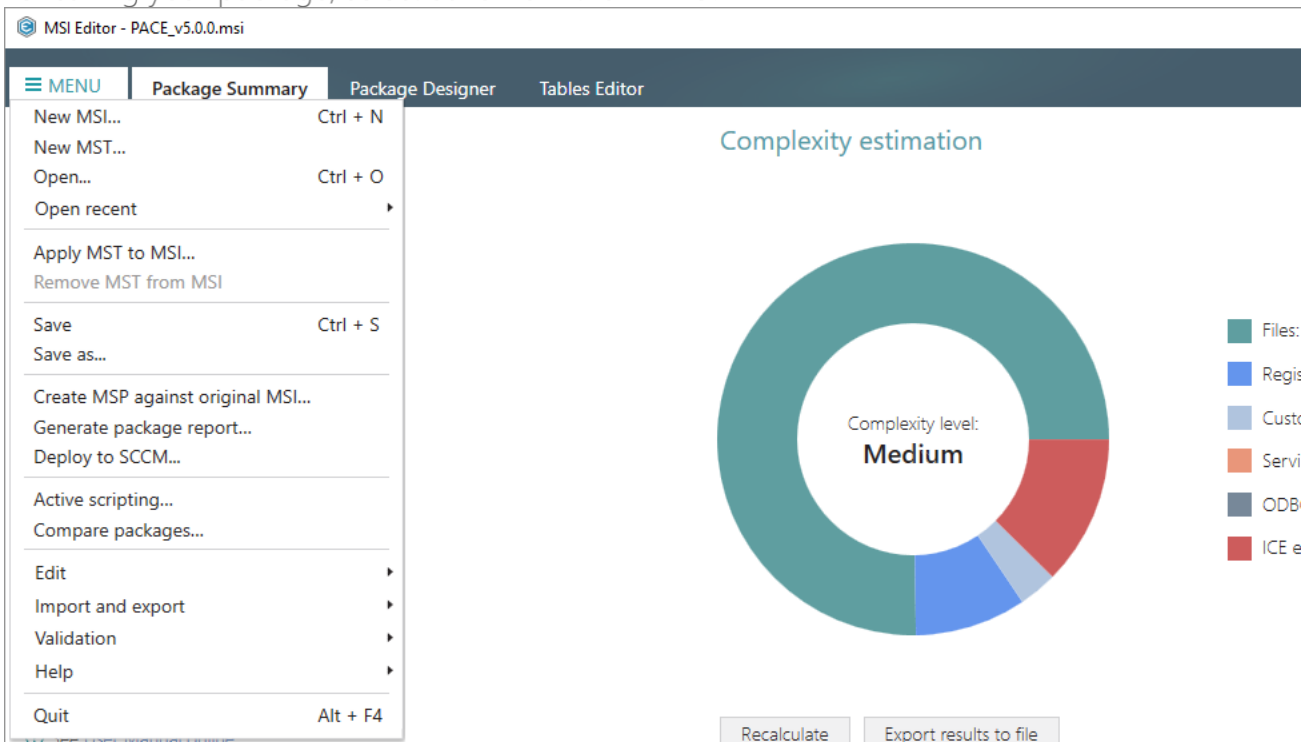
- Select **Properties** from the context menu of an INI file, which you want to edit.
  NOTE Using this approach you can review and modify content only those INI files, which are saved to the IniFile table (see the **MSI table** column).



- Update the necessary values such as file name, installation folder, file content and click Save.



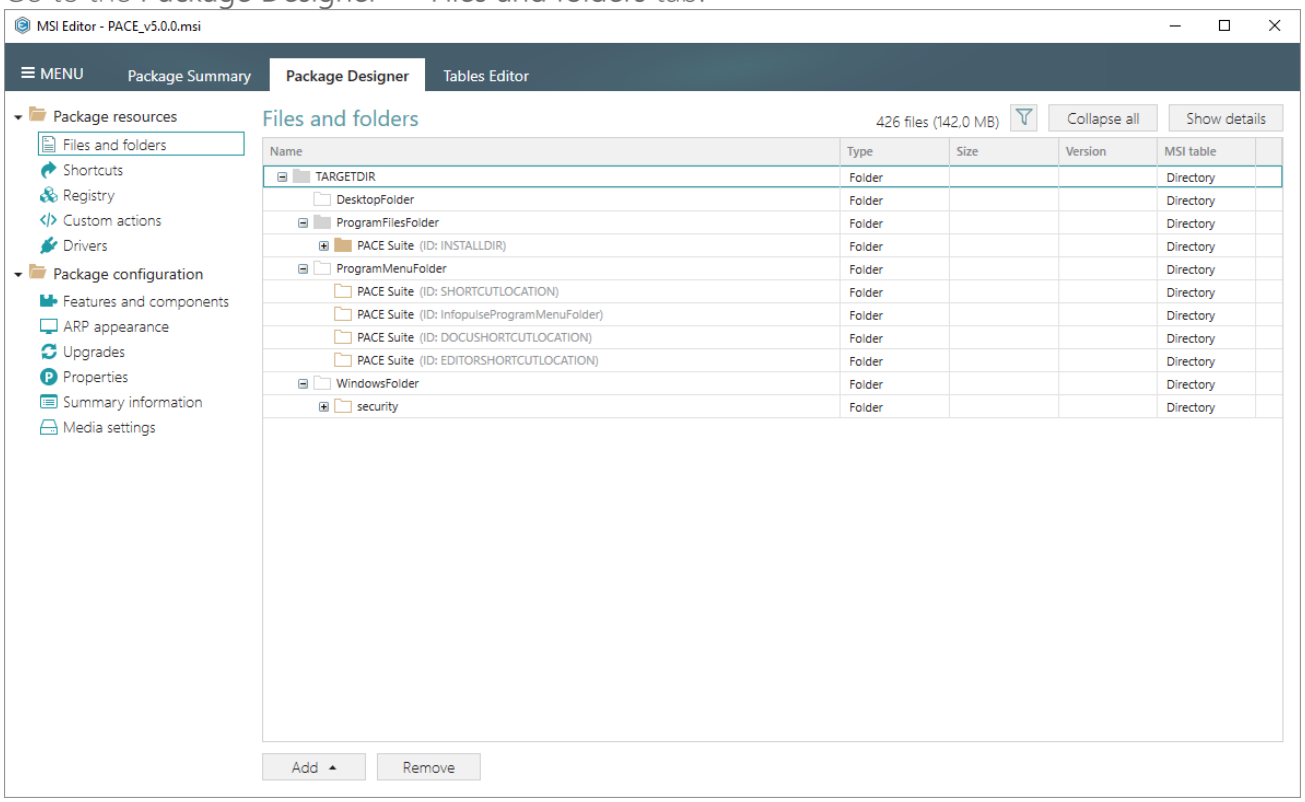
- For saving your package, select Save from the MENU.



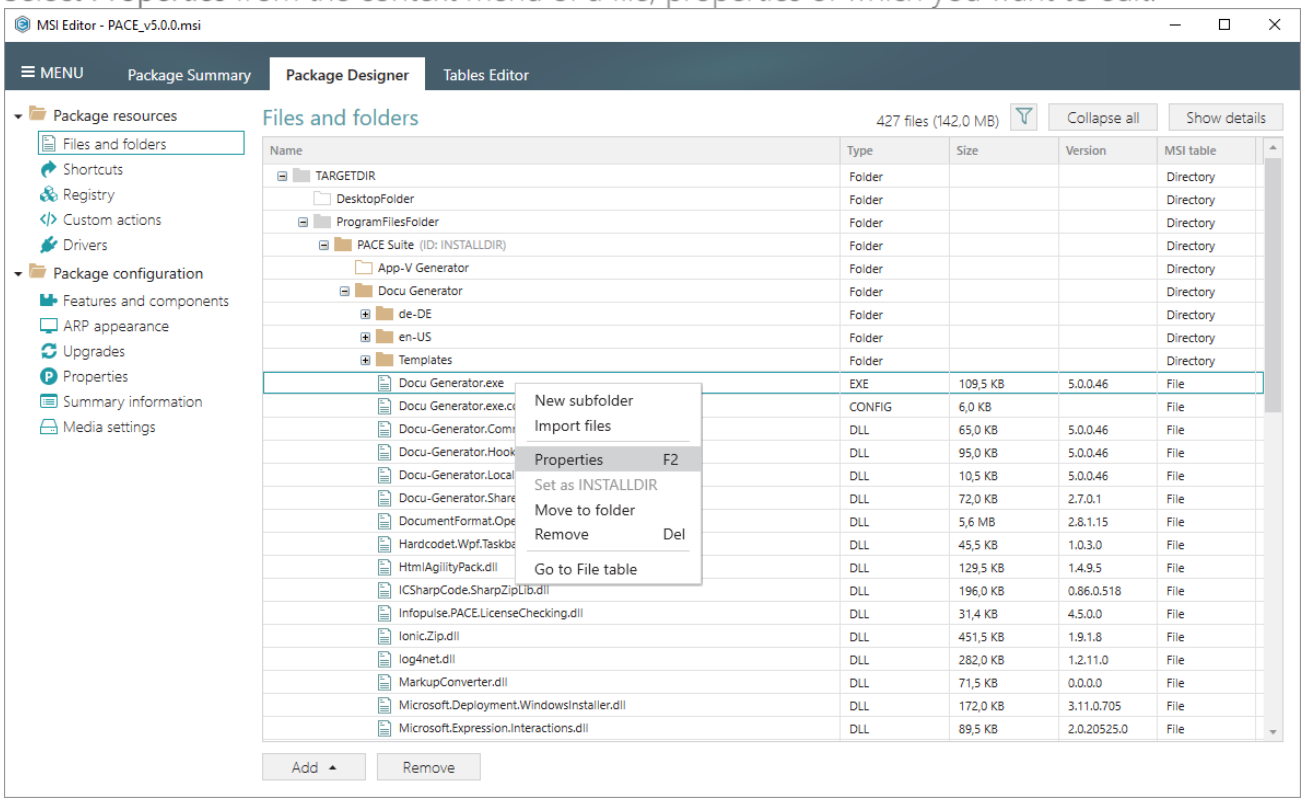
3.7.4.12 Edit File Properties

Edit a name, identifier, destination location, component and many other attributes of the file, saved to the File table.

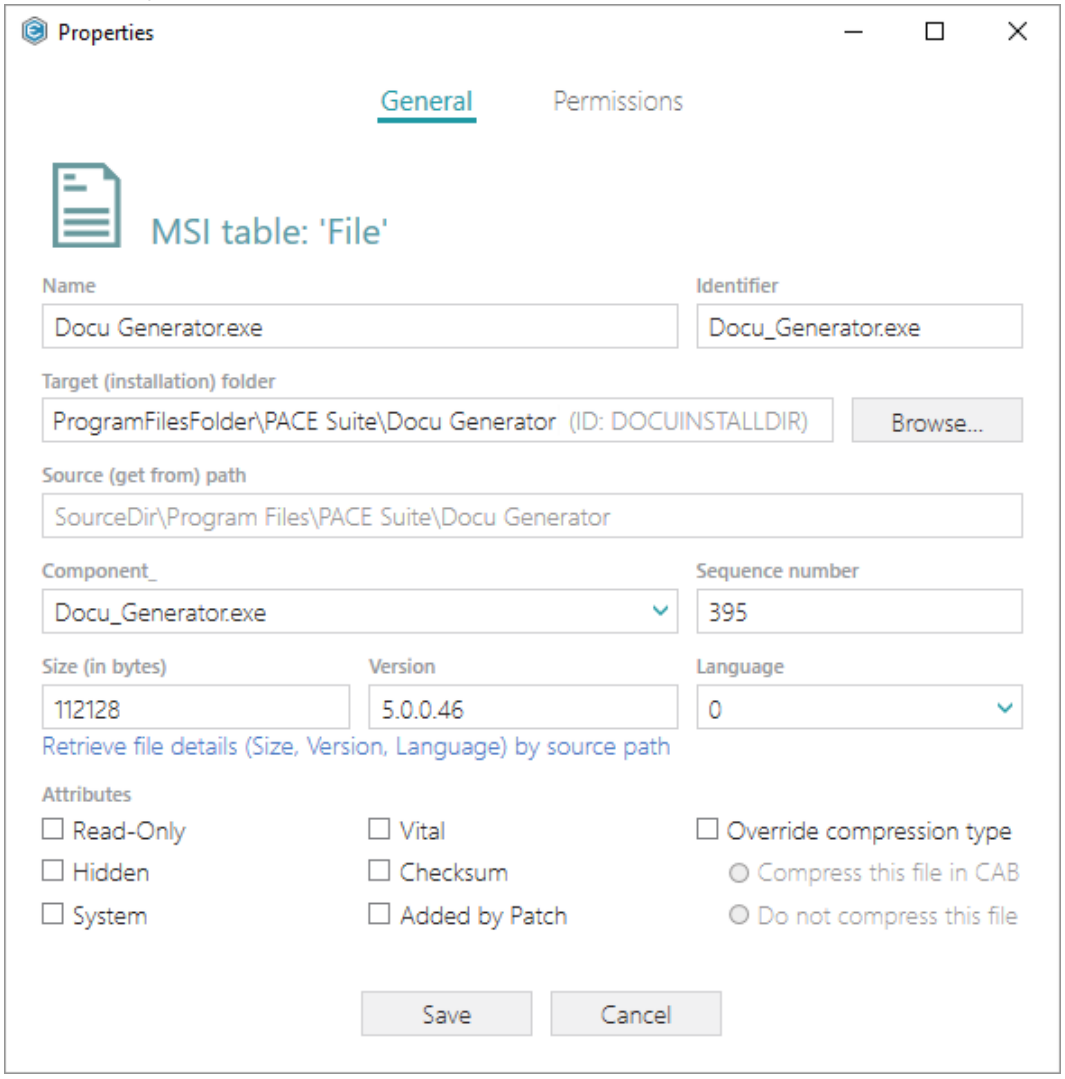
- Go to the Package Designer -> Files and folders tab.



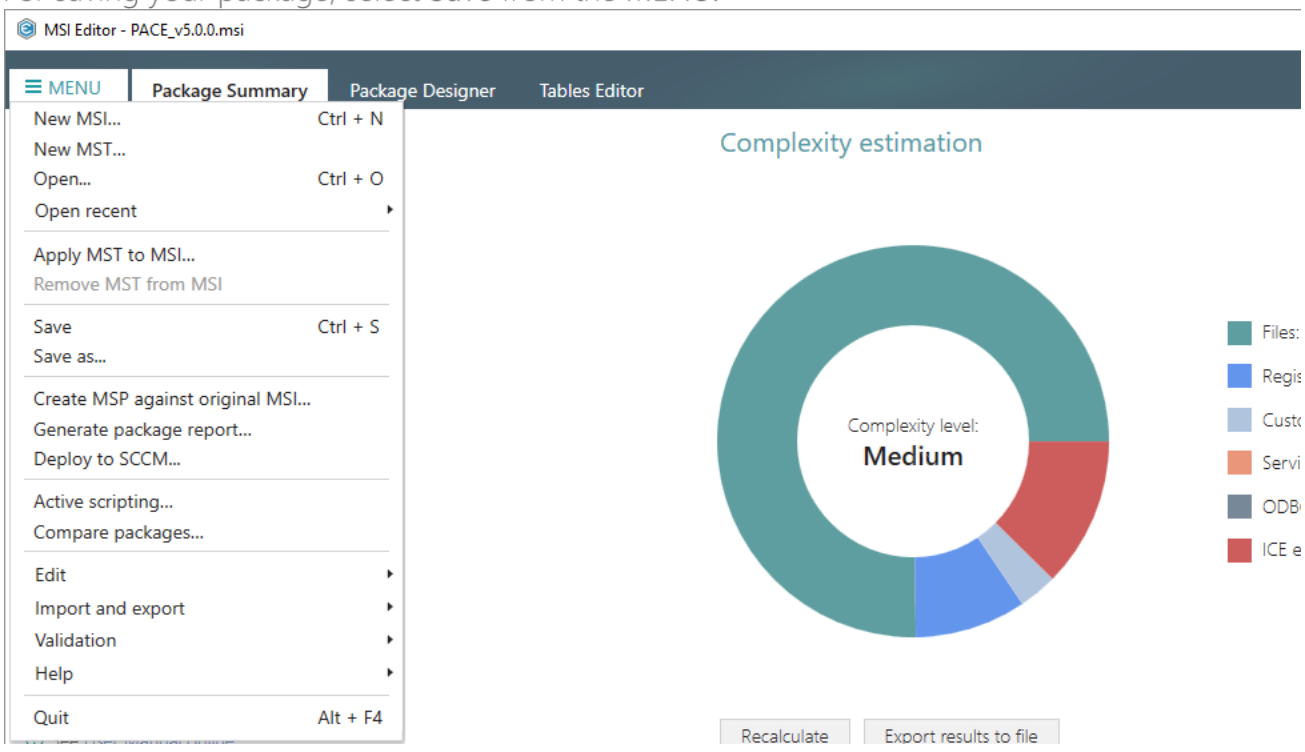
- Select Properties from the context menu of a file, properties of which you want to edit.



- Update the necessary values such as file name, installation folder, attributes and so on. Afterwards, click Save.



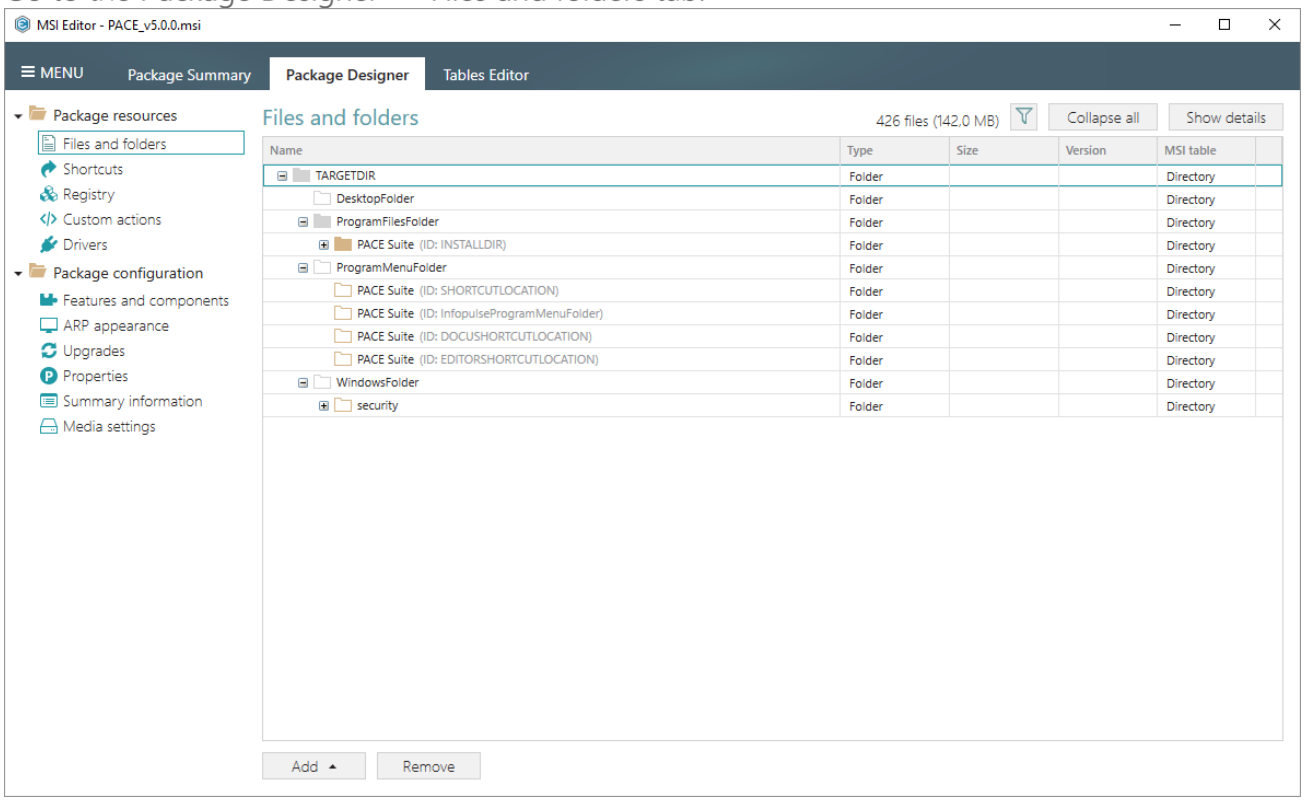
- For saving your package, select Save from the MENU.



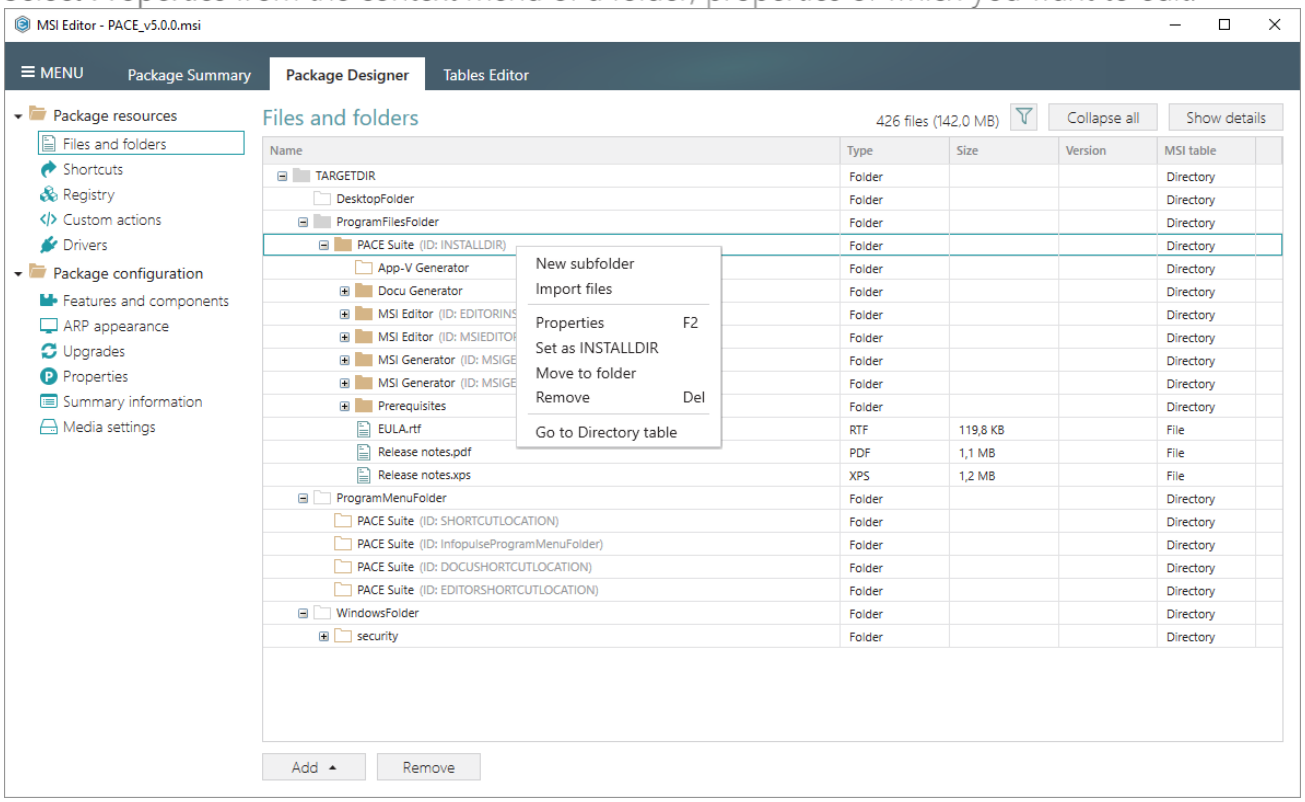
3.7.4.13 Edit Folder Properties

Edit a folder target and source names, identifier, and its parent folder.

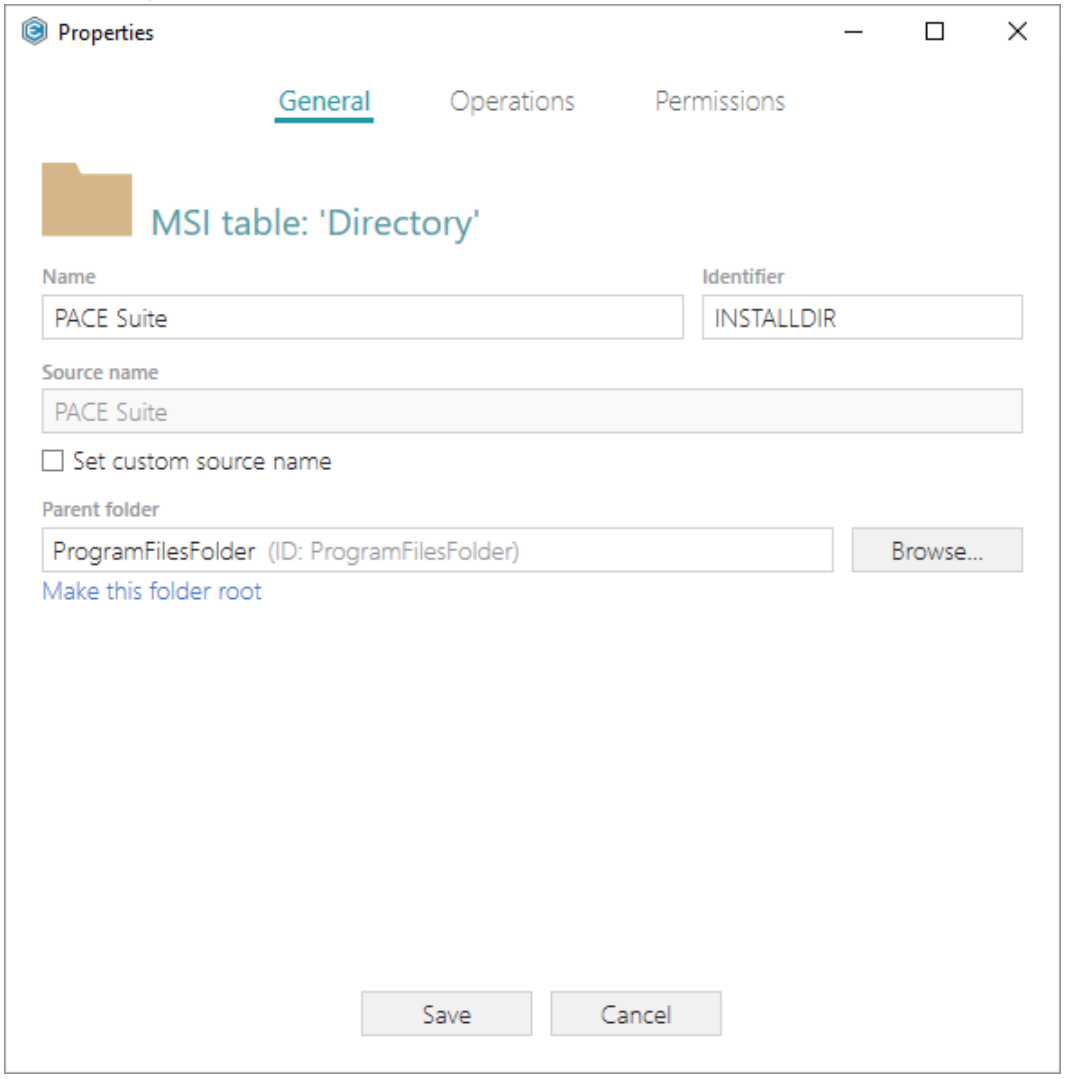
- Go to the Package Designer -> Files and folders tab.



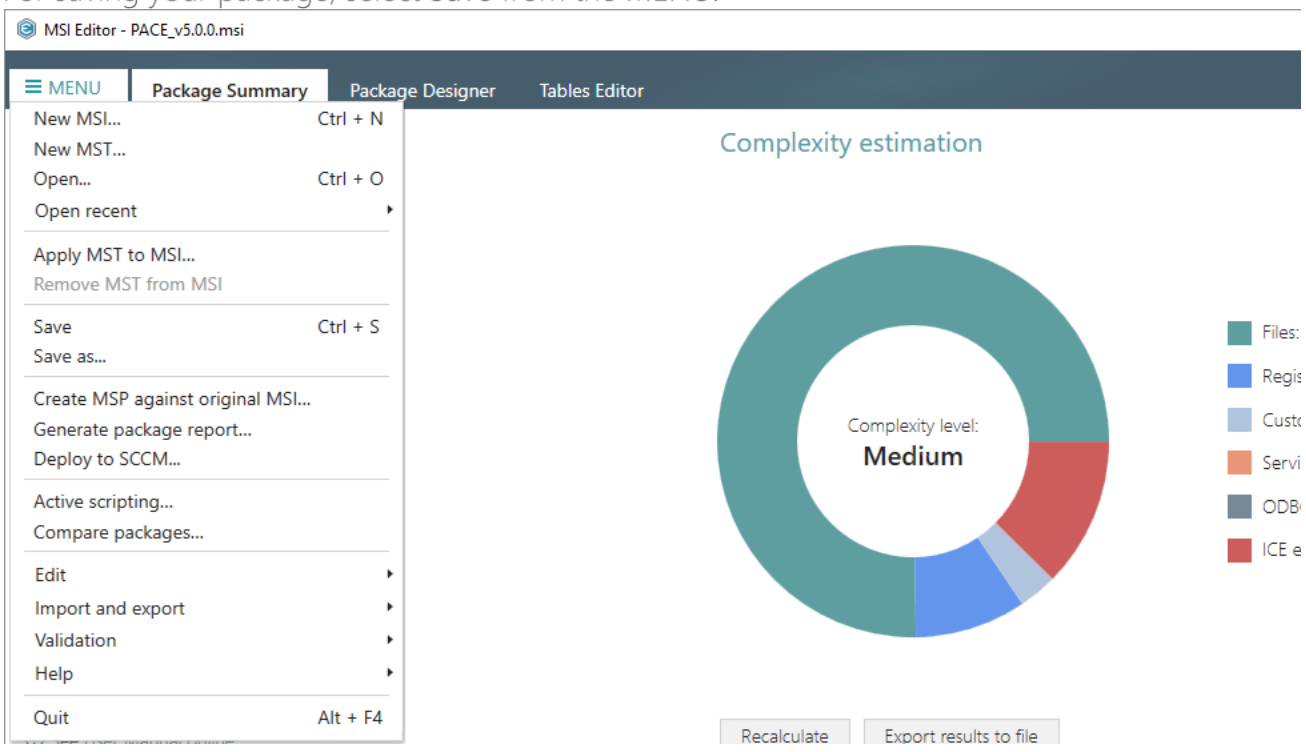
- Select Properties from the context menu of a folder, properties of which you want to edit.



- Update the necessary values such as folder target and source names, its parent and so on. Afterwards, click Save.



- For saving your package, select Save from the MENU.



3.7.5 Shortcuts

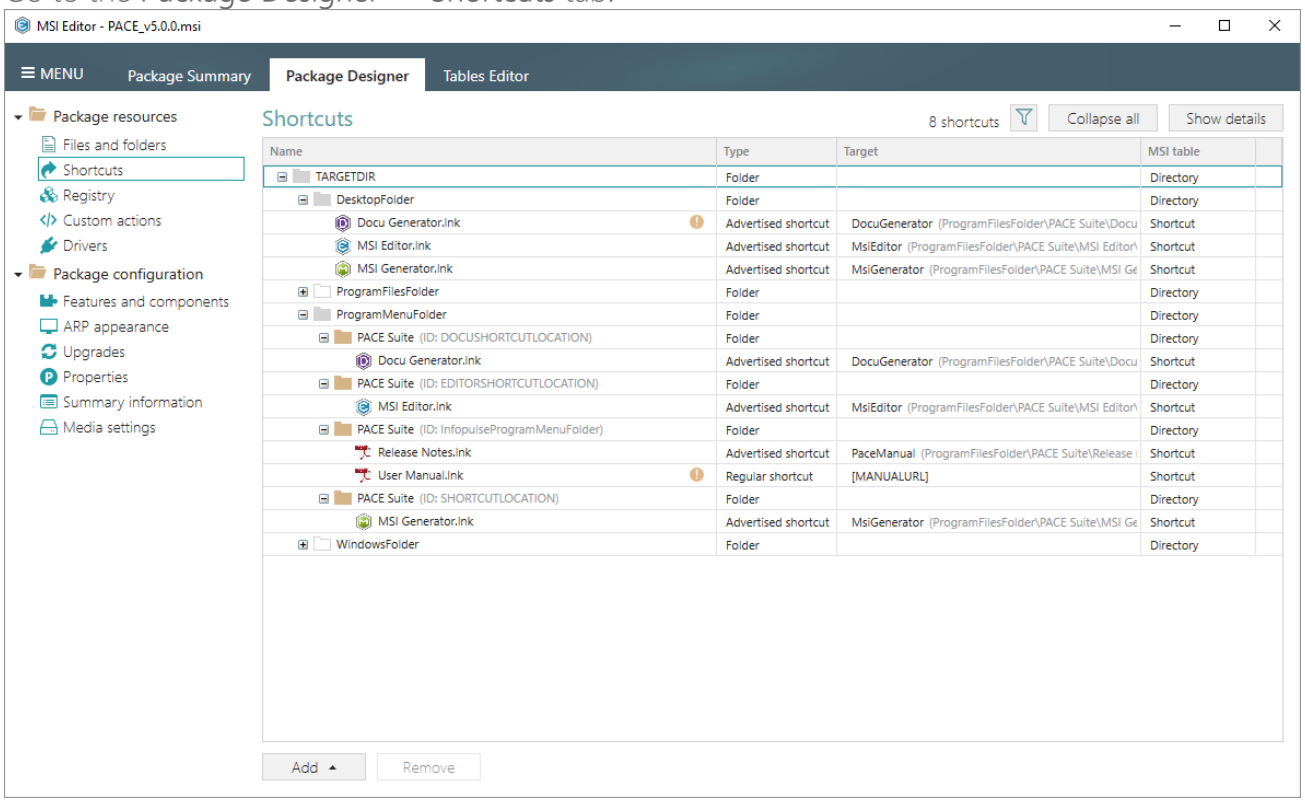
Choose a scenario that better suits your needs:

- Add Shortcut, described in section 3.7.5.1
  Add a regular or an advertised shortcut that points to a file, folder or URL.
- Import Shortcuts, described in section 3.7.5.2
  Import shortcuts from the file system to the package.
- Move Shortcut, described in section 3.7.5.3
  Move a shortcut to another folder.
- Edit Shortcut Properties, described in section 3.7.5.4
  Edit shortcut details such as icon, name, installation folder, shortcut type, arguments, description and the like.

3.7.5.1 Add Shortcut

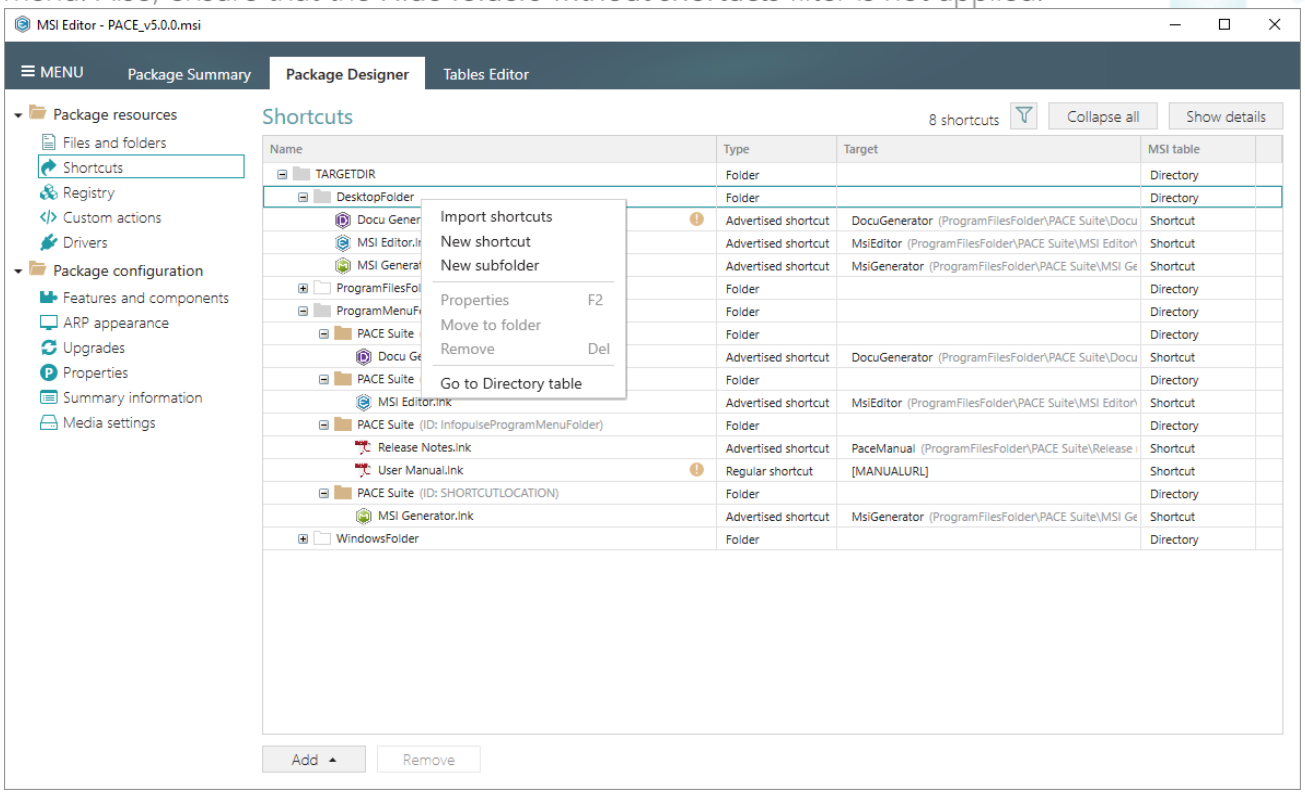
Add a regular or an advertised shortcut that points to a file, folder or URL.

- Go to the Package Designer -> Shortcuts tab.

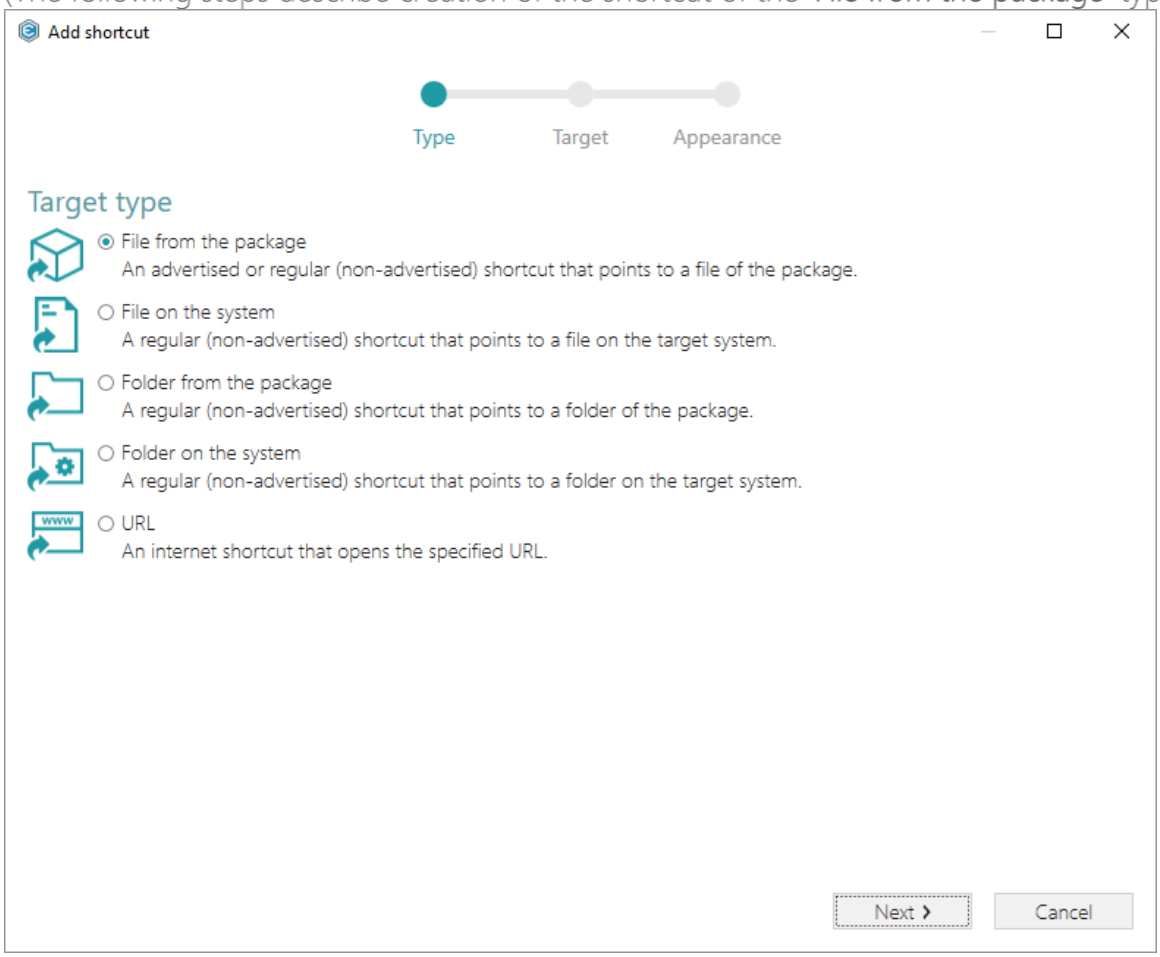


- Select New shortcut from the context menu of a folder, where you want to create a shortcut.
  NOTE To add the missing system folders (e.g. DesktopFolder, ProgramMenuFolder) or subfolders, select Add -> System folders or Add -> Subfolder respectively from the toolbox

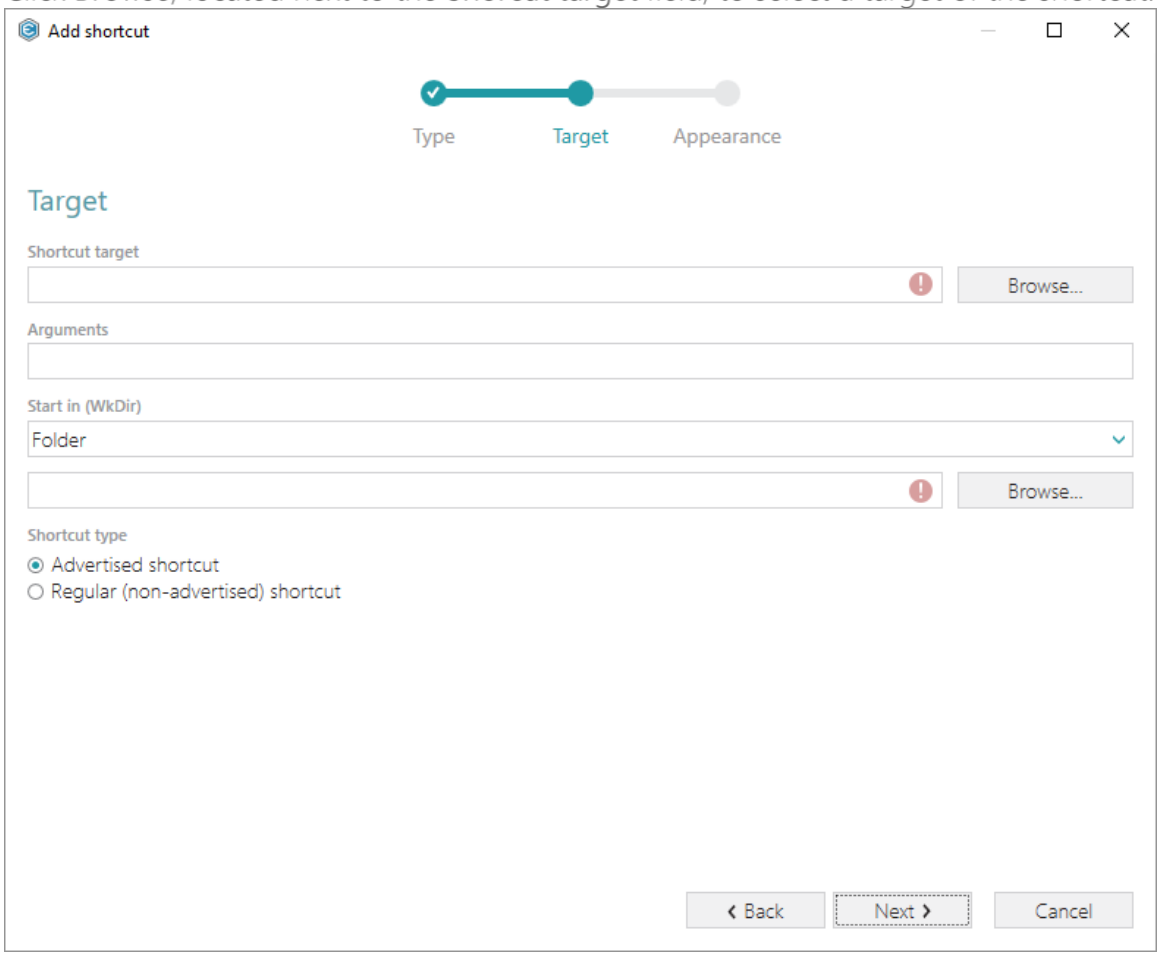
menu. Also, ensure that the Hide folders without shortcuts filter is not applied.



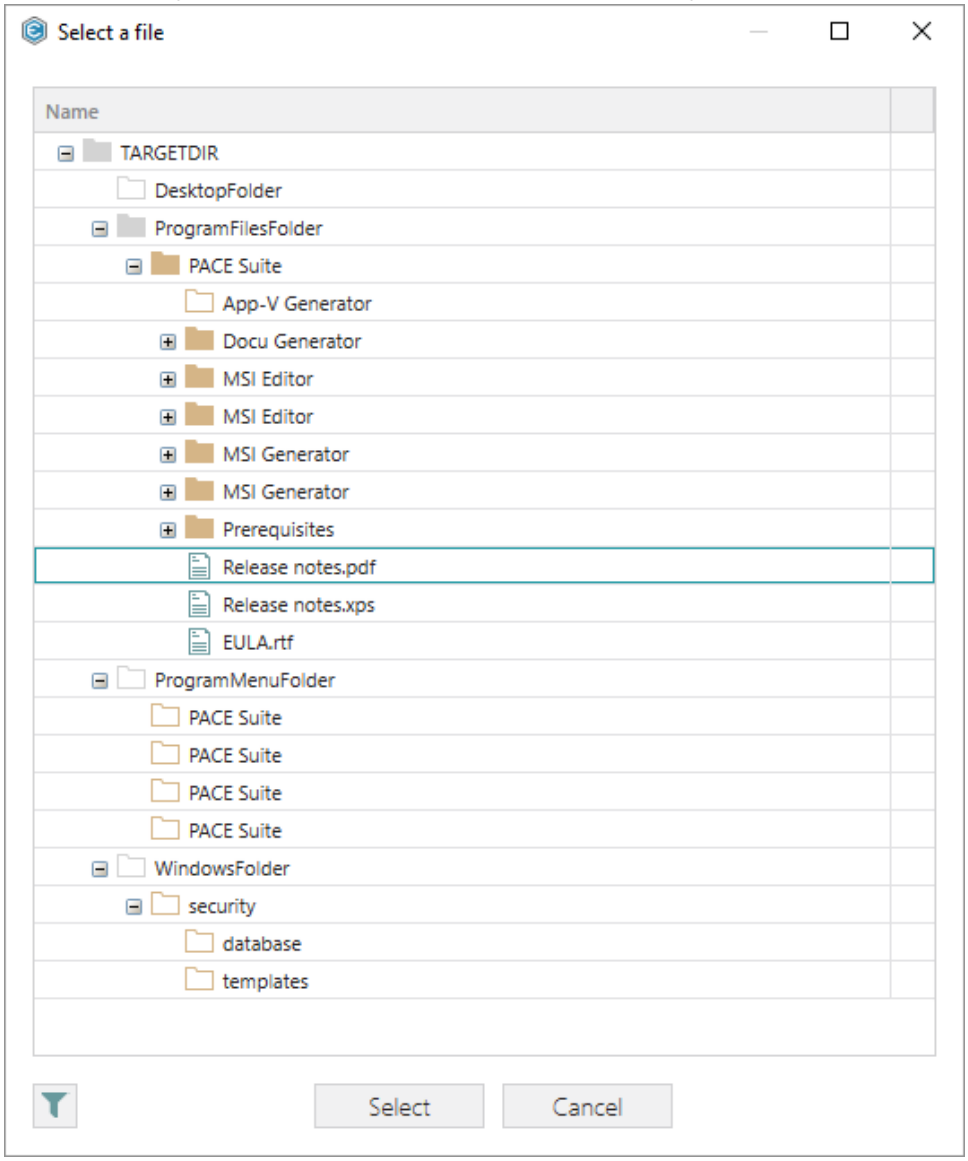
- Select a type of the shortcut target and click Next.
(The following steps describe creation of the shortcut of the 'File from the package' type.)



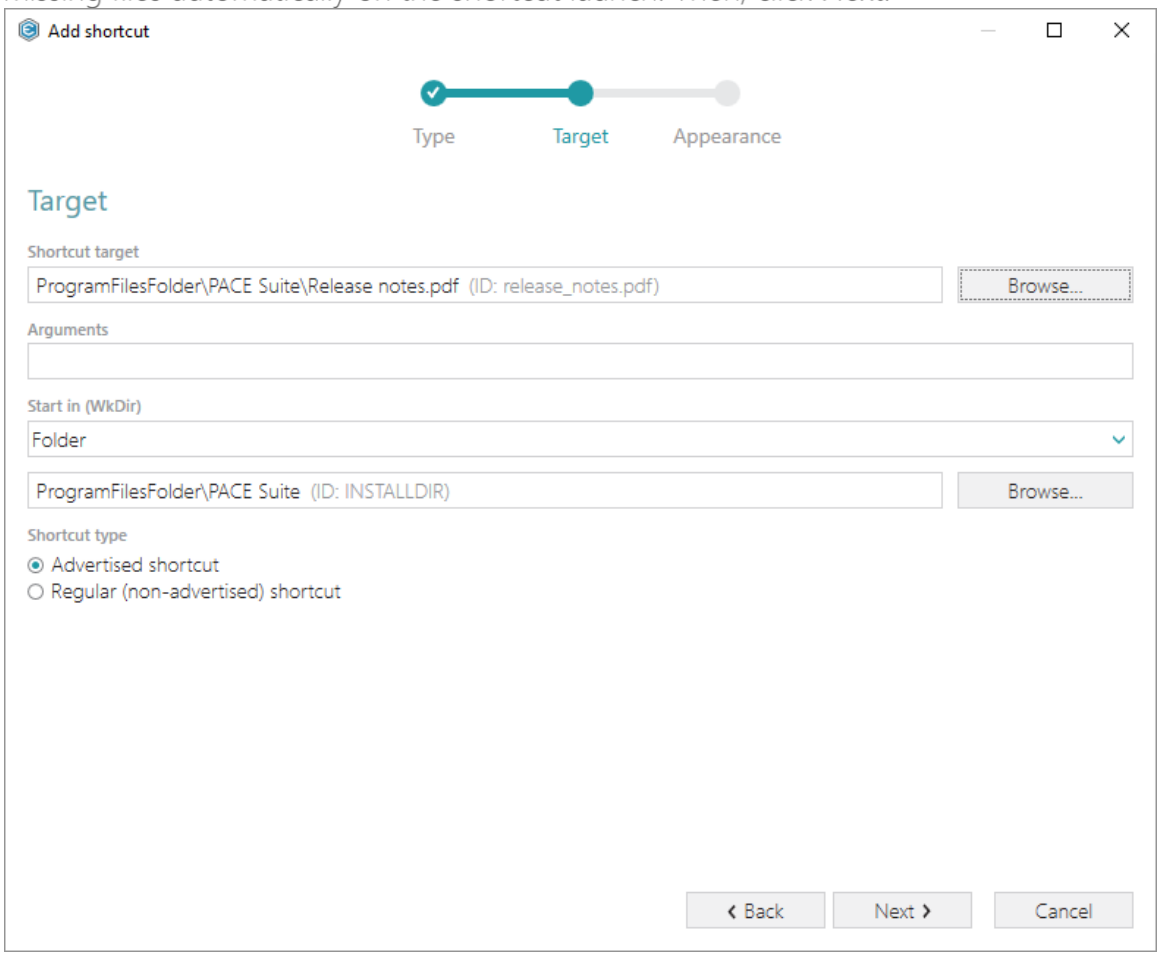
- Click **Browse**, located next to the **Shorcut target** field, to select a target of the shortcut.



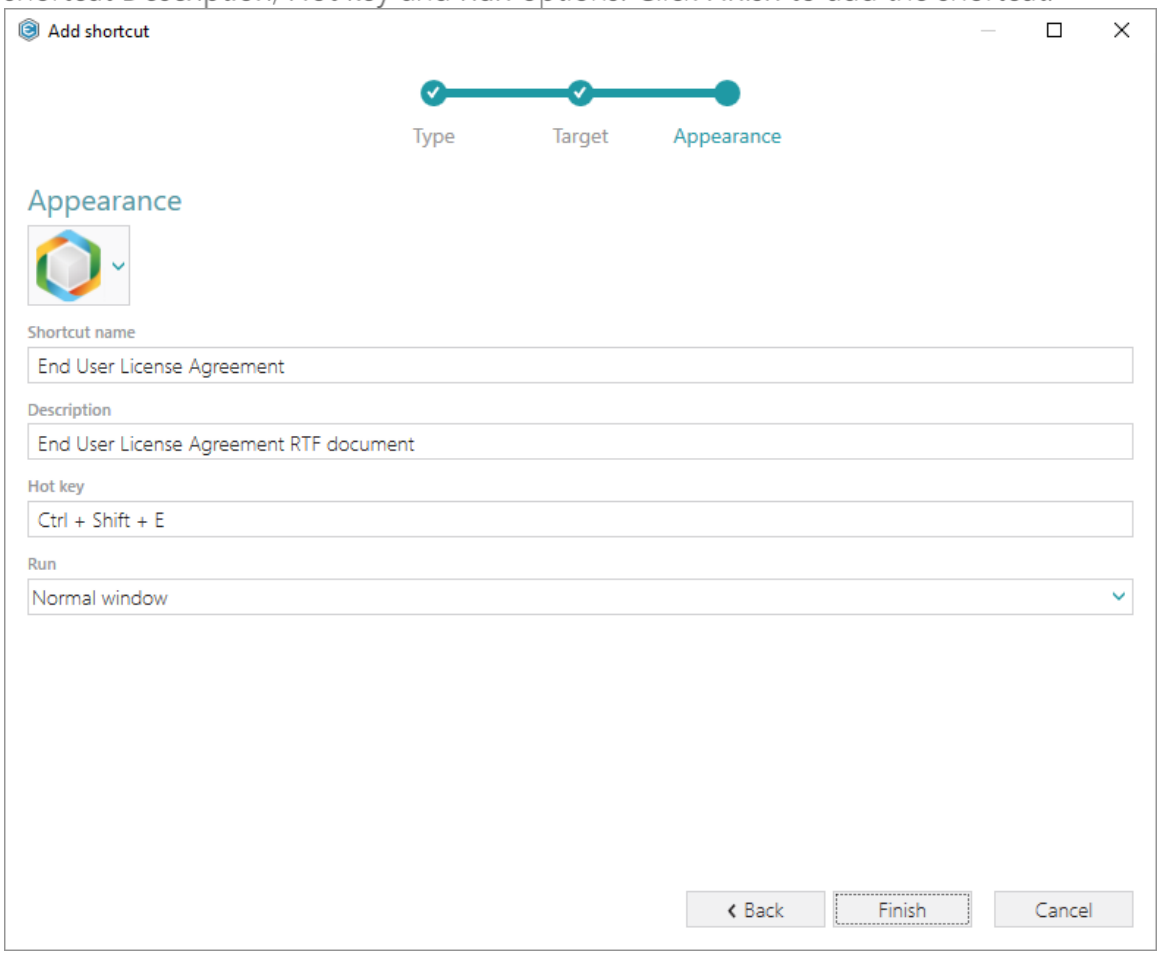
- Choose a file, for which this shortcut will be created, and click Select.



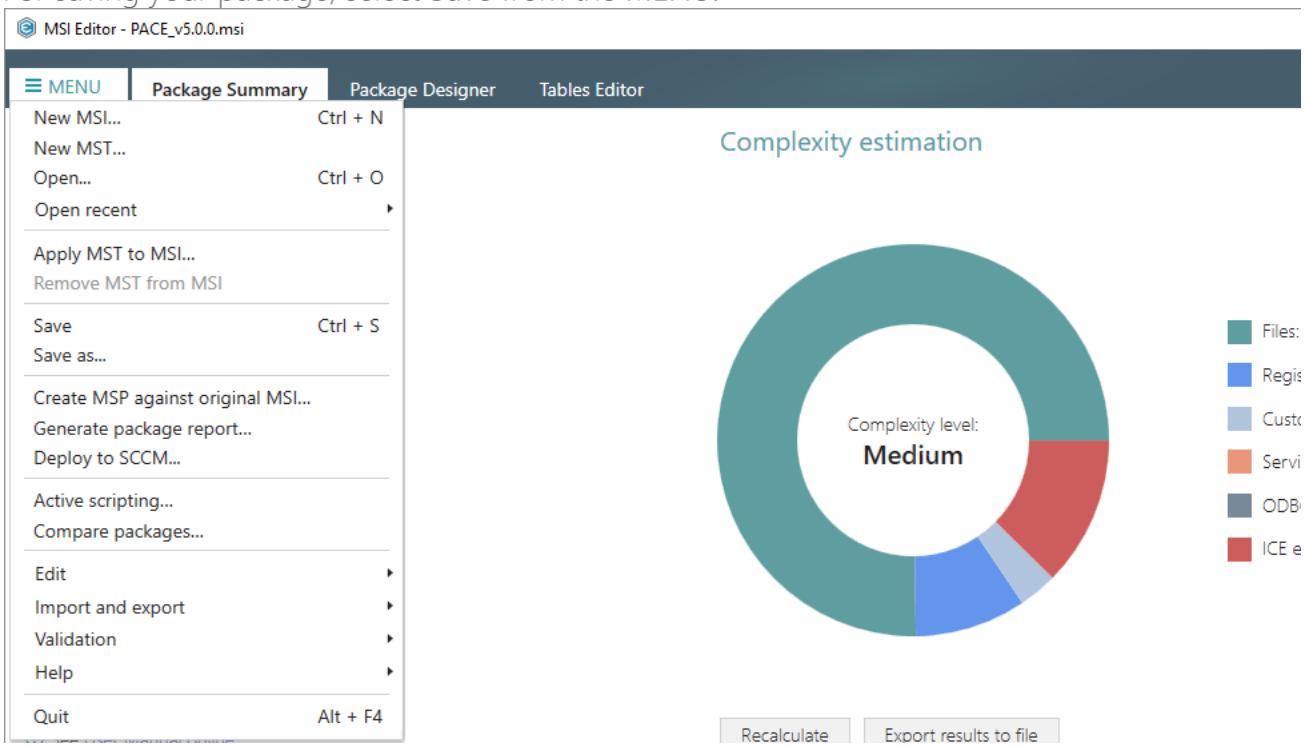
- If needed, specify a shortcut **Arguments**, change the **Start in** path (select a folder or a property), and select a **Shortcut type**.
  NOTE Only Advertised shortcuts support advertisement and repair, which allow to restore the missing files automatically on the shortcut launch. Then, click **Next**.



- On the last step, enter a **Shortcut name**, choose a shortcut icon and, if needed, specify the shortcut **Description**, **Hot key** and **Run** options. Click **Finish** to add the shortcut.



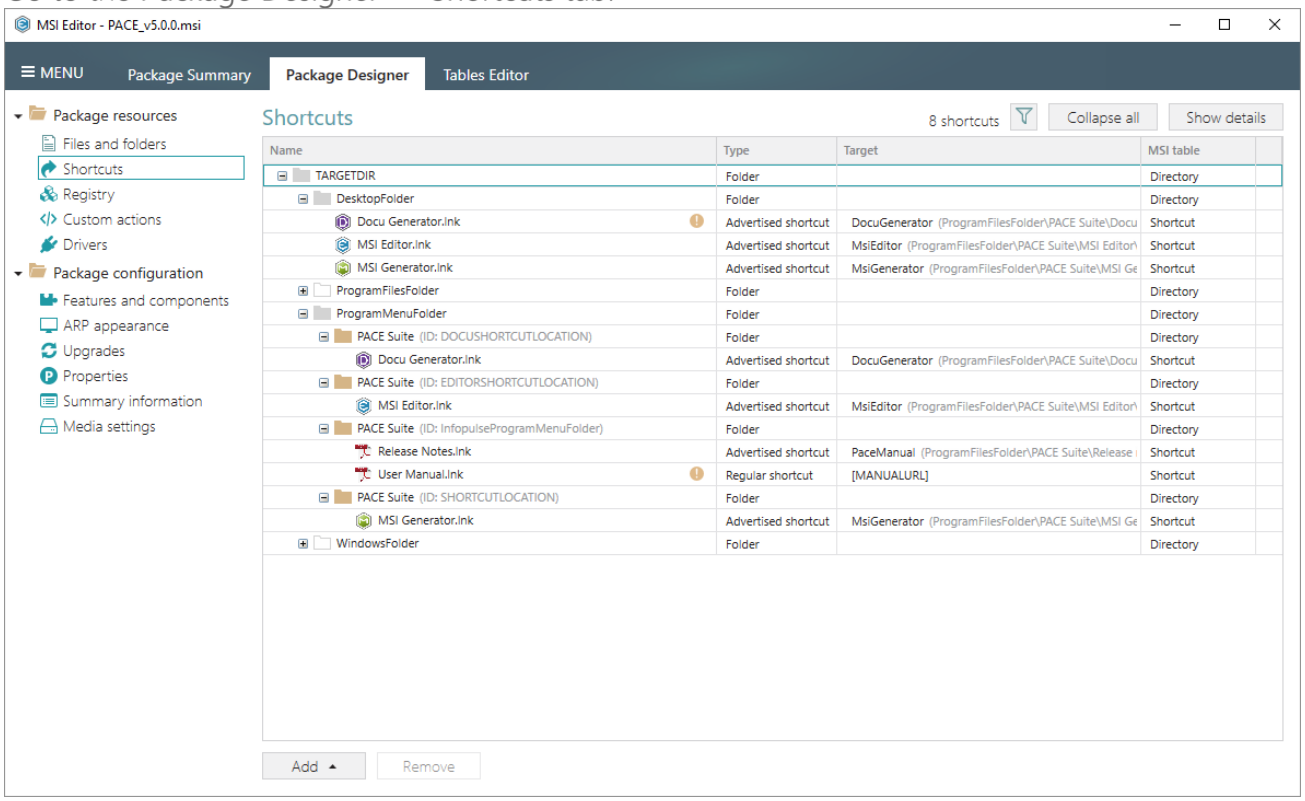
- For saving your package, select **Save** from the **MENU**.



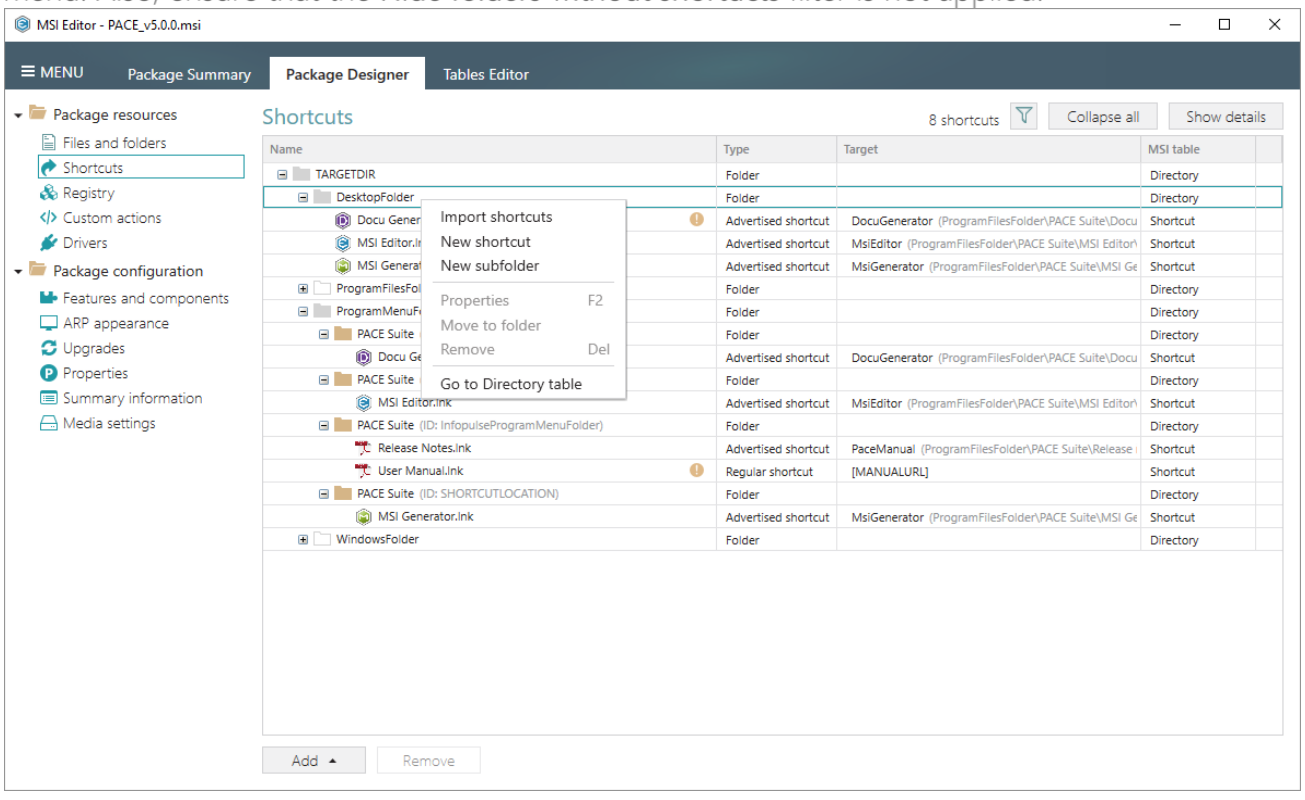
3.7.5.2 Import Shortcuts

Import shortcuts from the file system to the package.

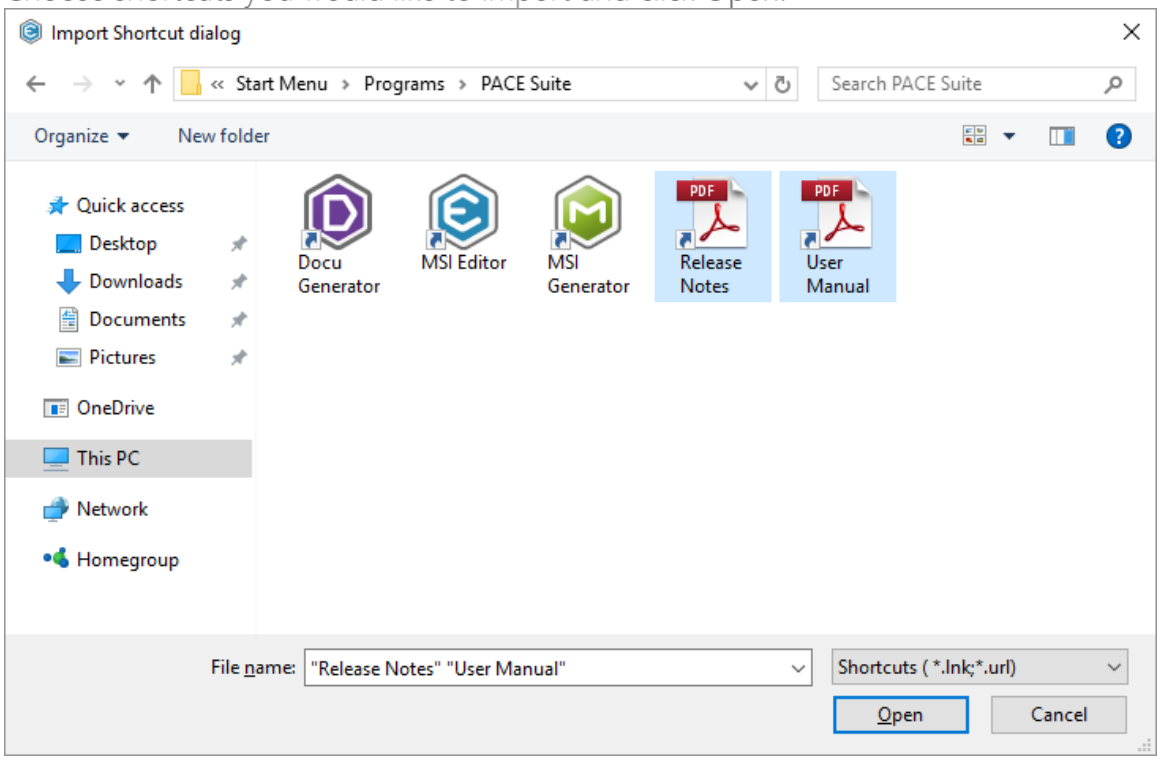
- Go to the Package Designer -> Shortcuts tab.



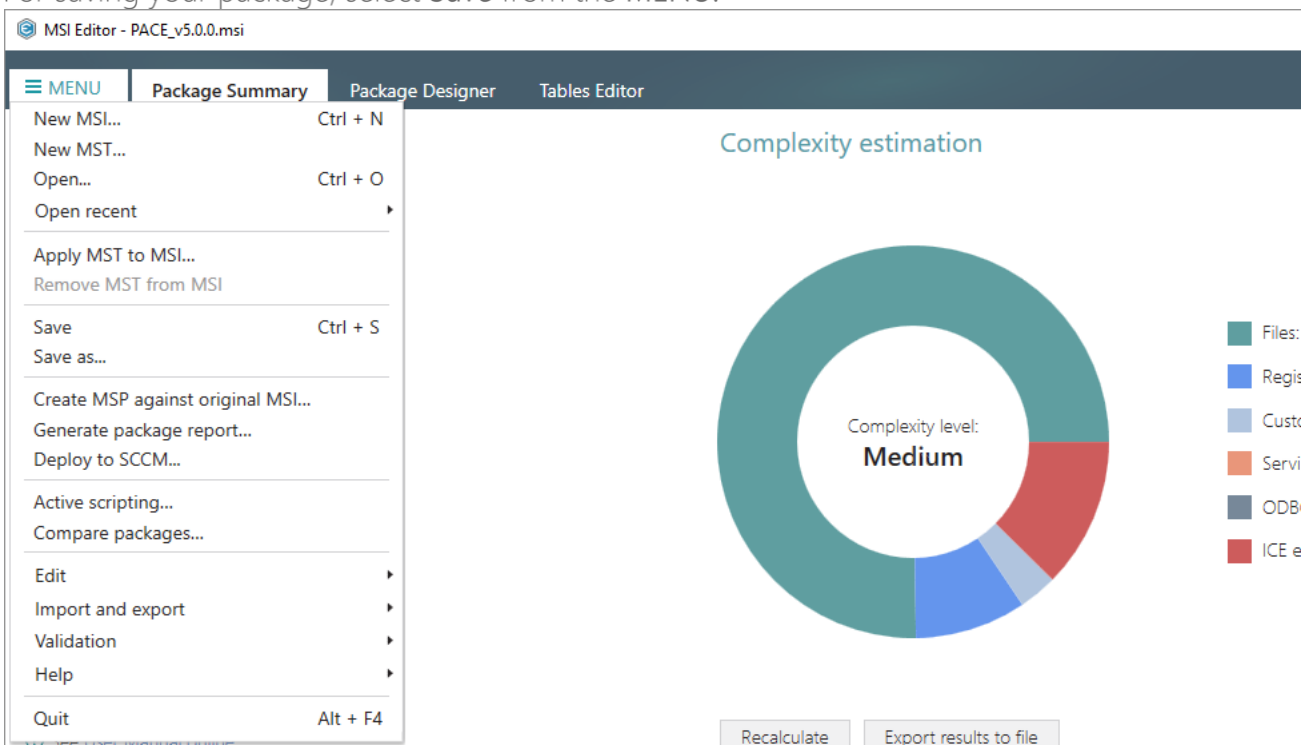
- Select Import shortcuts from the context menu of a folder, to which you want to import shortcuts from the file system.
  NOTE To add the missing system folders (e.g. DesktopFolder, ProgramMenuFolder) or subfolders, select Add -> System folders or Add -> Subfolder respectively from the toolbox menu. Also, ensure that the Hide folders without shortcuts filter is not applied.



- Choose shortcuts you would like to import and click Open.



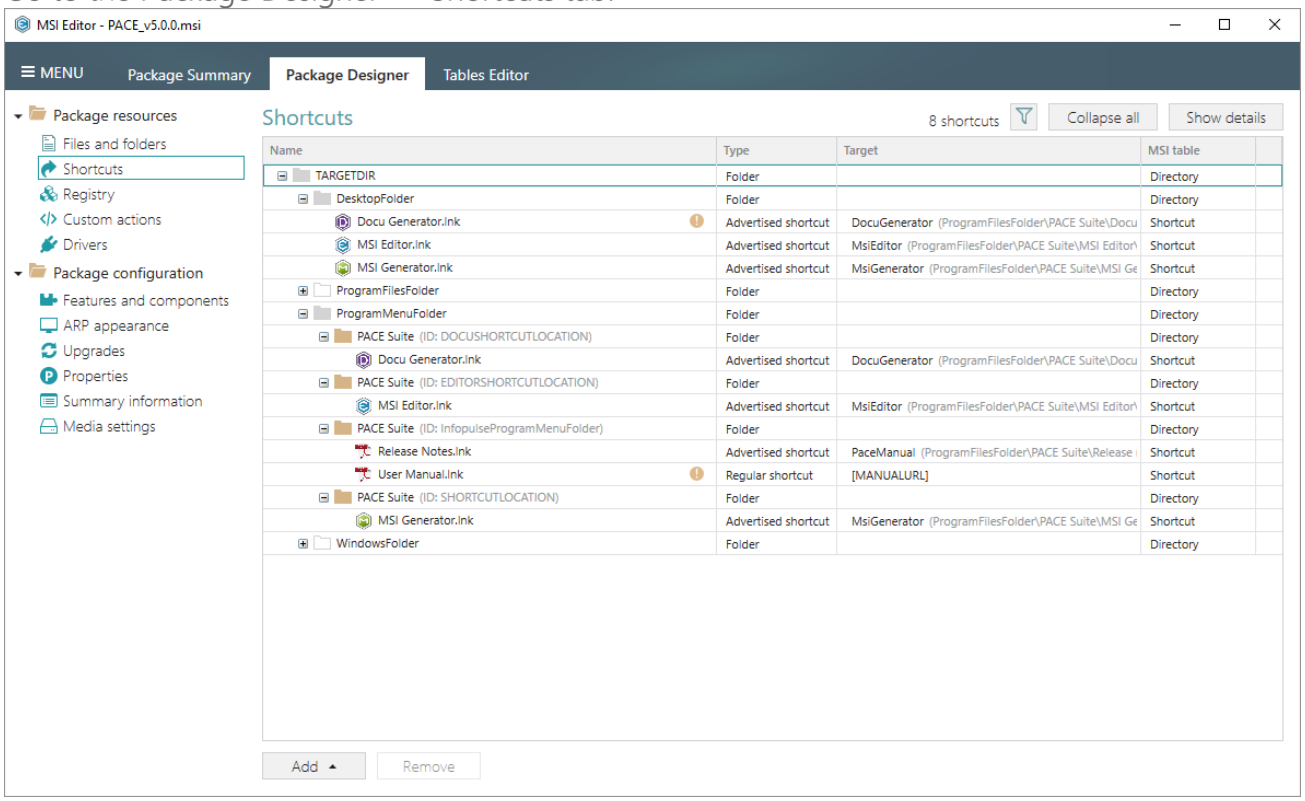
- For saving your package, select Save from the MENU.



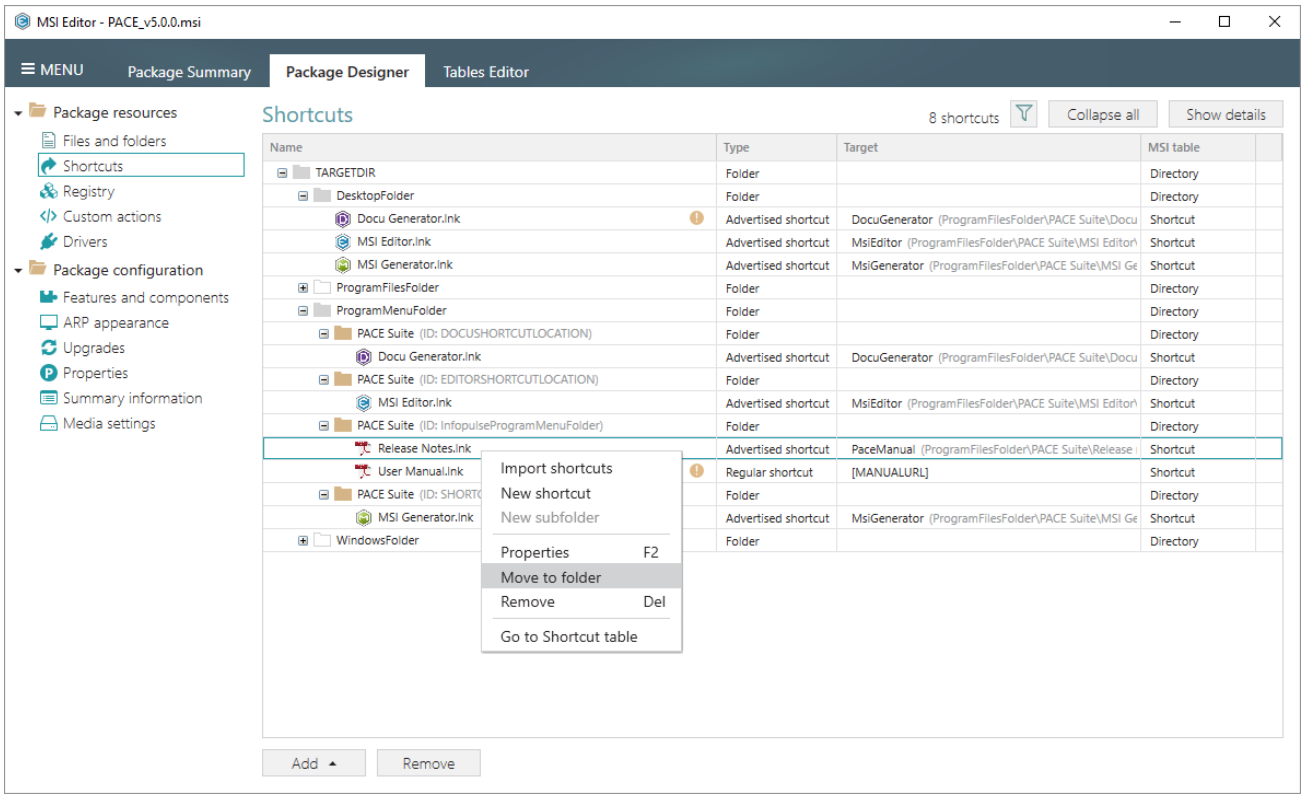
3.7.5.3 Move Shortcut

Move a shortcut to another folder.

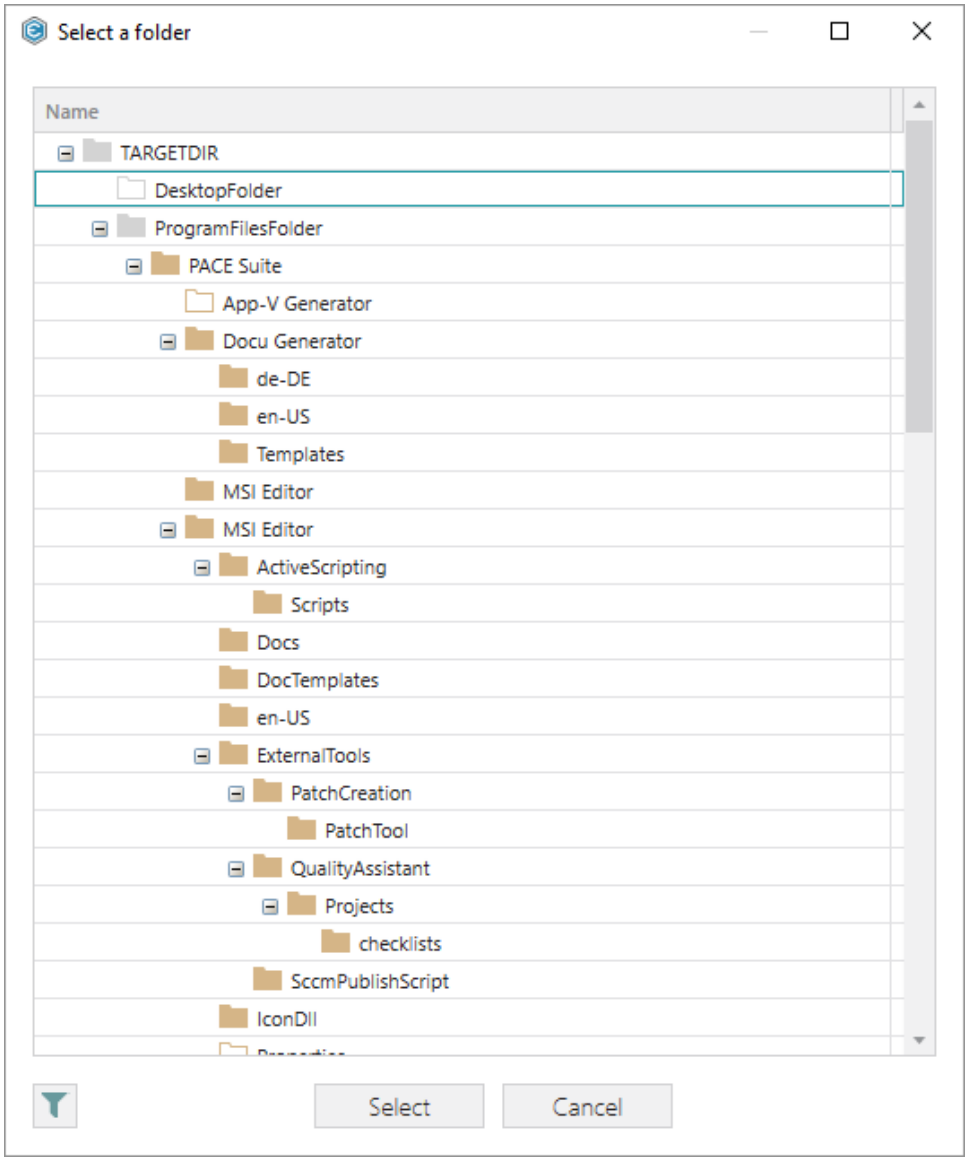
- Go to the **Package Designer** -> **Shortcuts** tab.



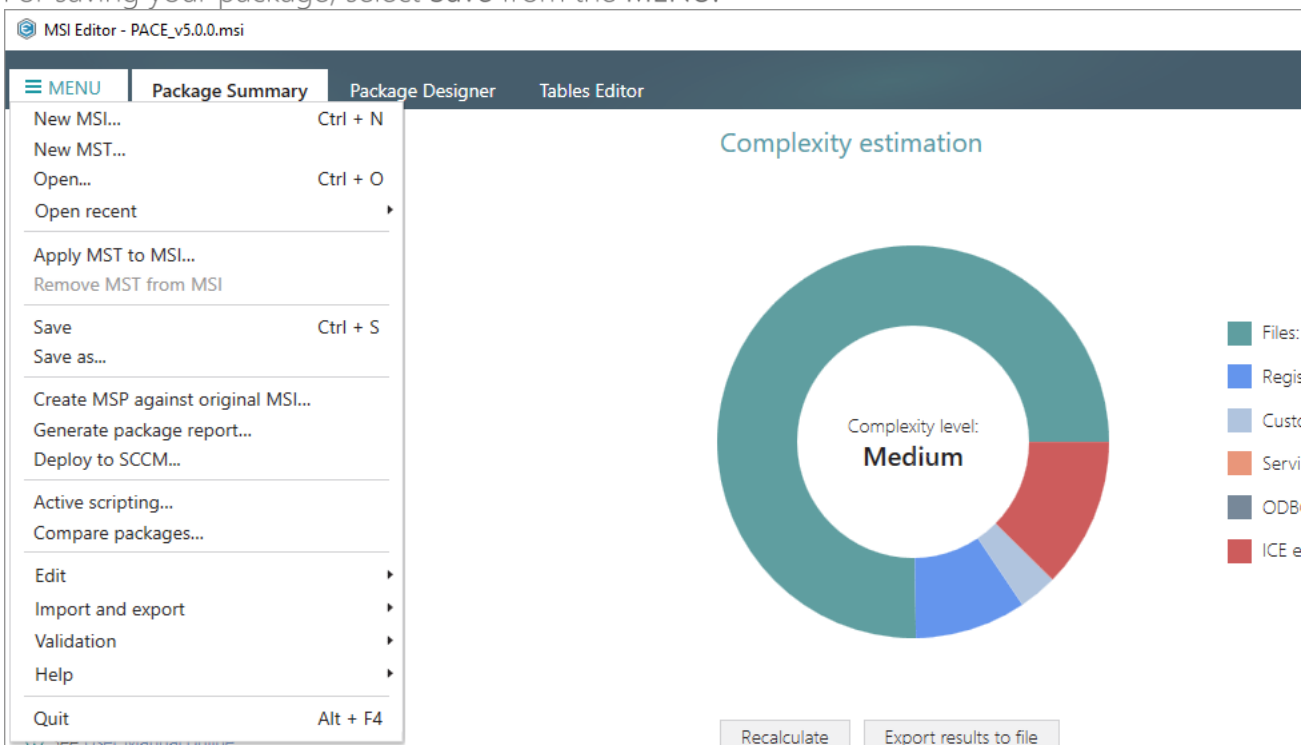
- Select **Move to folder** from the context menu of a shortcut, which you want to move to another folder.



- Select a new destination folder from the tree and click Select.



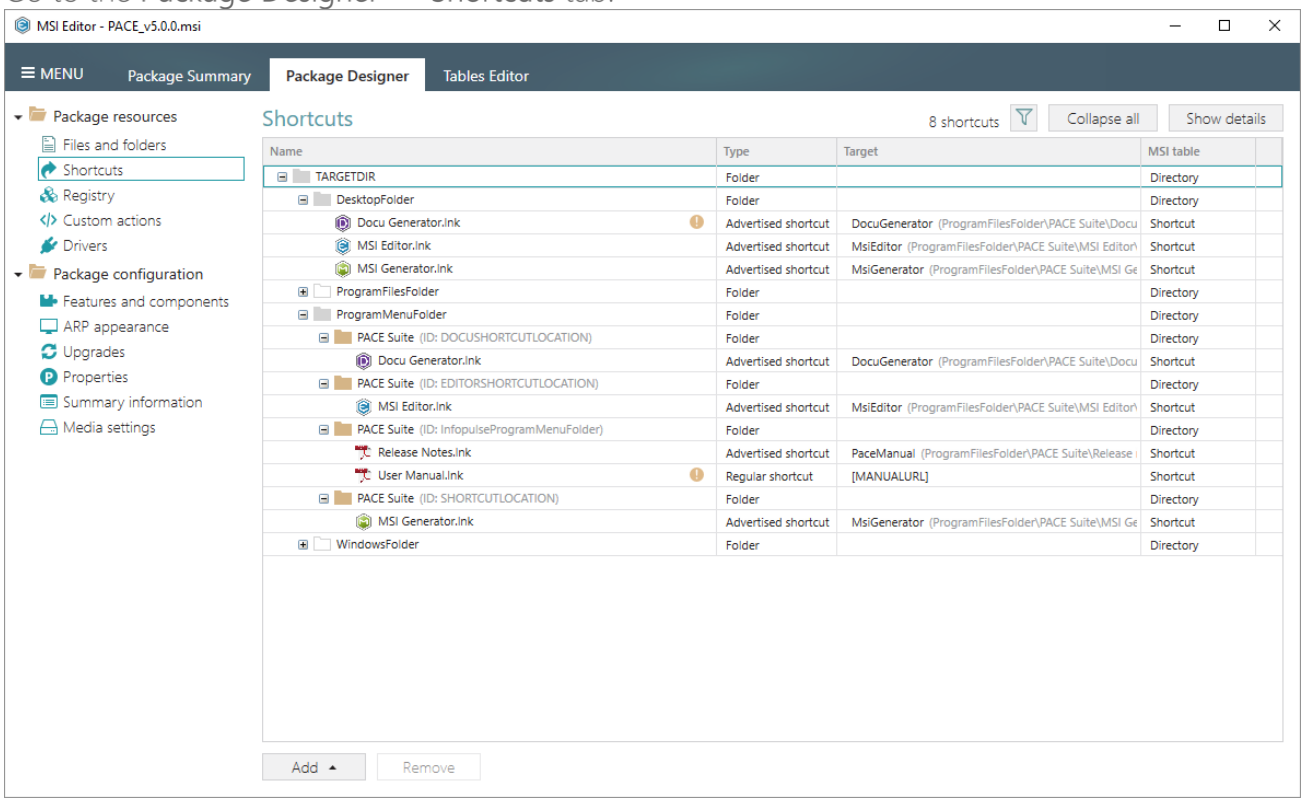
- For saving your package, select Save from the MENU.



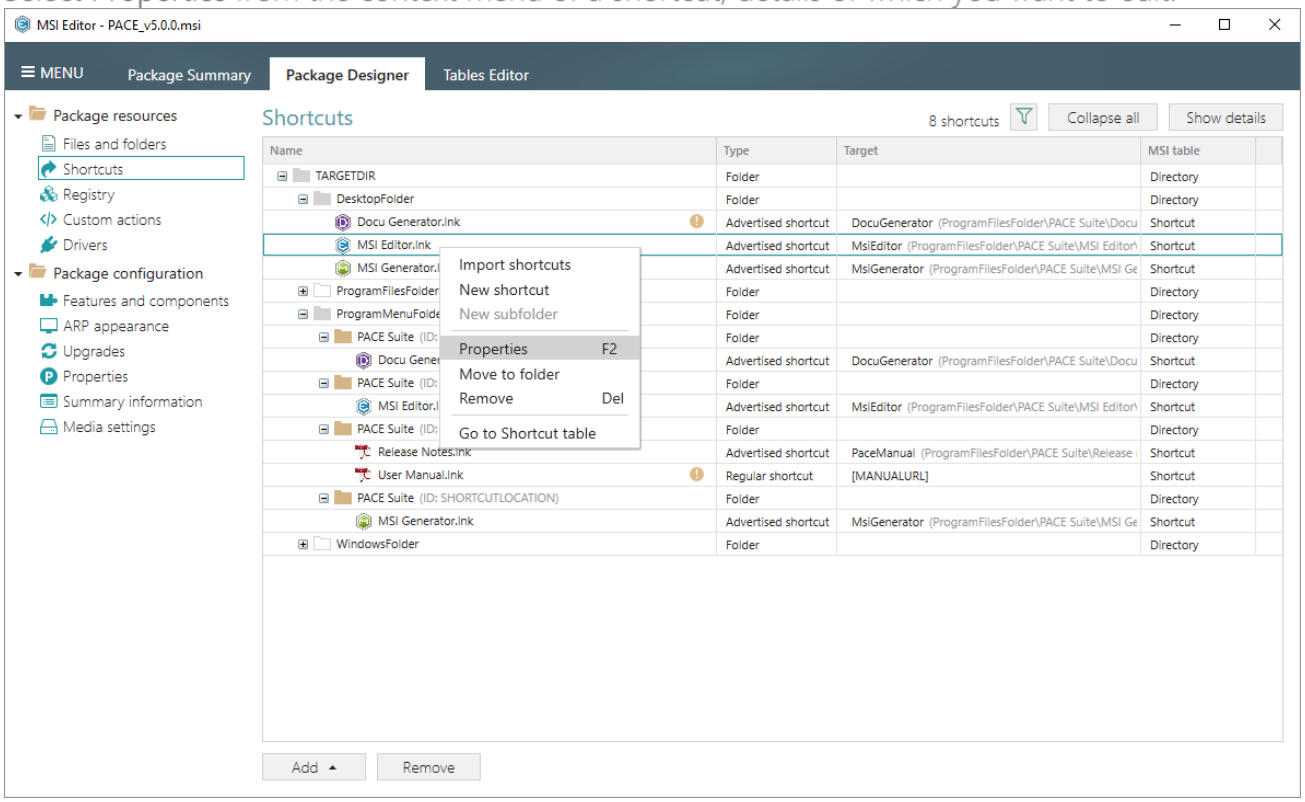
3.7.5.4 Edit Shortcut Properties

Edit shortcut details such as icon, name, installation folder, shortcut type, arguments, description and the like.

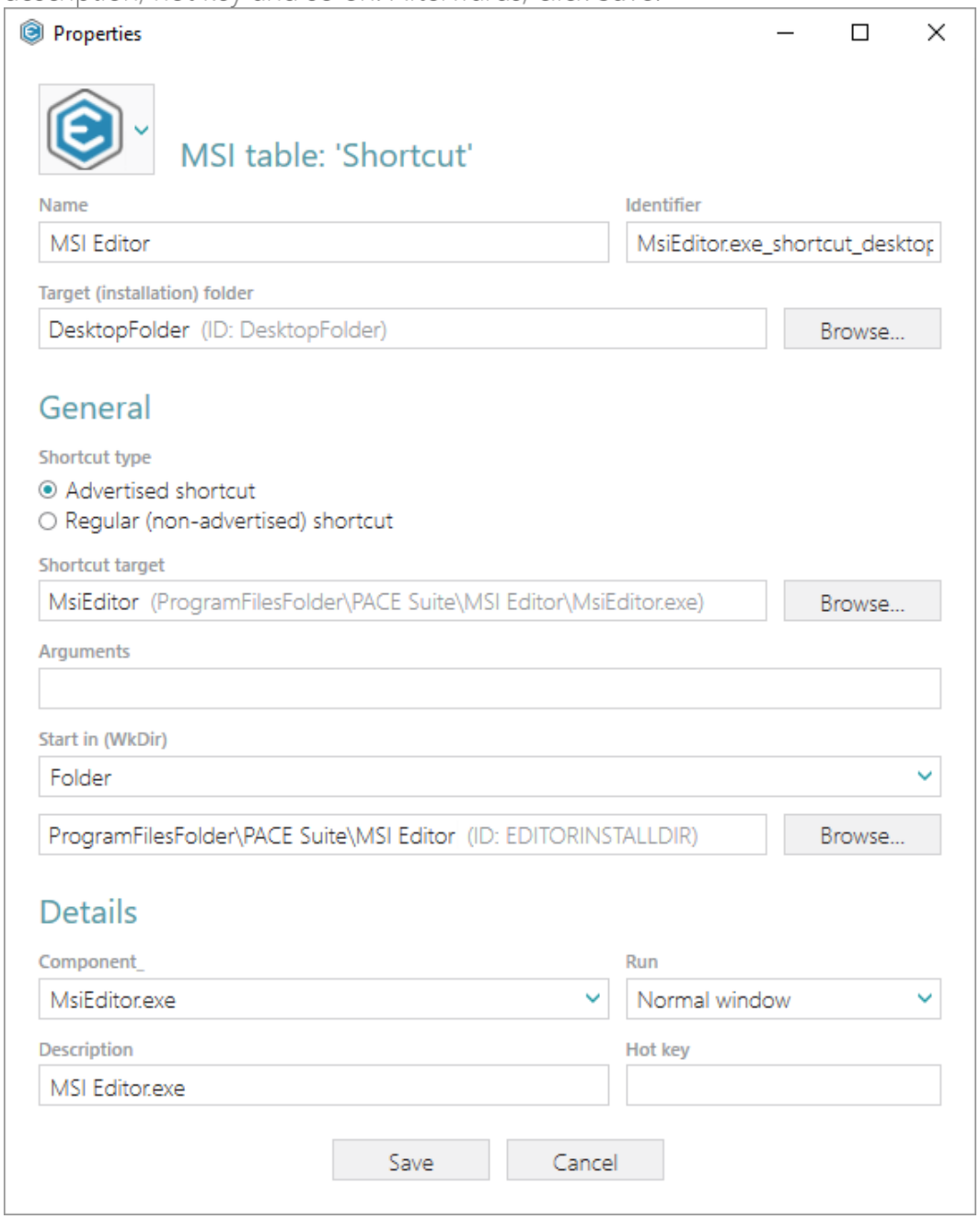
- Go to the Package Designer -> Shortcuts tab.



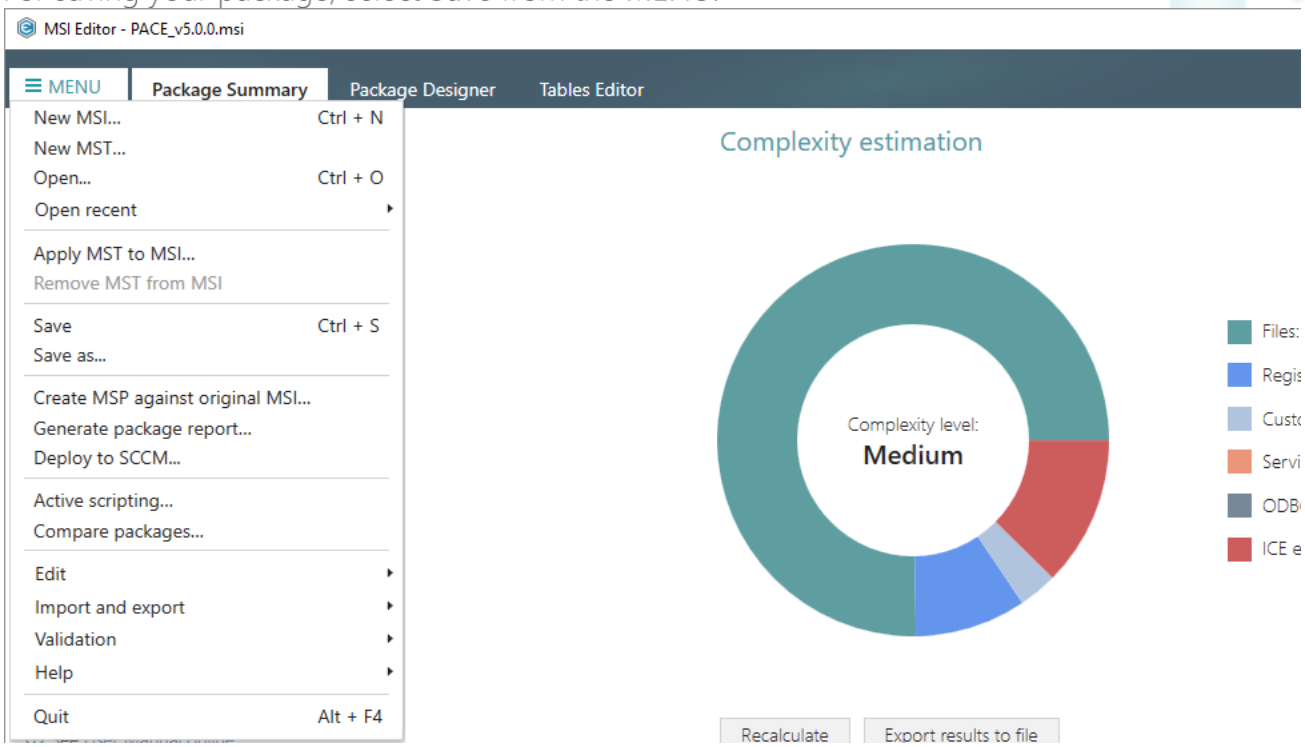
- Select Properties from the context menu of a shortcut, details of which you want to edit.



- Update the necessary values such as icon, shortcut name, installation folder, type, arguments, description, hot key and so on. Afterwards, click Save.



- For saving your package, select Save from the MENU.



3.7.6 Registry

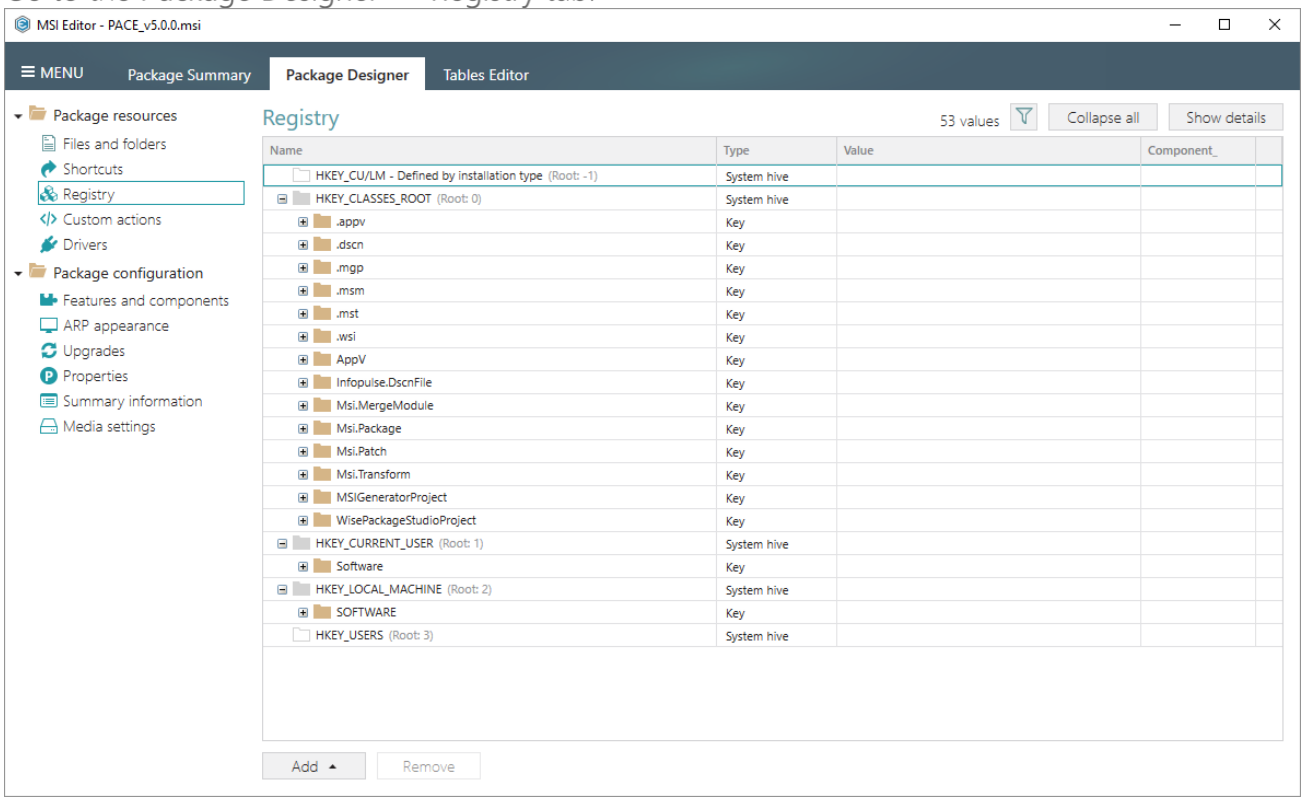
Choose a scenario that better suits your needs:

- Add Registry Key, described in section 3.7.6.1
  Add a new registry key or sub-key manually.
- Add Registry Value, described in section 3.7.6.2
  Add a new registry value manually.
- Add ActiveSetup, described in section 3.7.6.2.1
  Intergrate ActiveSetup into your package to propagate the per-user resources for every logged-in user of the system.
- Add Value/Key Removal, described in section 3.7.6.3
  Add the RemoveRegistry to force a registry value or key removal on install.
- Import Registry, described in section 3.7.6.4
  Import registry from the REG file.
- Export Registry, described in section 3.7.6.5
  Export the selected registry to the REG file.
- Set Permissions, described in section 3.7.6.6
  Set custom permissions for the registry key.
- Remove Permissions, described in section 3.7.6.7
  Locate registry key with custom permissions and remove the unnecessary permissions.
- Move Value, Key, described in section 3.7.6.8
  Move a registry value or a key from one parent key to another one.
- Edit Registry Key, described in section 3.7.6.9
  Modify details and options of the selected registry key.
- Edit Registry Value, described in section 3.7.6.10
  Modify details of the selected registry value.
- Edit Value/Key Removal, described in section 3.7.6.11
  Modify details of the selected value/key removal item (RemoveRegistry) to secure a value or a key removal during the package installation.

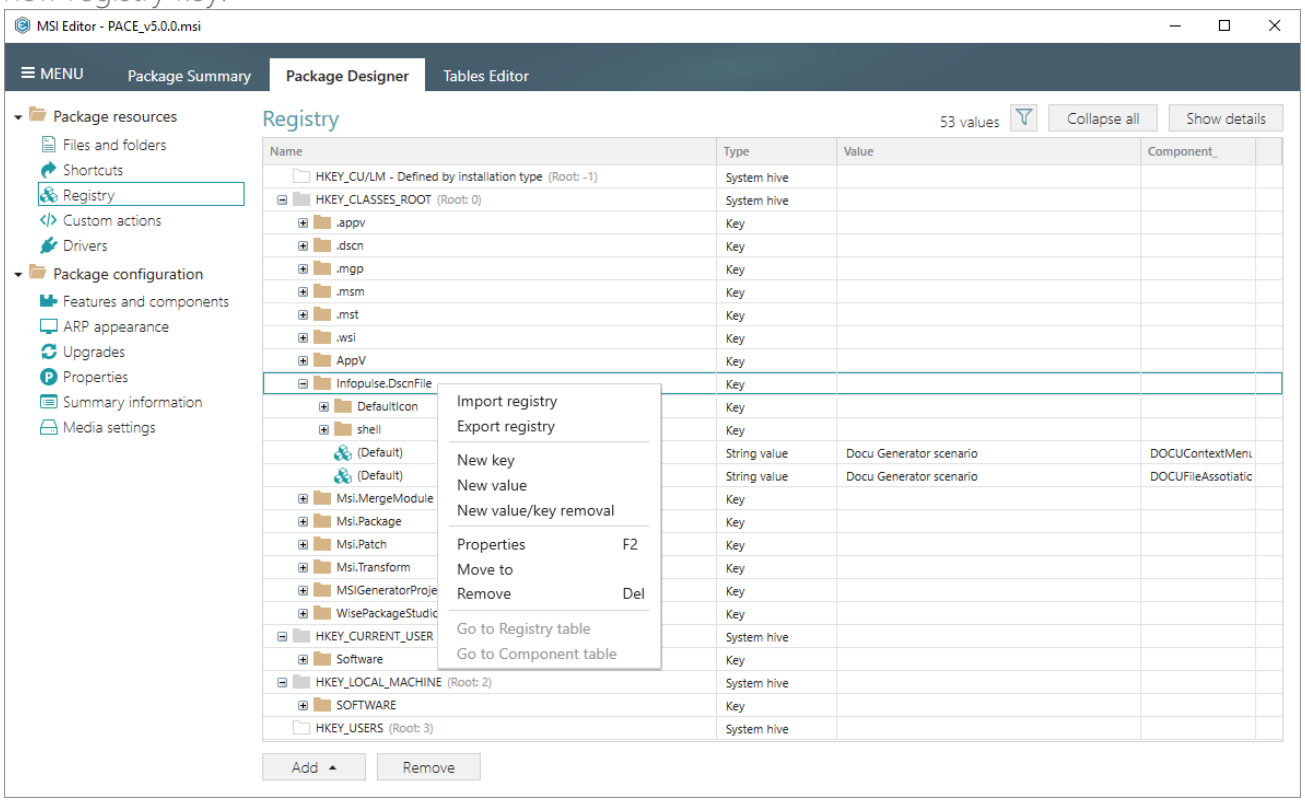
3.7.6.1 Add Registry Key

Add a new registry key or sub-key manually.

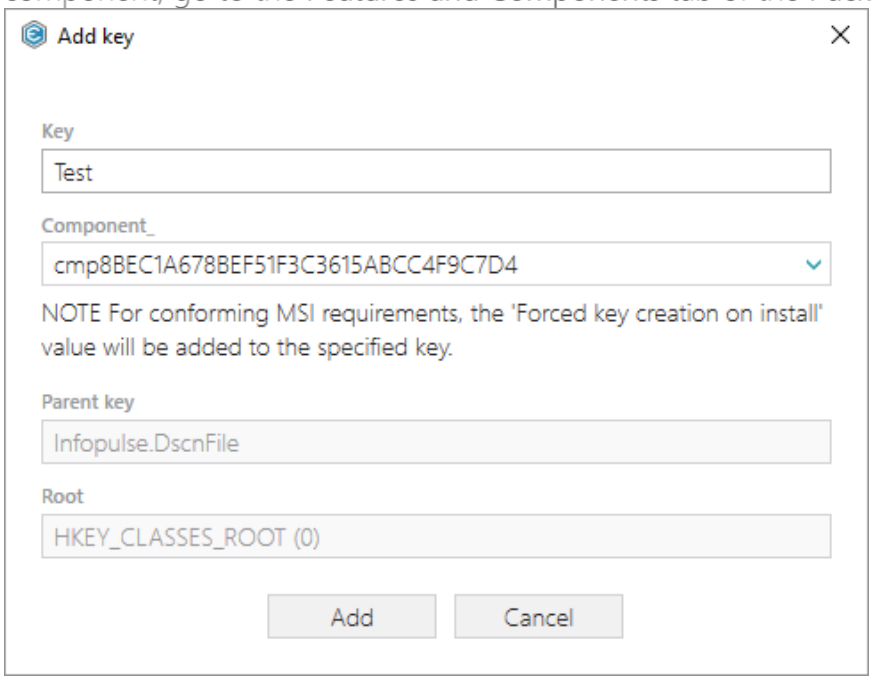
- Go to the Package Designer -> Registry tab.



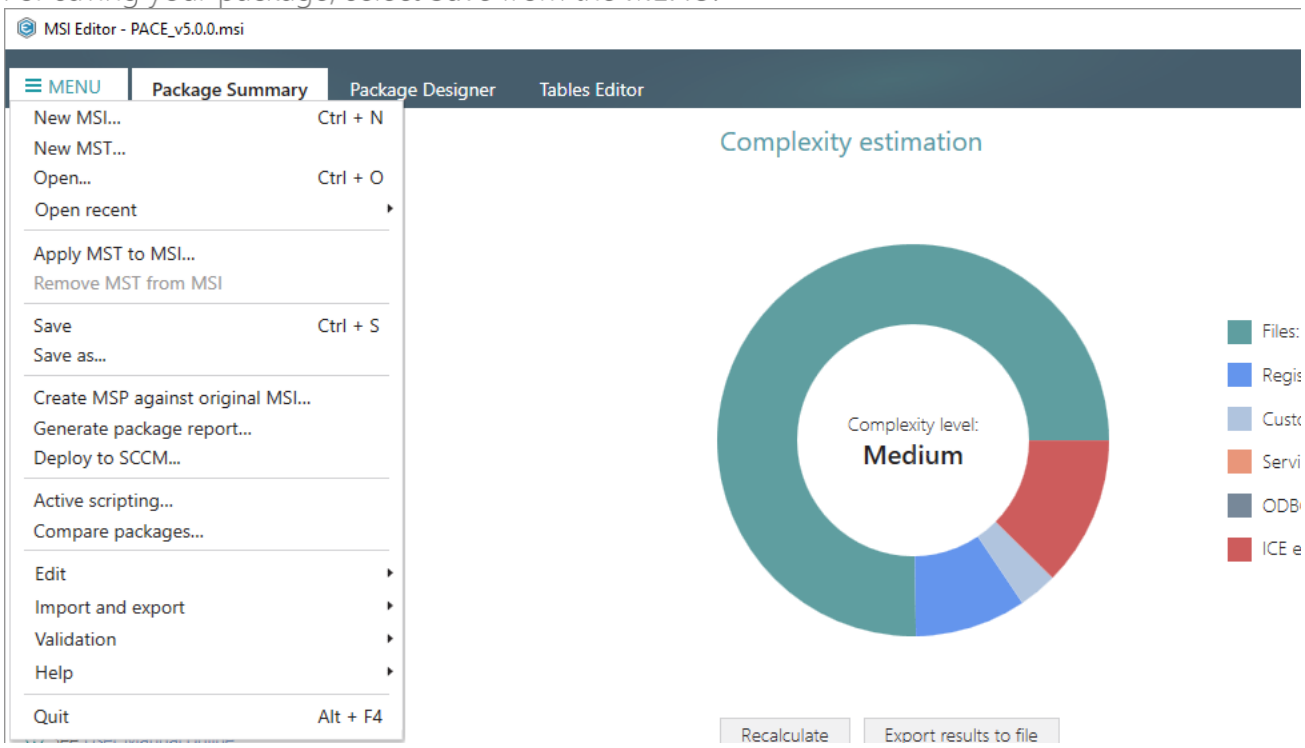
- Select **New key** from the context menu of a root hive or a key, where you want to create a new registry key.



- Enter a key name, select a component from the list, then click **Add**. For creating a new component, go to the **Features and Components** tab of the **Package Designer**.



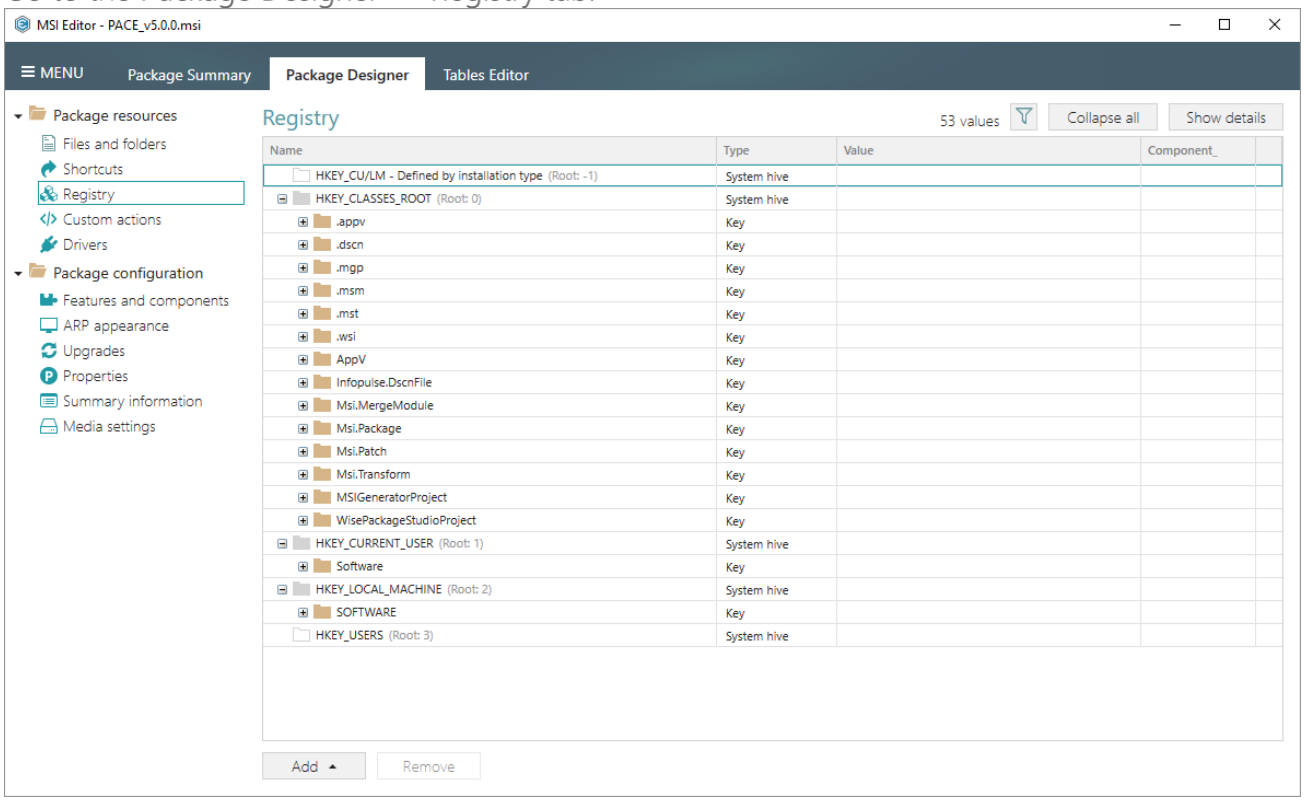
- For saving your package, select **Save** from the **MENU**.



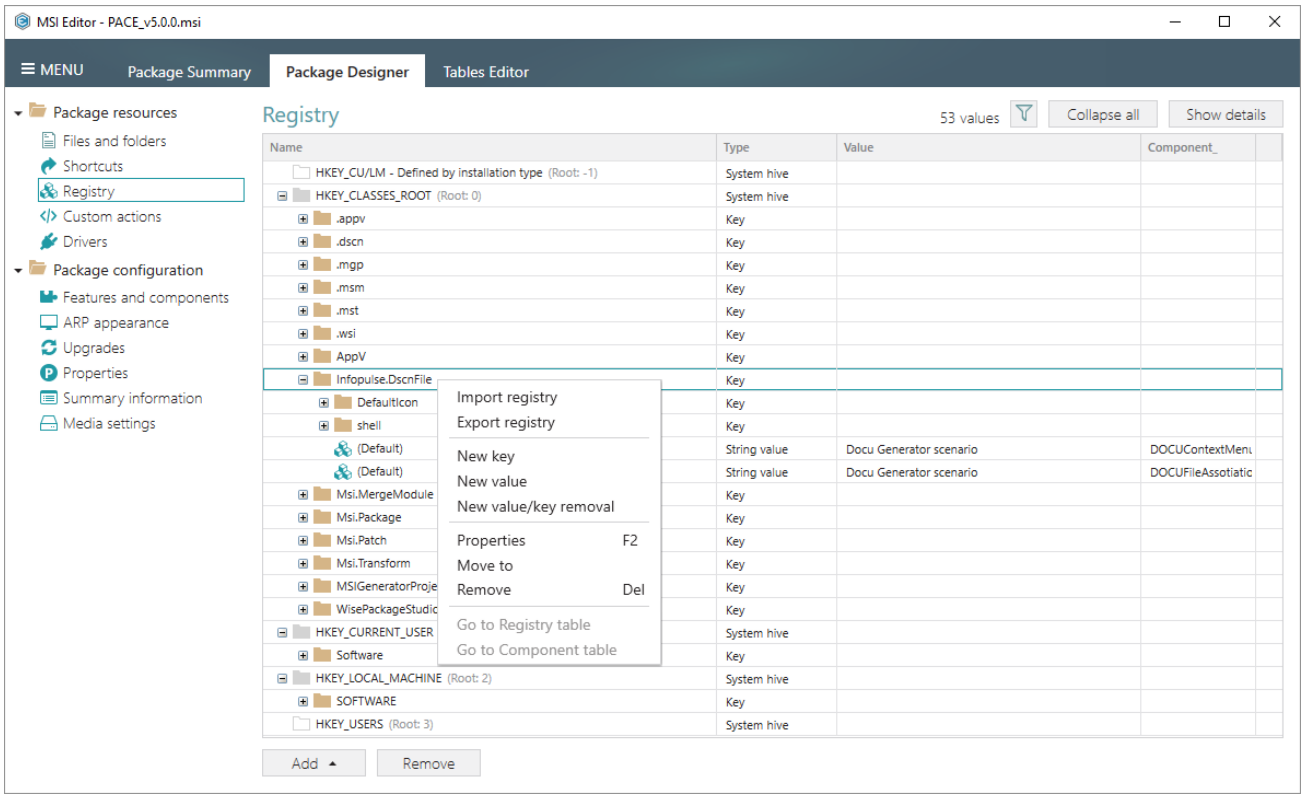
3.7.6.2 Add Registry Value

Add a new registry value manually.

- Go to the **Package Designer** -> **Registry** tab.



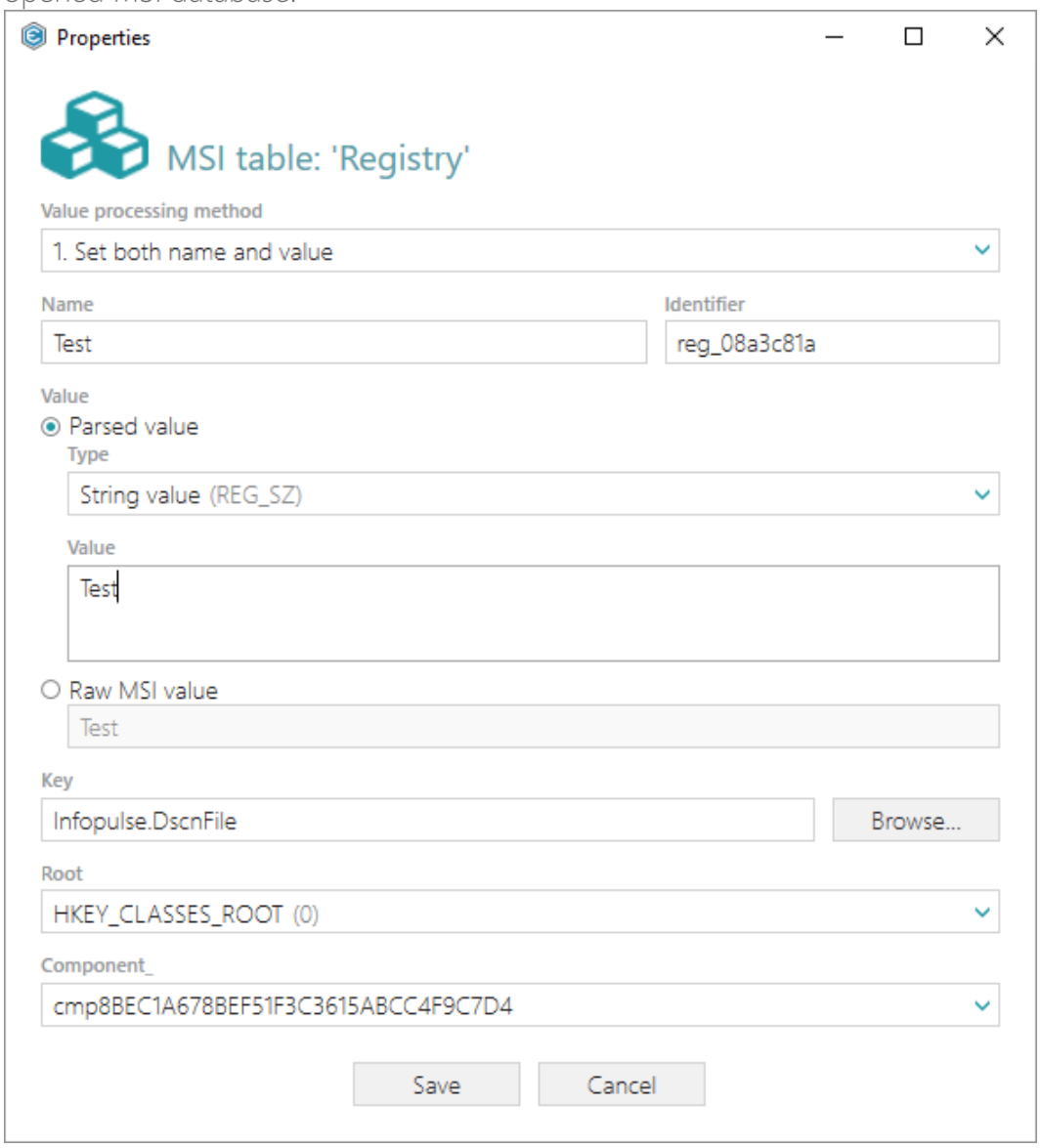
- Select **New value** from the context menu of a key, where you want to create a new registry value.



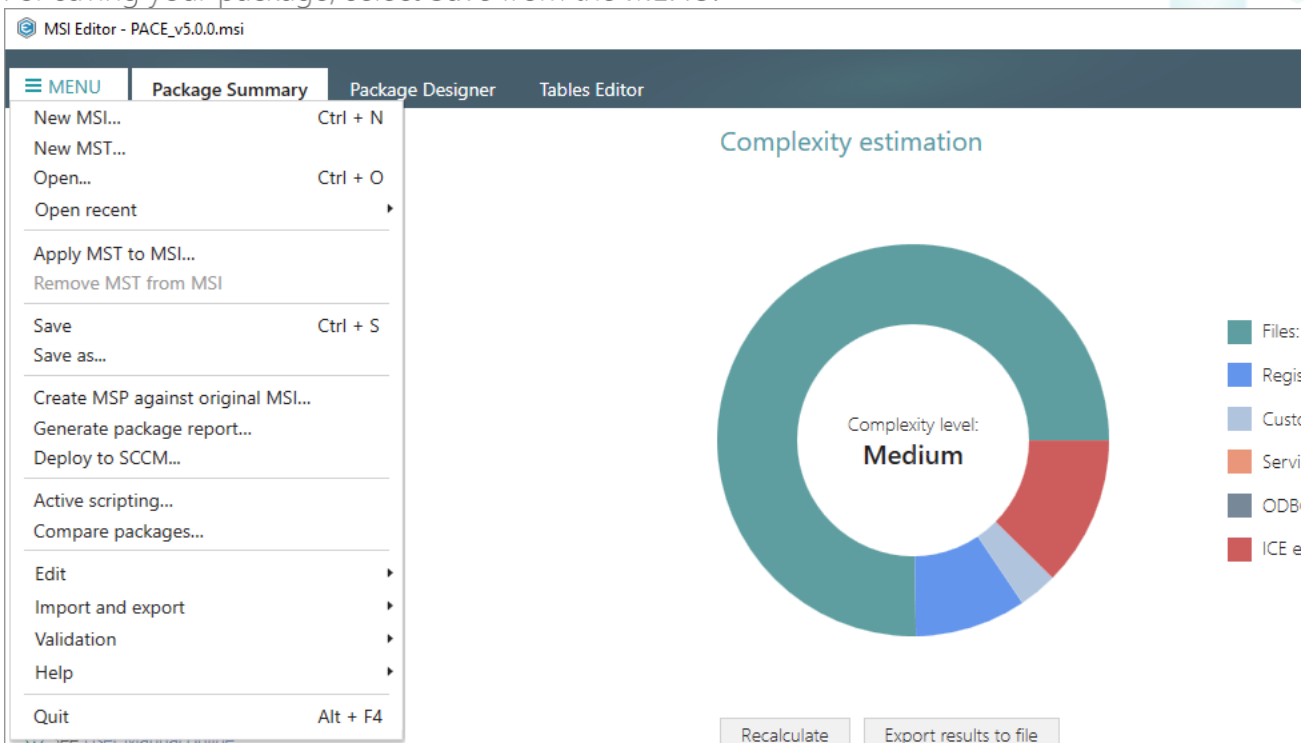
- For creating an ordinary registry value, just specify a **Name**, **Value** itself, select a **Component** from the list, to which this value will be assigned, and then click **Save**.

  For creating a Default key value, or securing an empty key creation on install and/or its removal on uninstall, select a respective option from the **Value processing method** field.

  The **Parsed value** option serves easier creation of a registry value of the proper value type. However, the **Raw MSI value** option gives ability to manage what exactly will be saved to the opened MSI database.



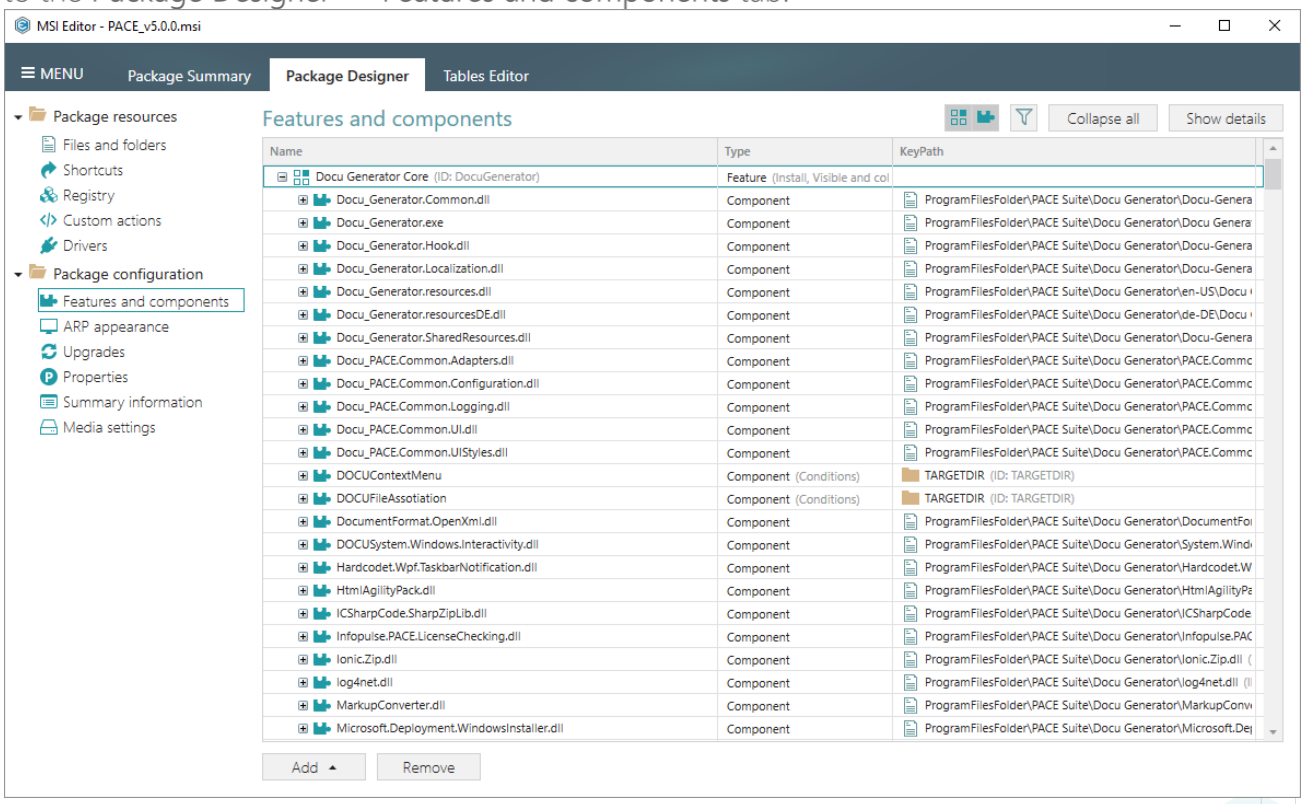
- For saving your package, select Save from the MENU.



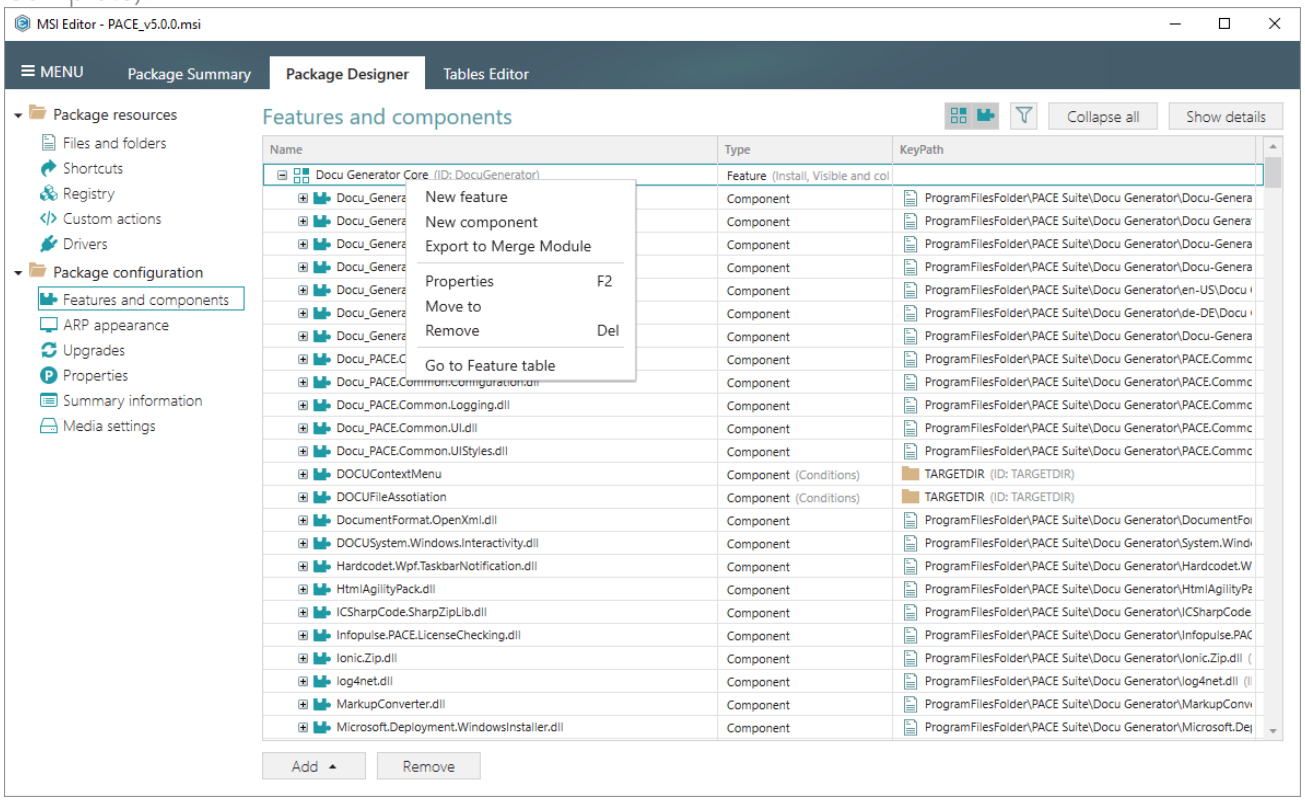
3.7.6.2.1 Add ActiveSetup

Intergrate ActiveSetup into your package to propagate the per-user resources for every logged-in user of the system. The ActiveSetup is represented as the registry value that triggers the repair function of the Windows Installer during the next user log-in in order to restore the missing resources.

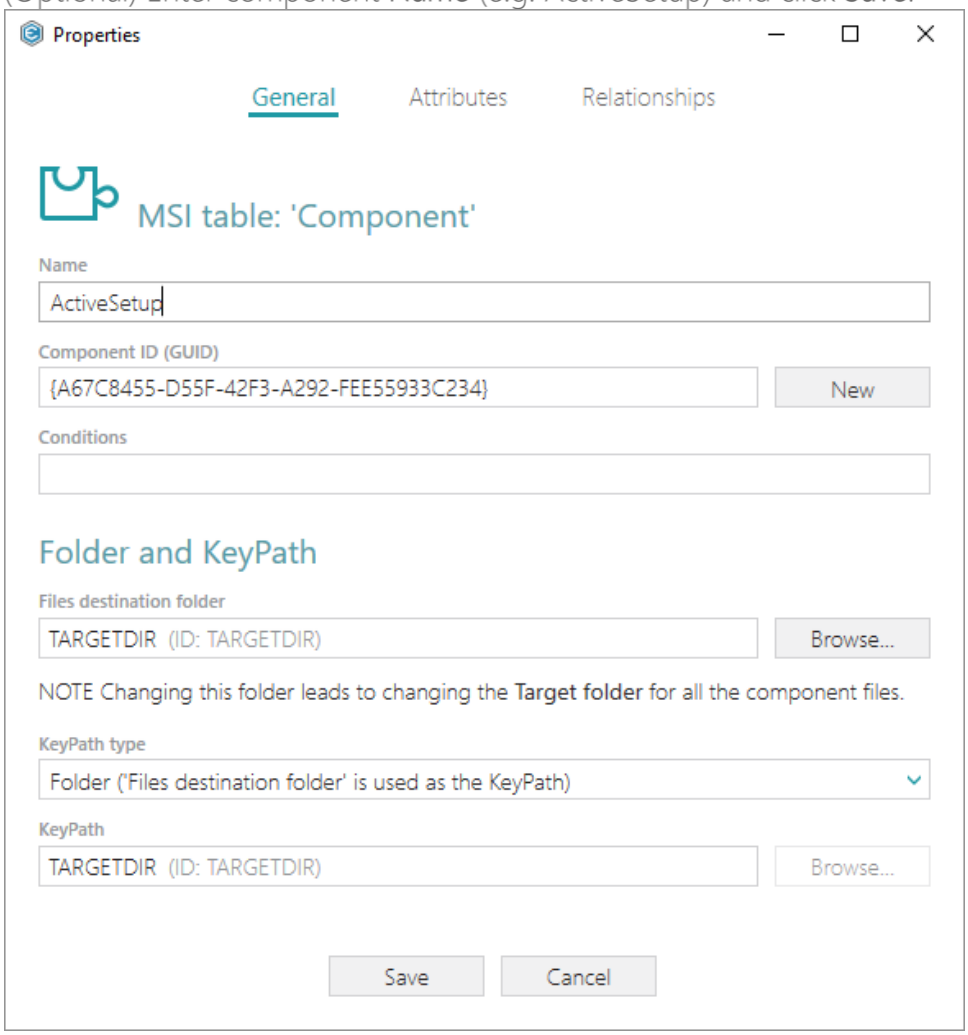
- (Optional) First, create a new component for the ActiveSetup registry key. Alternatively, you can use the existing component with the per-machine resources. To create a component, go to the Package Designer -> Features and components tab.



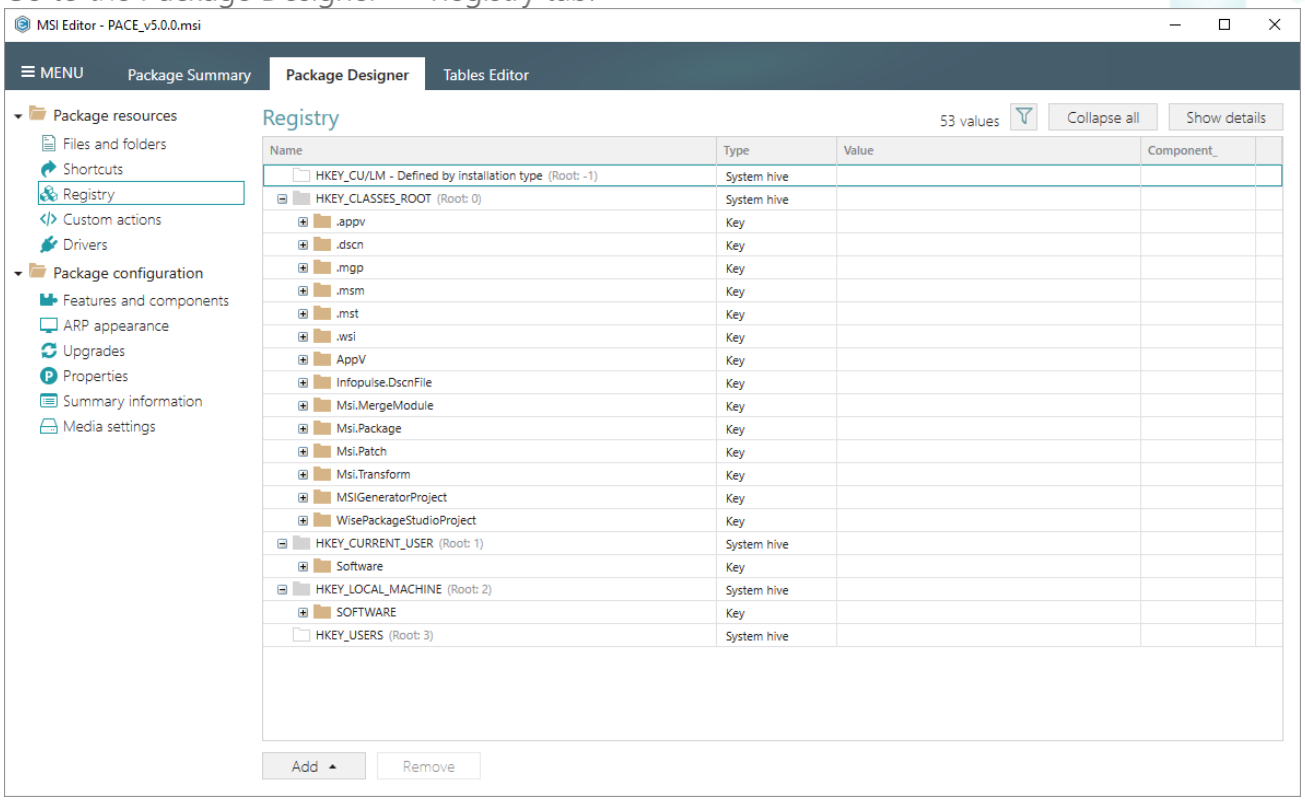
- (Optional) Select **New component** from the context menu of any feature. It is recommended to use feature that always will be installed and contains per-machine resources (e.g. Complete).



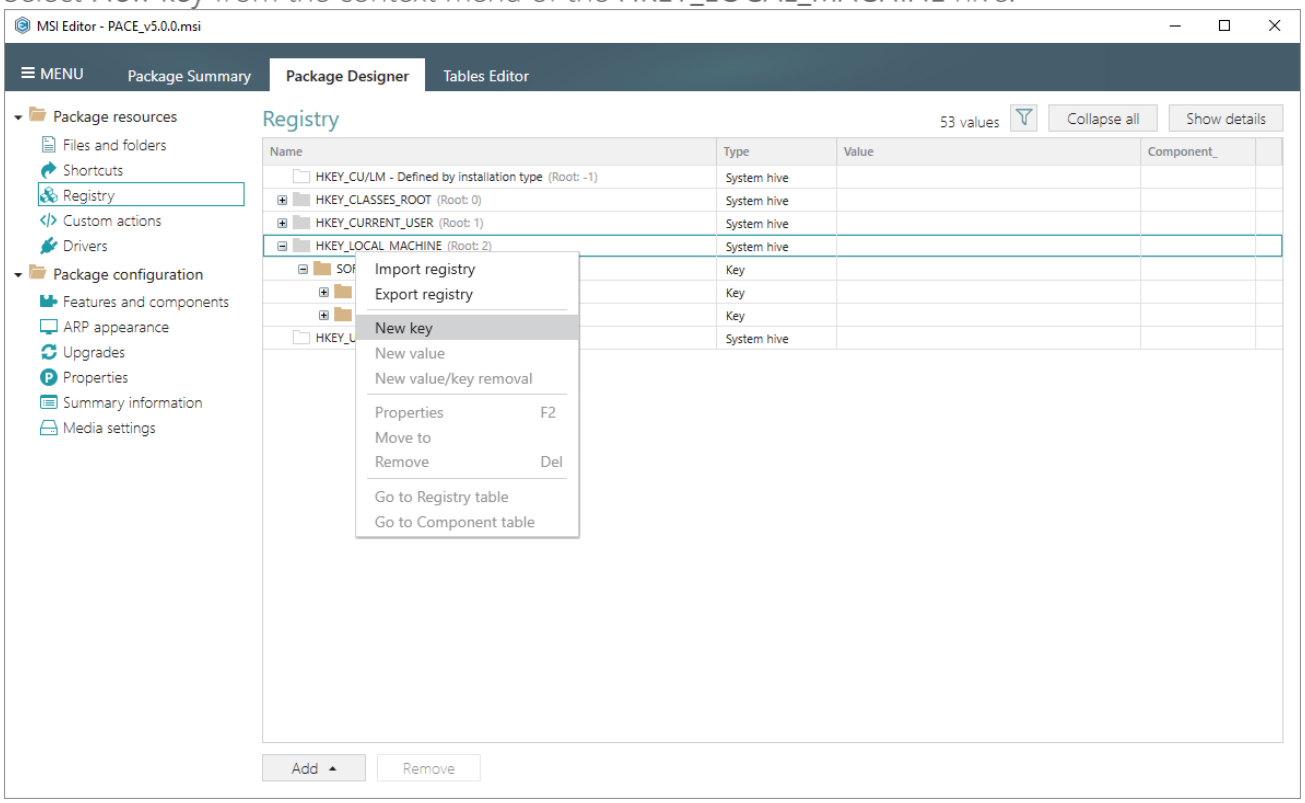
- (Optional) Enter component **Name** (e.g. ActiveSetup) and click **Save**.



- Go to the Package Designer -> Registry tab.

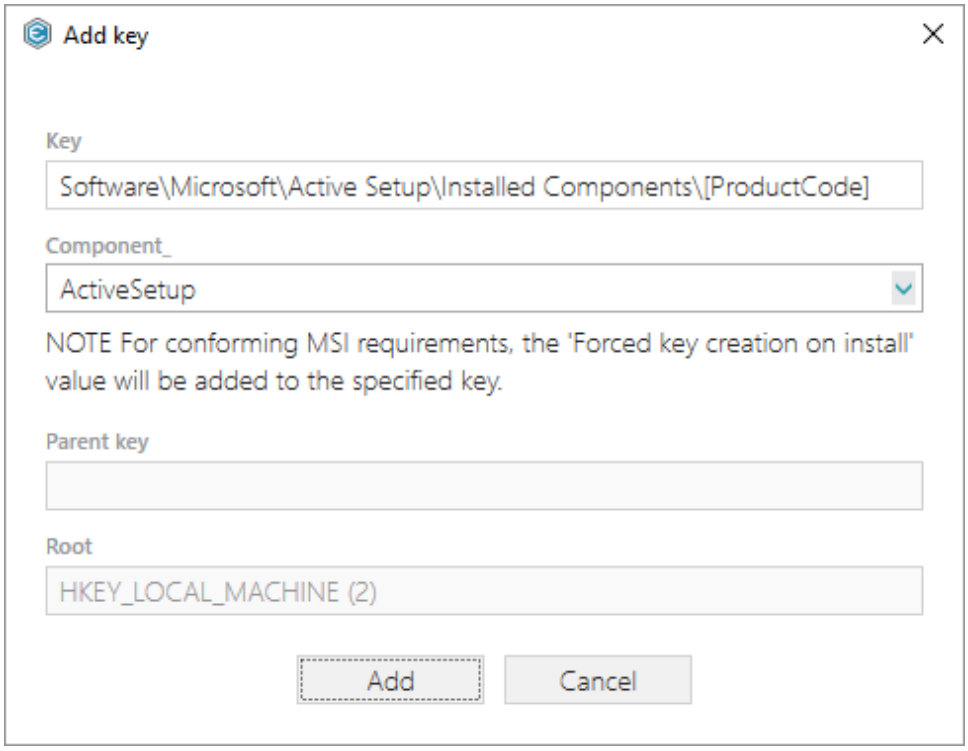


- Select New key from the context menu of the HKEY_LOCAL_MACHINE hive.

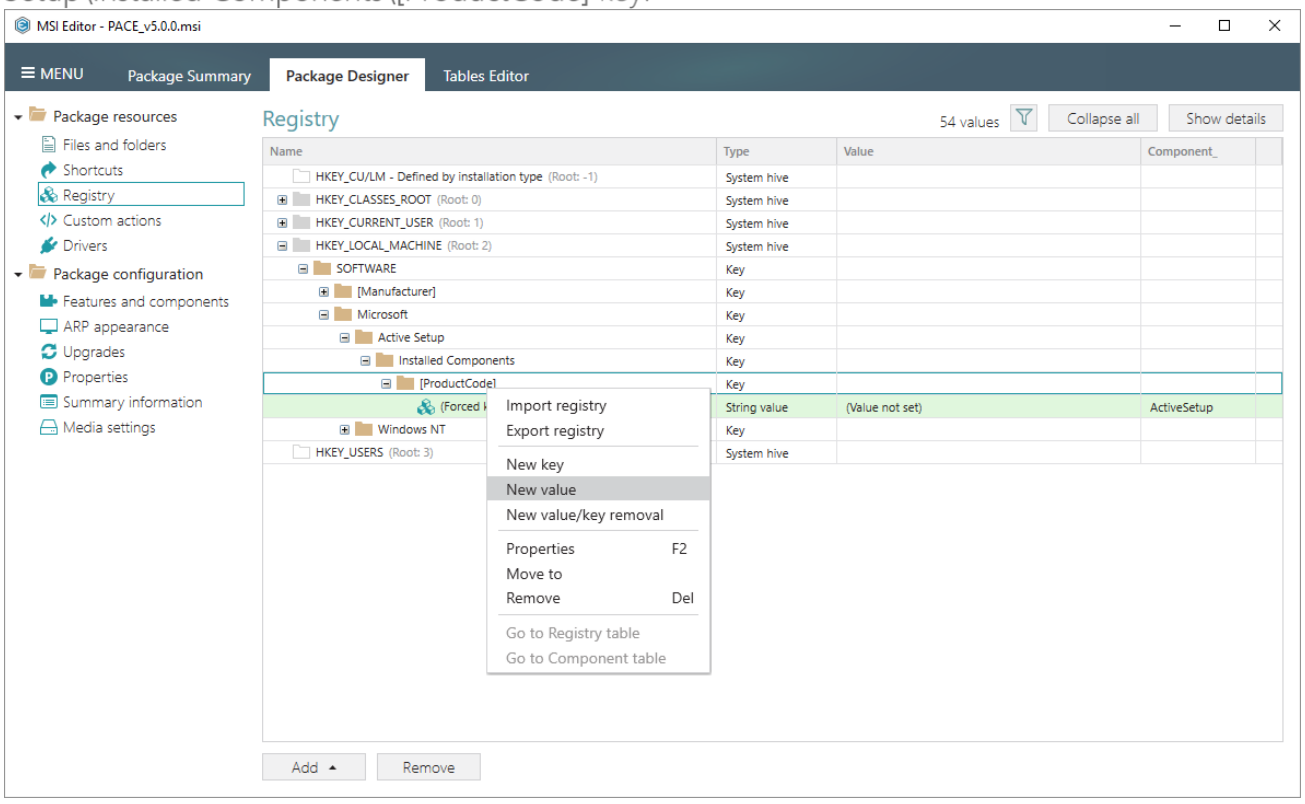


- Enter the Software\Microsoft\Active Setup\Installed Components\[ProductCode] key name, select the previously created ActiveSetup component (or use your existing component) and

click Add.

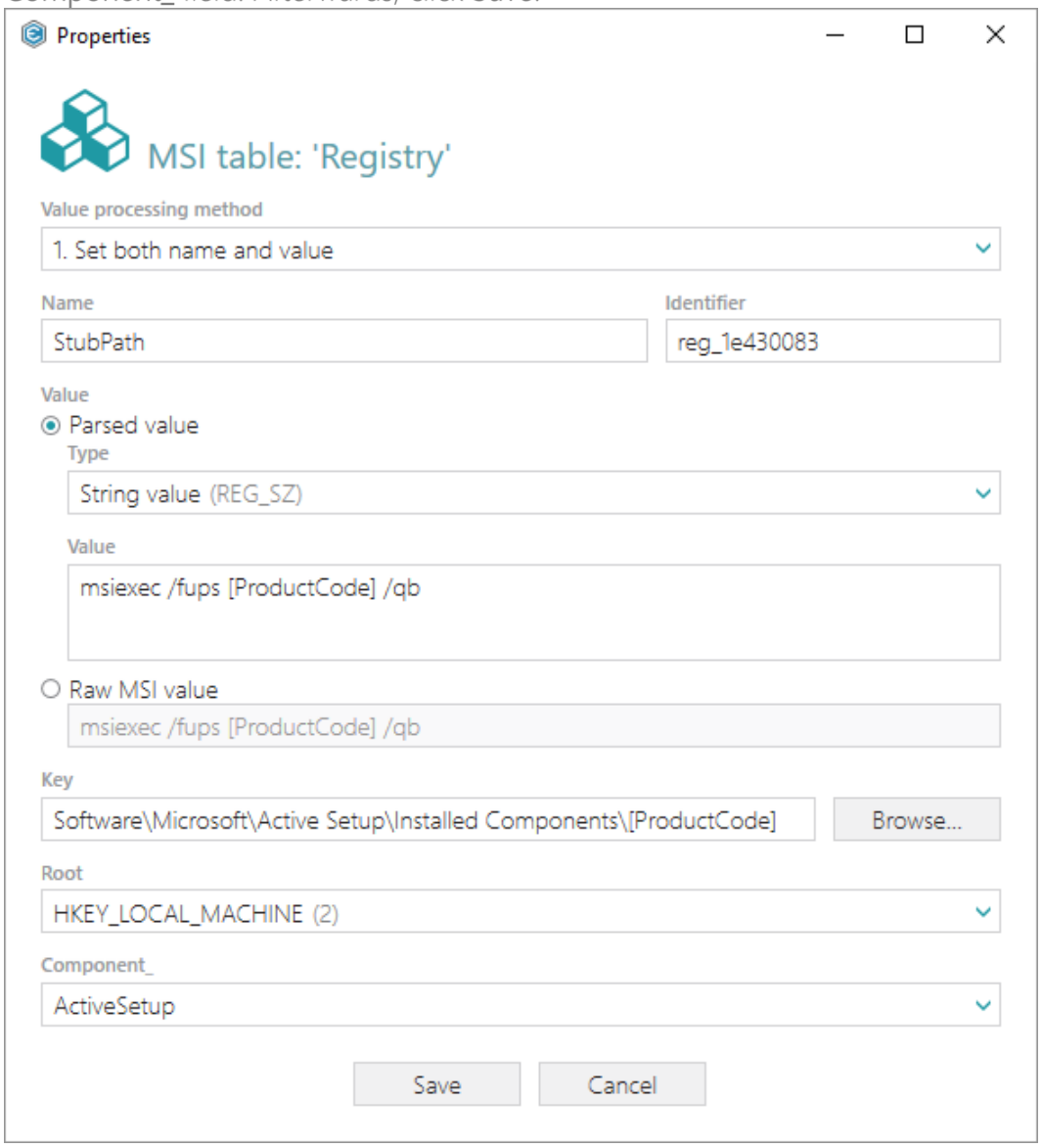


- Select **New value** from the context menu of the created **Software\Microsoft\Active Setup\Installed Components\[ProductCode]** key.

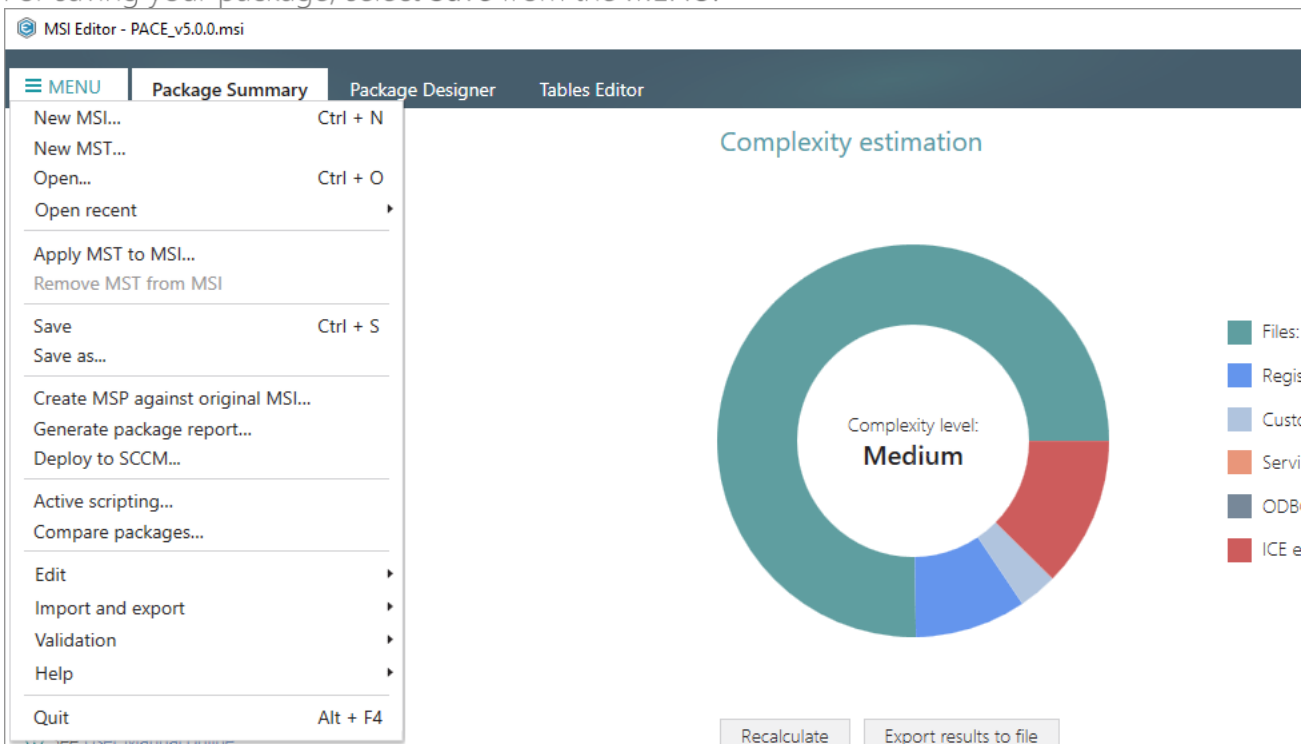


- Enter the "*StubPath*" to **Name**, the "*msiexec /fups [ProductCode] /qb*" to **Value**, and select the previously created **ActiveSetup** component (or use your existing component) from the

Component_ field. Afterwards, click Save.



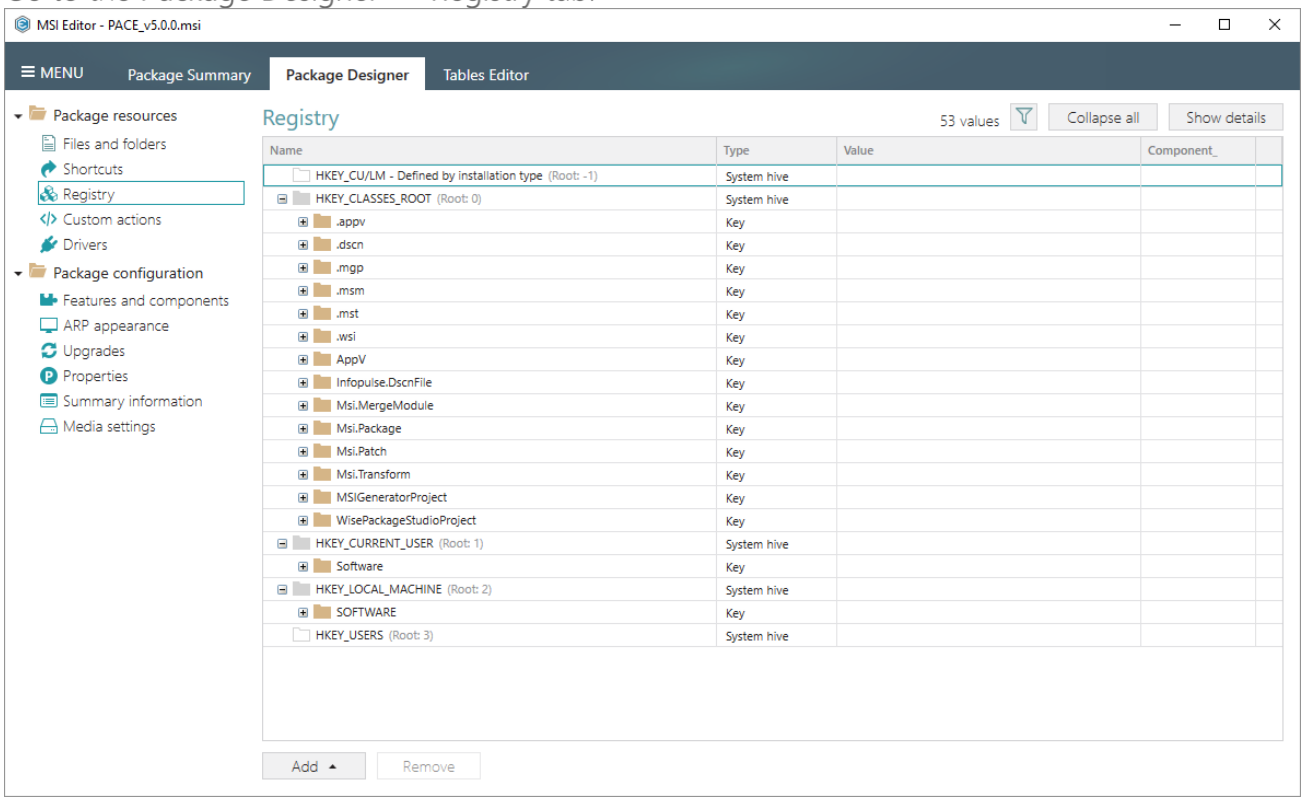
- For saving your package, select Save from the MENU.



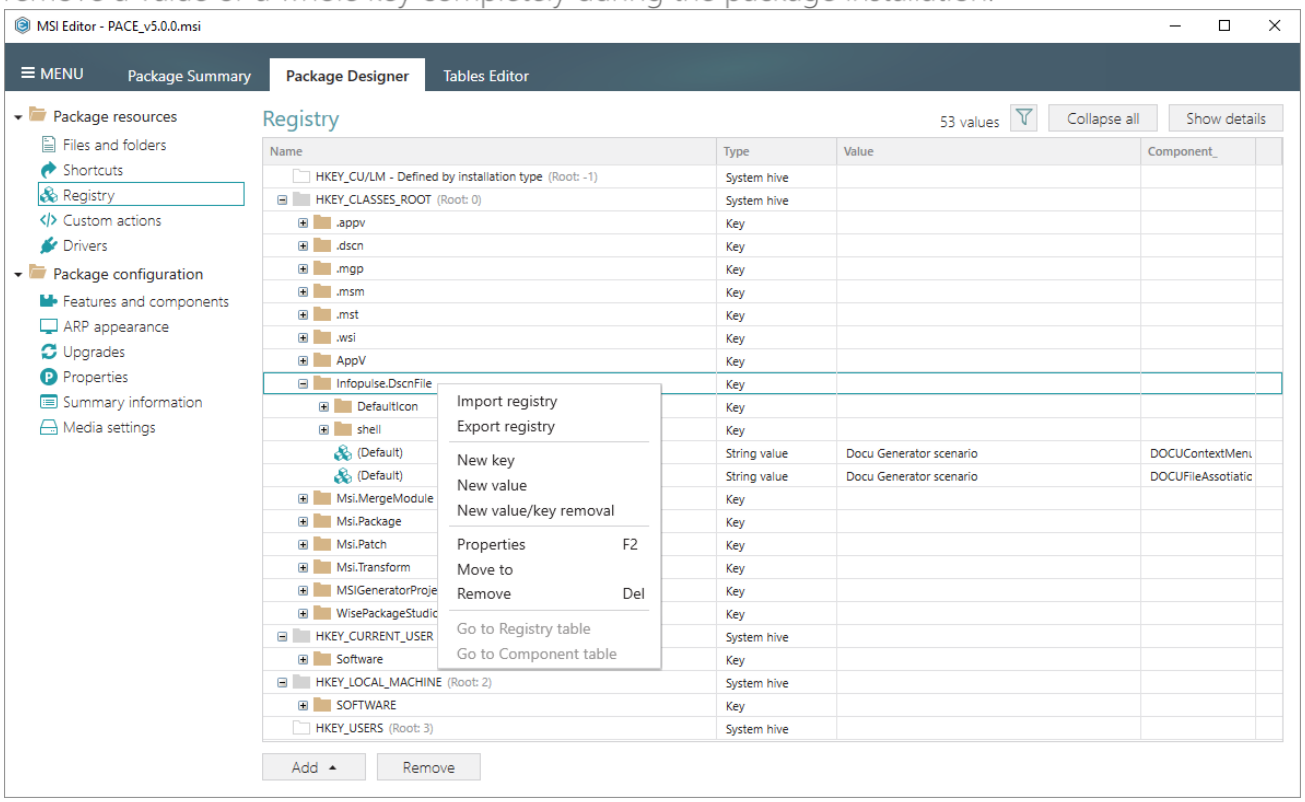
3.7.6.3 Add Value/Key Removal

Add the RemoveRegistry to force a registry value or key removal on install.

- Go to the Package Designer -> Registry tab.

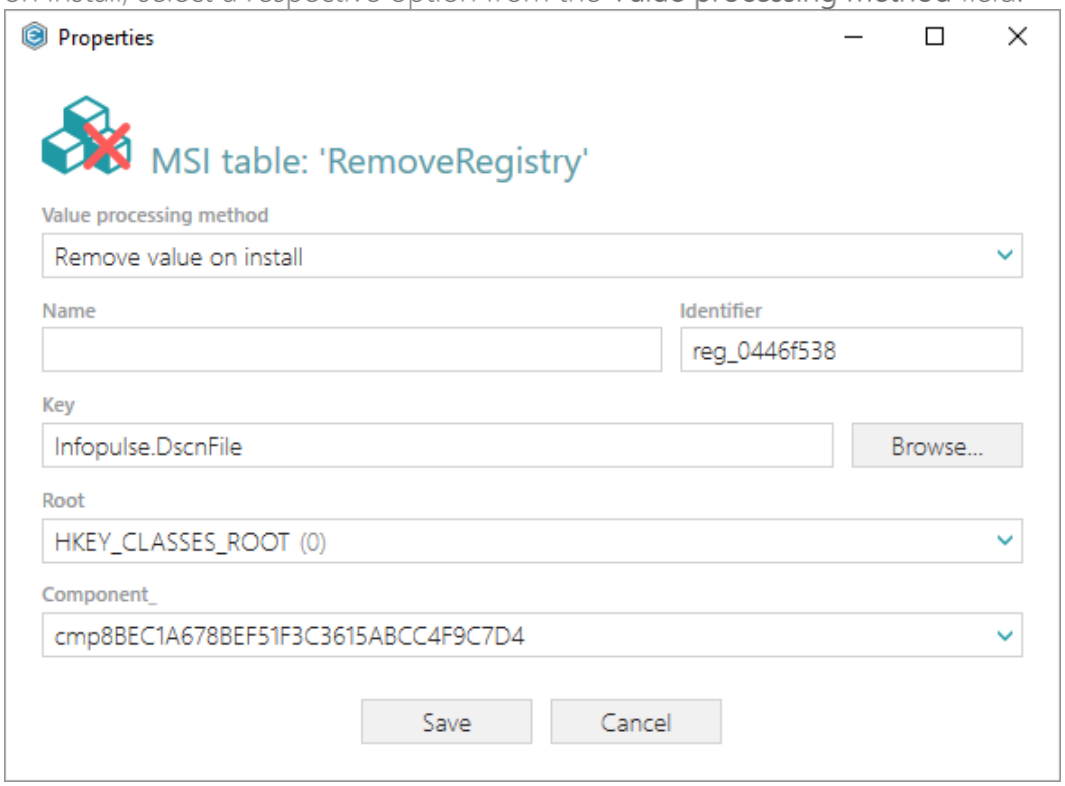


- Select New value/key removal from the context menu of a key, from which you want to remove a value or a whole key completely during the package installation.

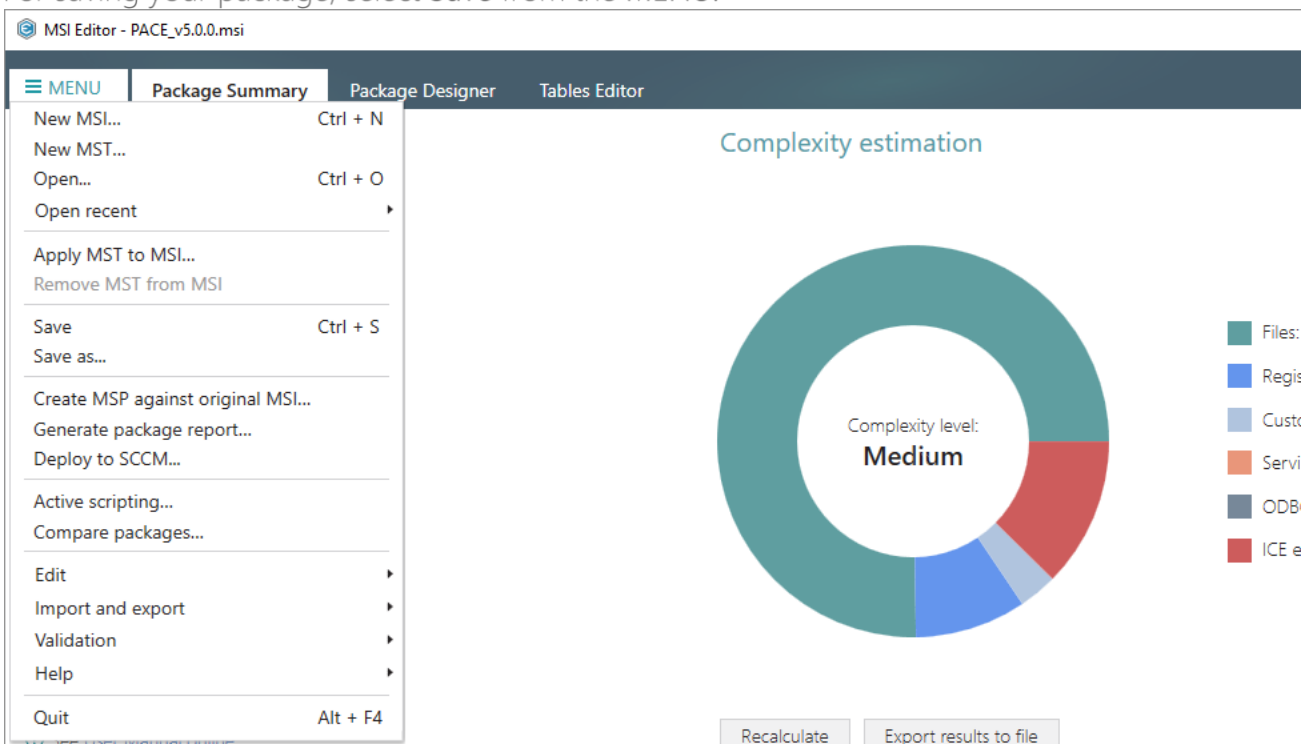


- For removing a registry value, specify its Name, select a Component from the list, which will controls the deletion of thee specified registry value, and then click Save.

  For cleaning-up a Default key value, or removing a whole key with all of its values and subkeys on install, select a respective option from the Value processing method field.



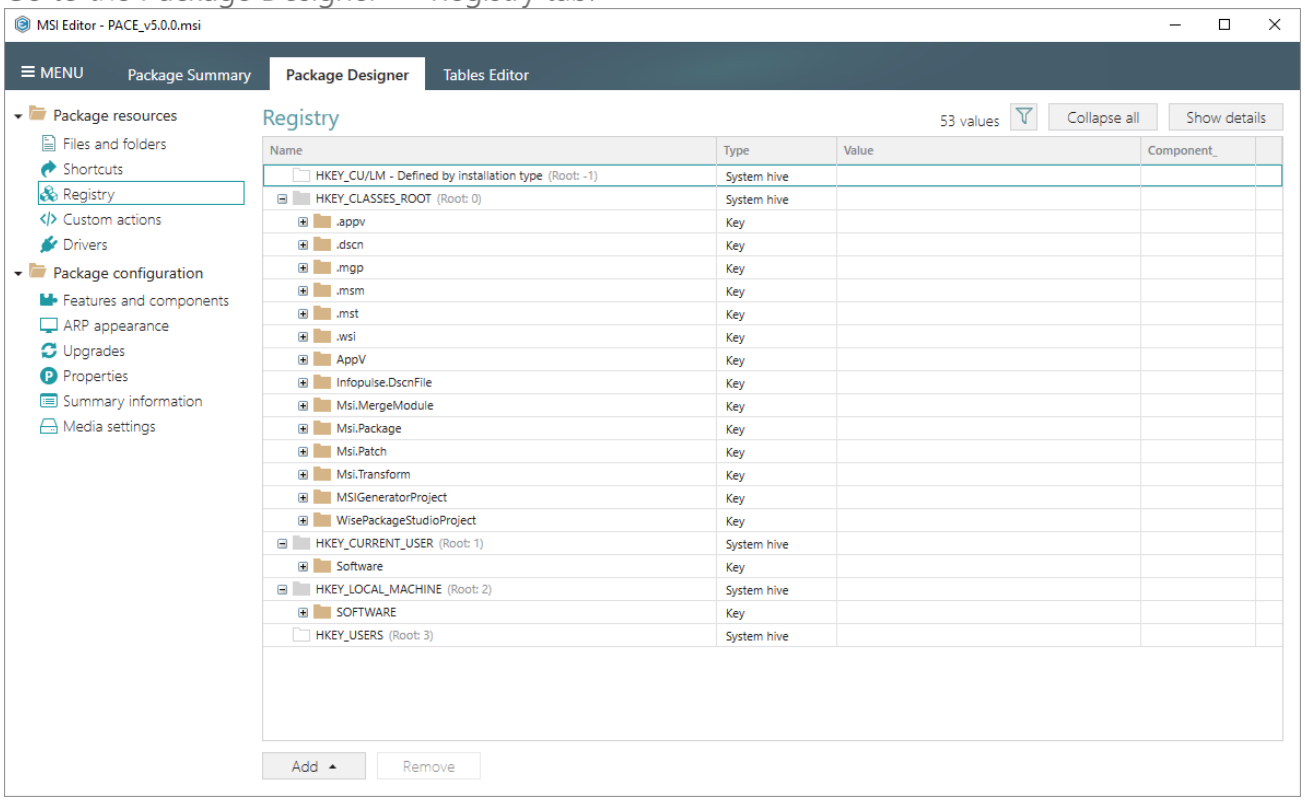
- For saving your package, select Save from the MENU.



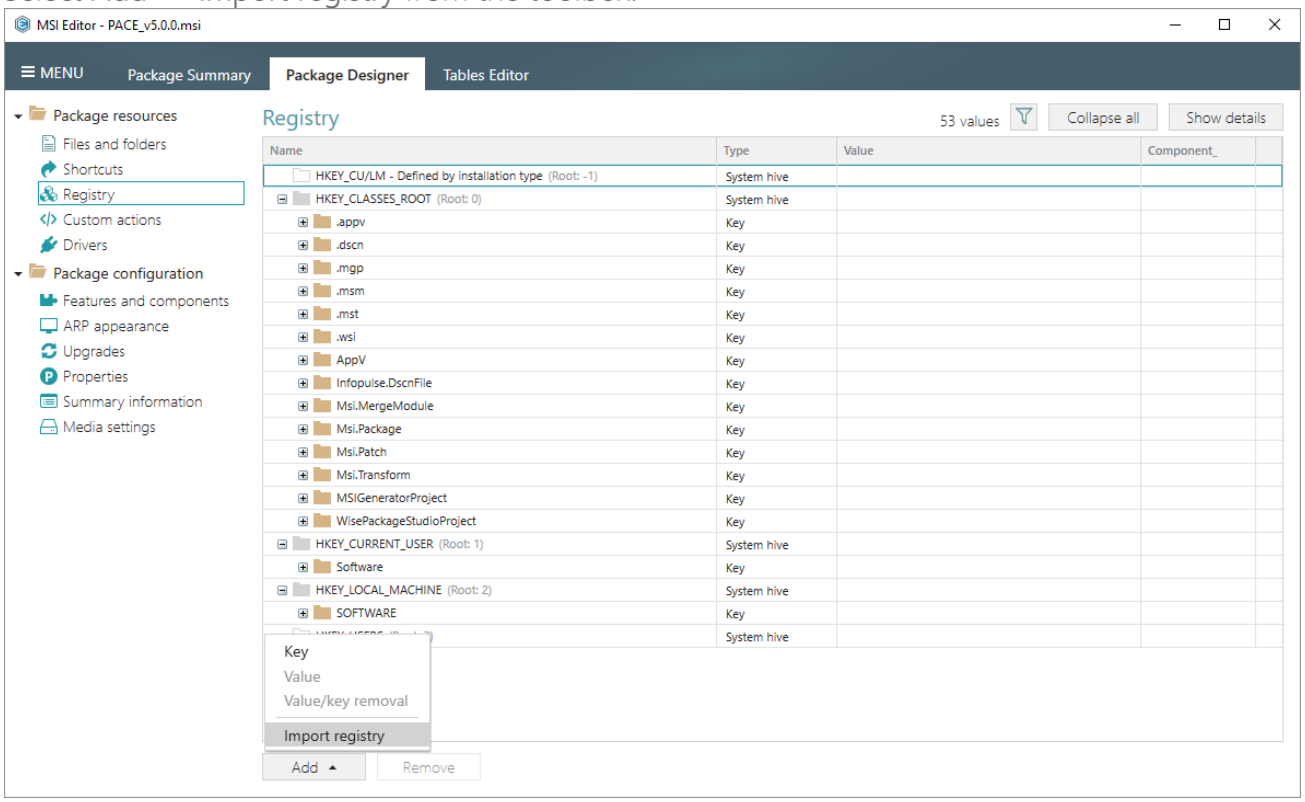
3.7.6.4 Import Registry

Import registry from the REG file.

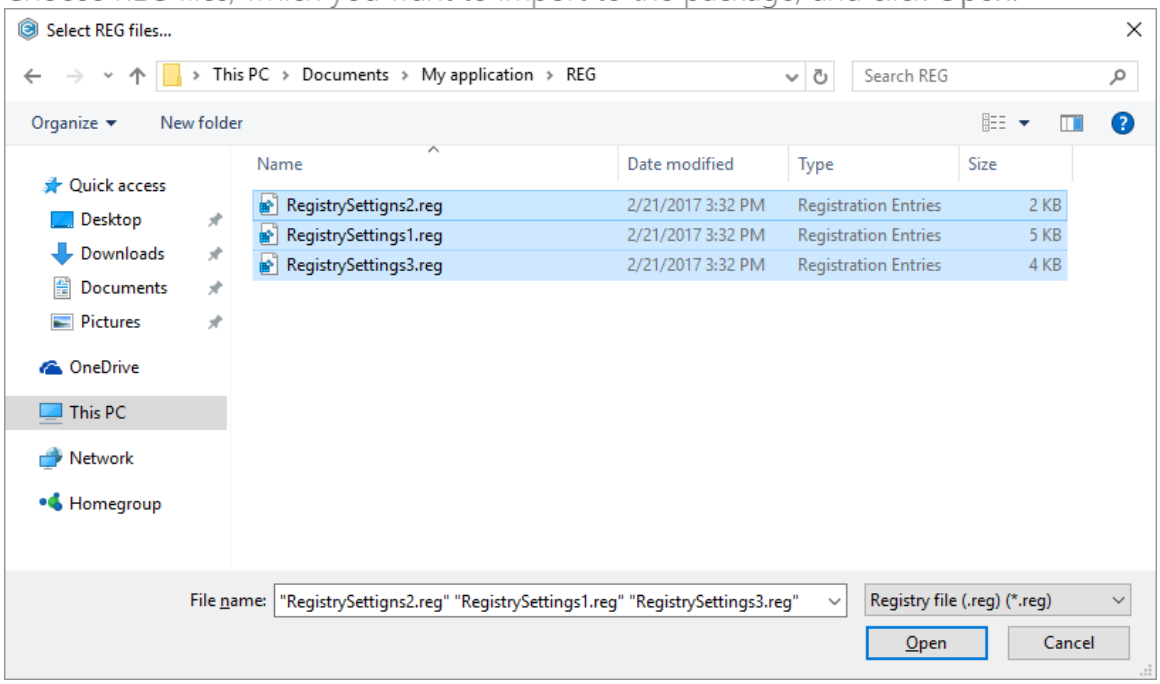
- Go to the Package Designer -> Registry tab.



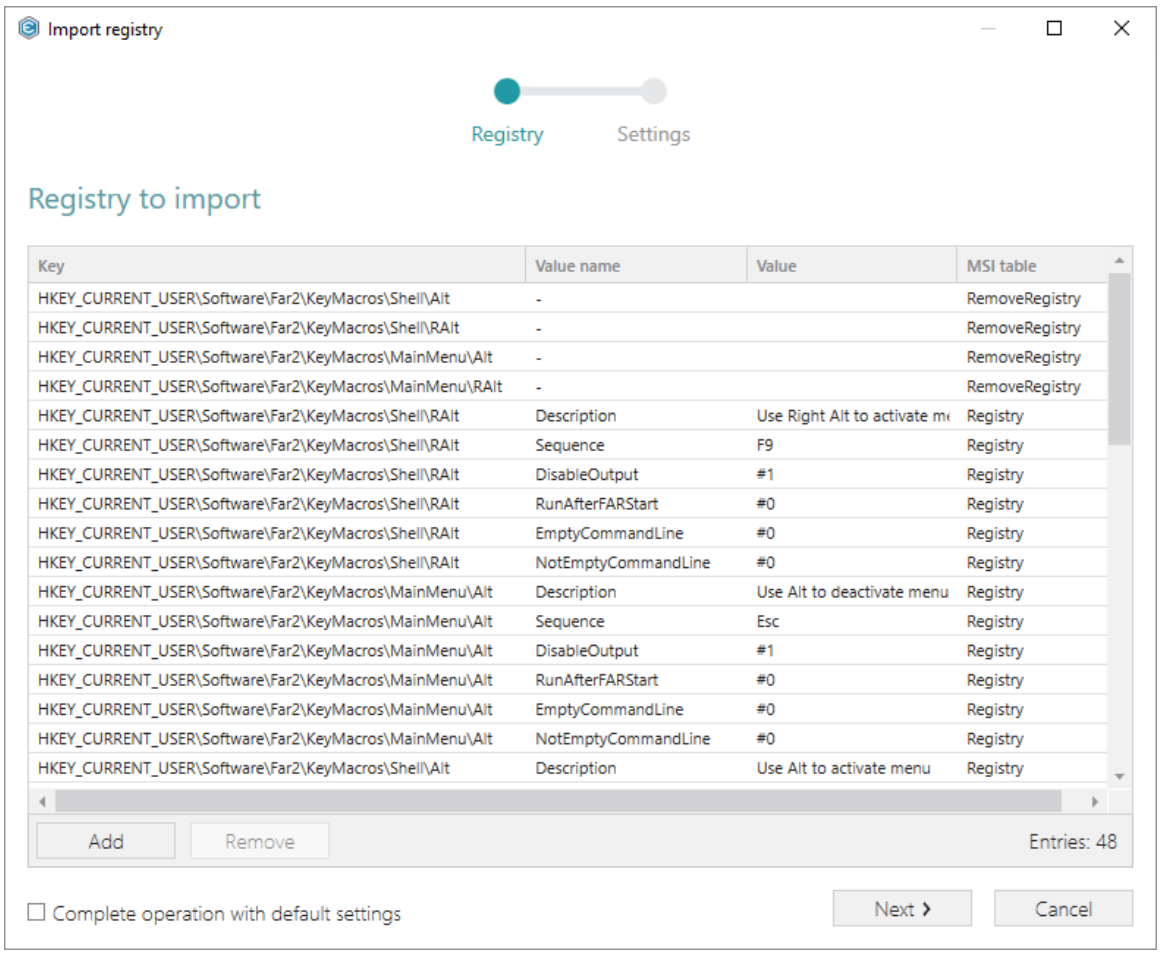
- Select Add -> Import registry from the toolbox.



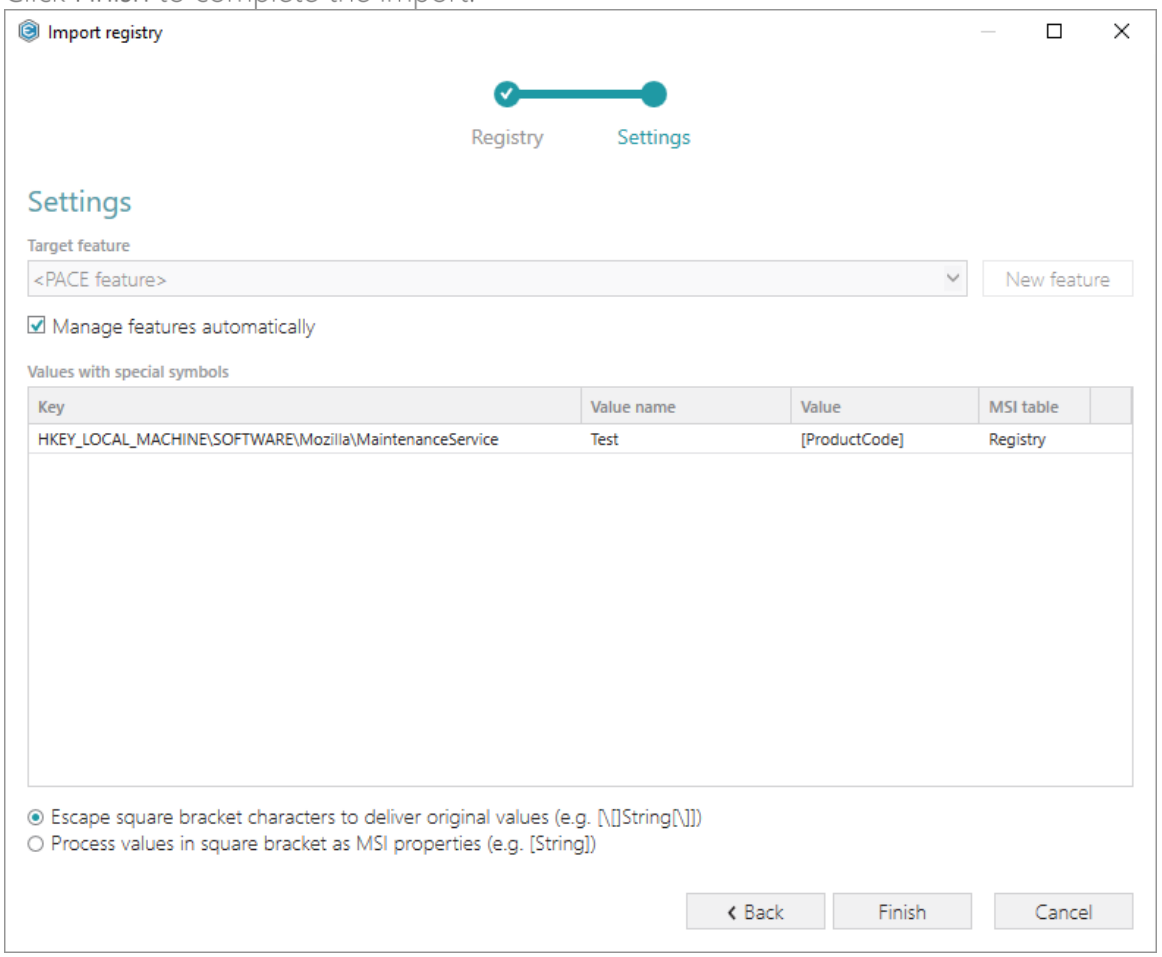
- Choose REG files, which you want to import to the package, and click Open.



- Review registry entries, parsed from the selected REG files, remove the unnecessary ones, and click Next.



- At the **Settings** step, you can select an MSI Feature, to which components with the registry entries will be assigned. If the registry entries, which have been read, contain values in square brackets, you can select how these values will be processed – as MSI properties or as ordinary registry values. Below, in the tables, you will find the detailed description of these settings. Click **Finish** to complete the import.



Feature settings

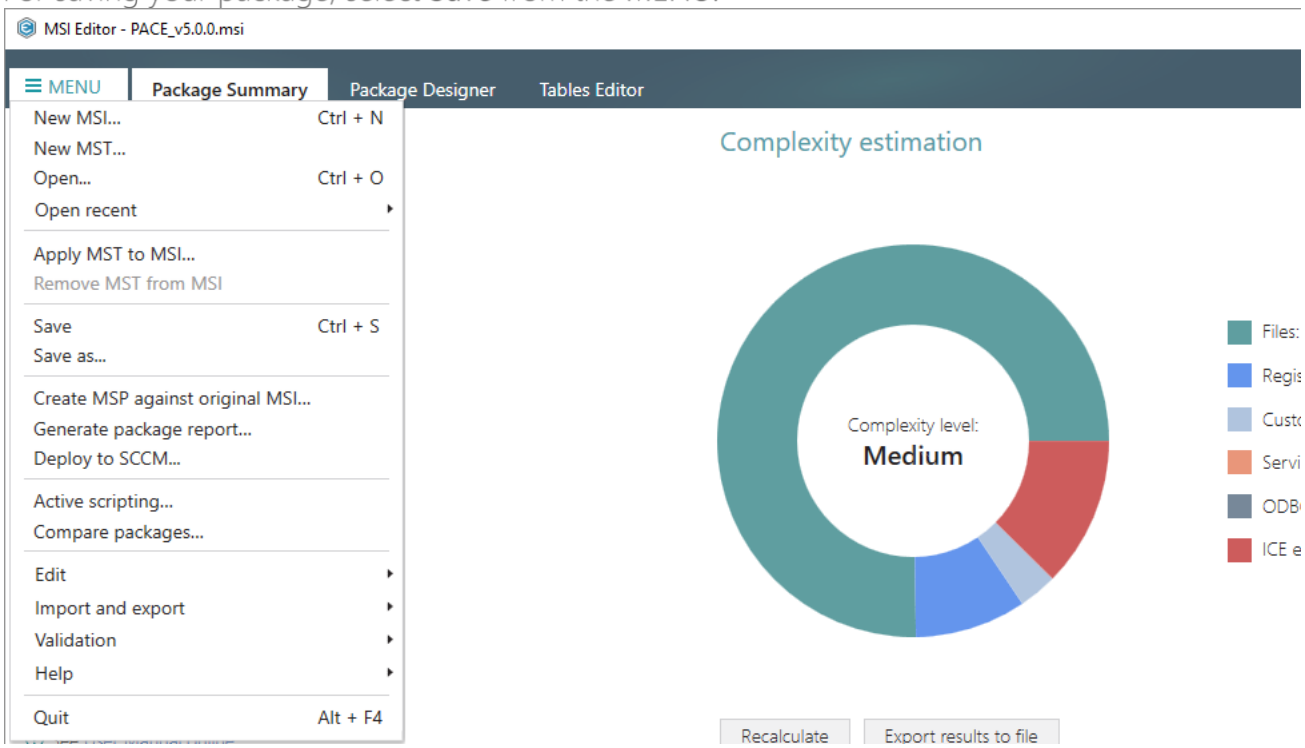
Manage features automatically	The specific PACE features will be created (or used if they exist) for the components, containing the imported registry. Feature structure is based on the packaging best practices and depends on the registry key location.
<a feature, selected from the list>	Components with registry will be assigned to a feature, selected from the list.

Values with special symbols

Escape square bracket characters to deliver original values	Values in square brackets, displayed in the list, will be saved to MSI as ordinary strings using the general rules for escaping the special characters. This approach guarantees the delivery of the imported registry entries to the target system with their original values.

Process values in square brackets as MSI properties	Values in square brackets, displayed in the list, will be saved to MSI as MSI properties. Thus, each MSI property will be replaced with its value during the MSI installation. If a property does not exist in MSI during the installation, it will be replaced with the blank value.

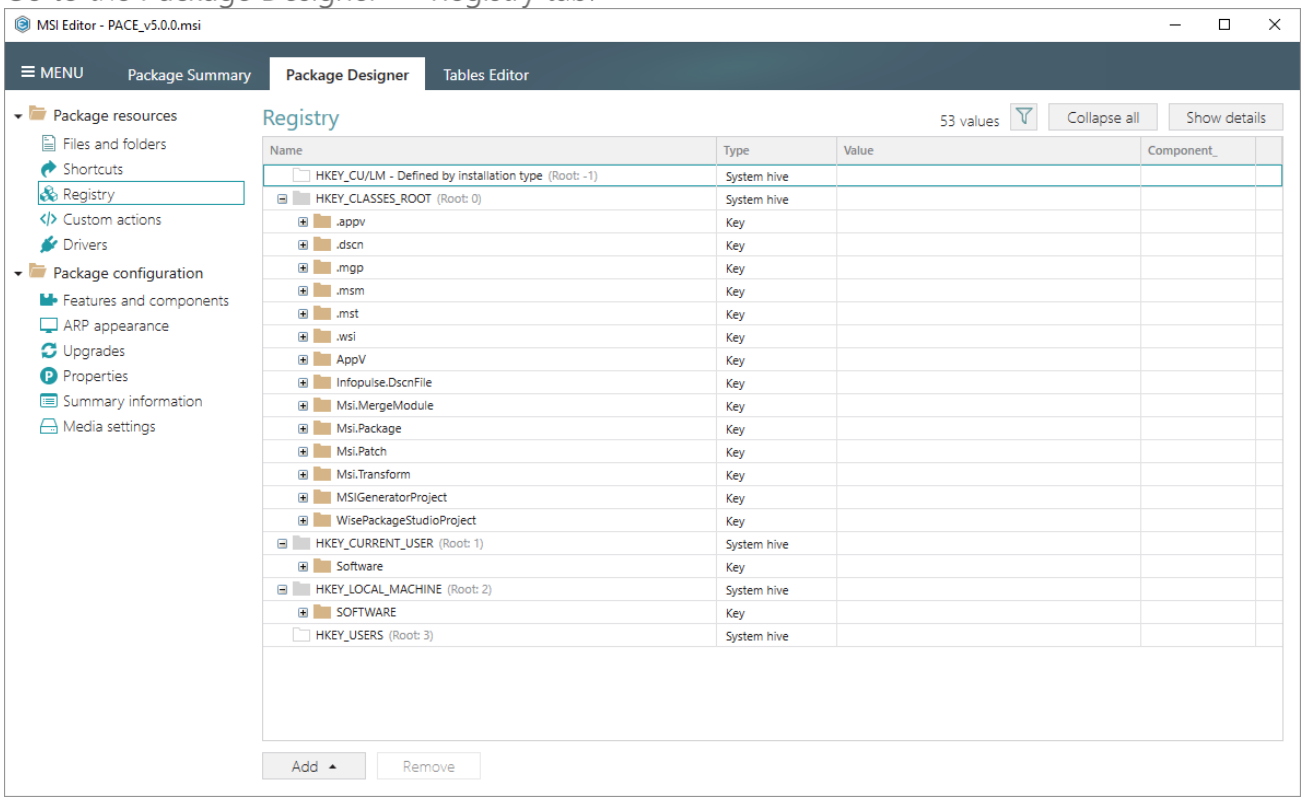
- For saving your package, select Save from the MENU.



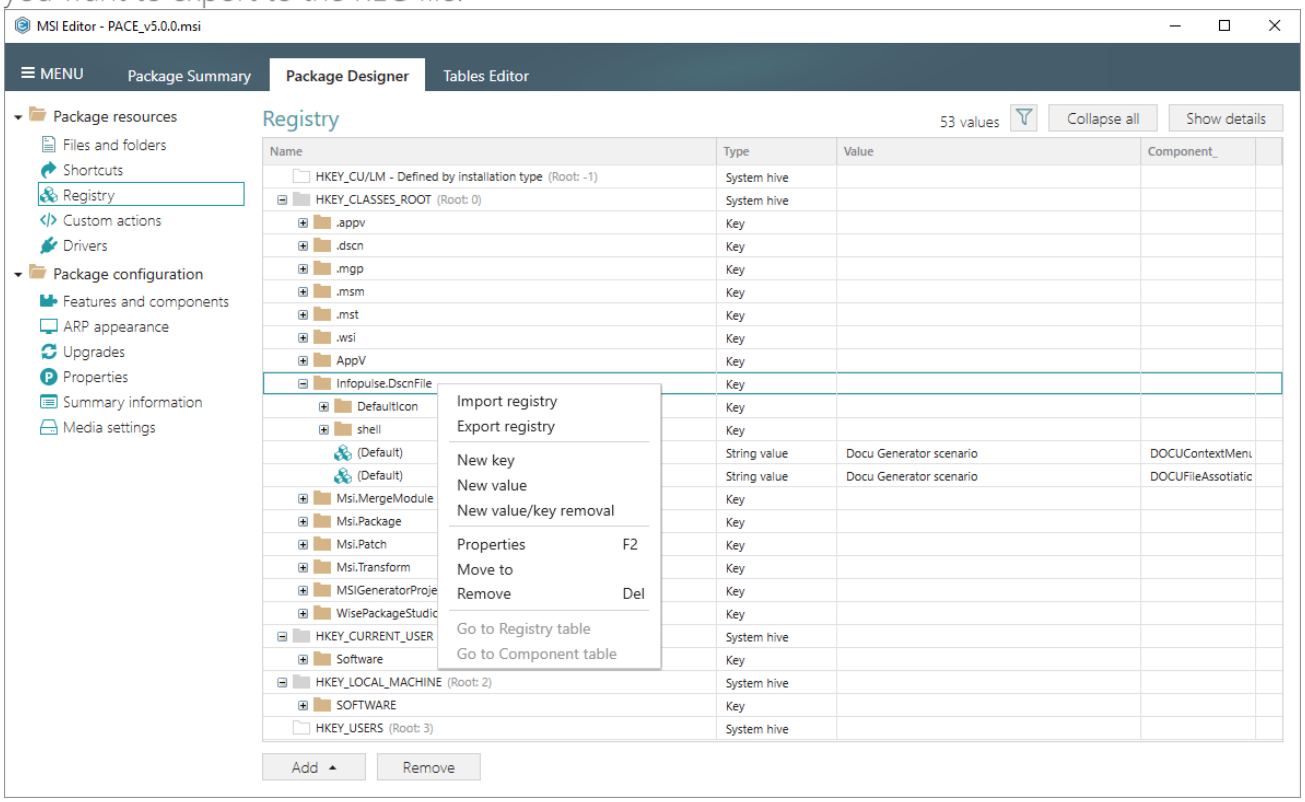
3.7.6.5 Export Registry

Export the selected registry to the REG file.

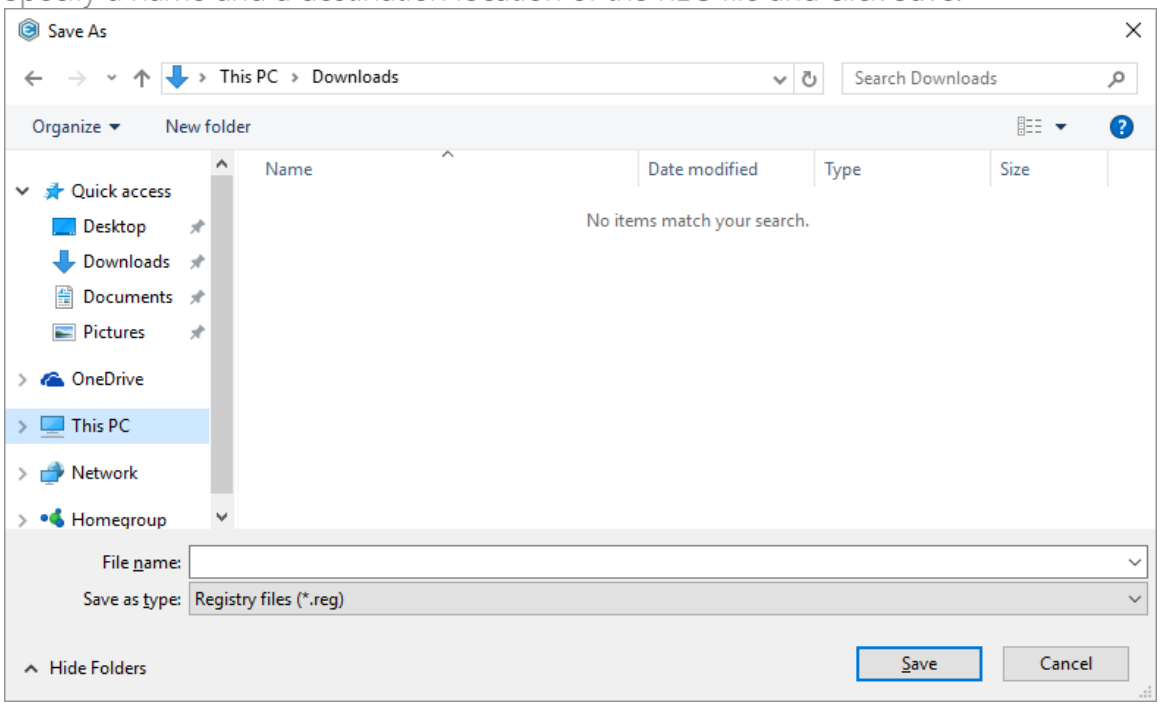
- Go to the **Package Designer** -> **Registry** tab.



- Select **Export registry** from the context menu of a registry item (root hive, key, or value), which you want to export to the REG file.



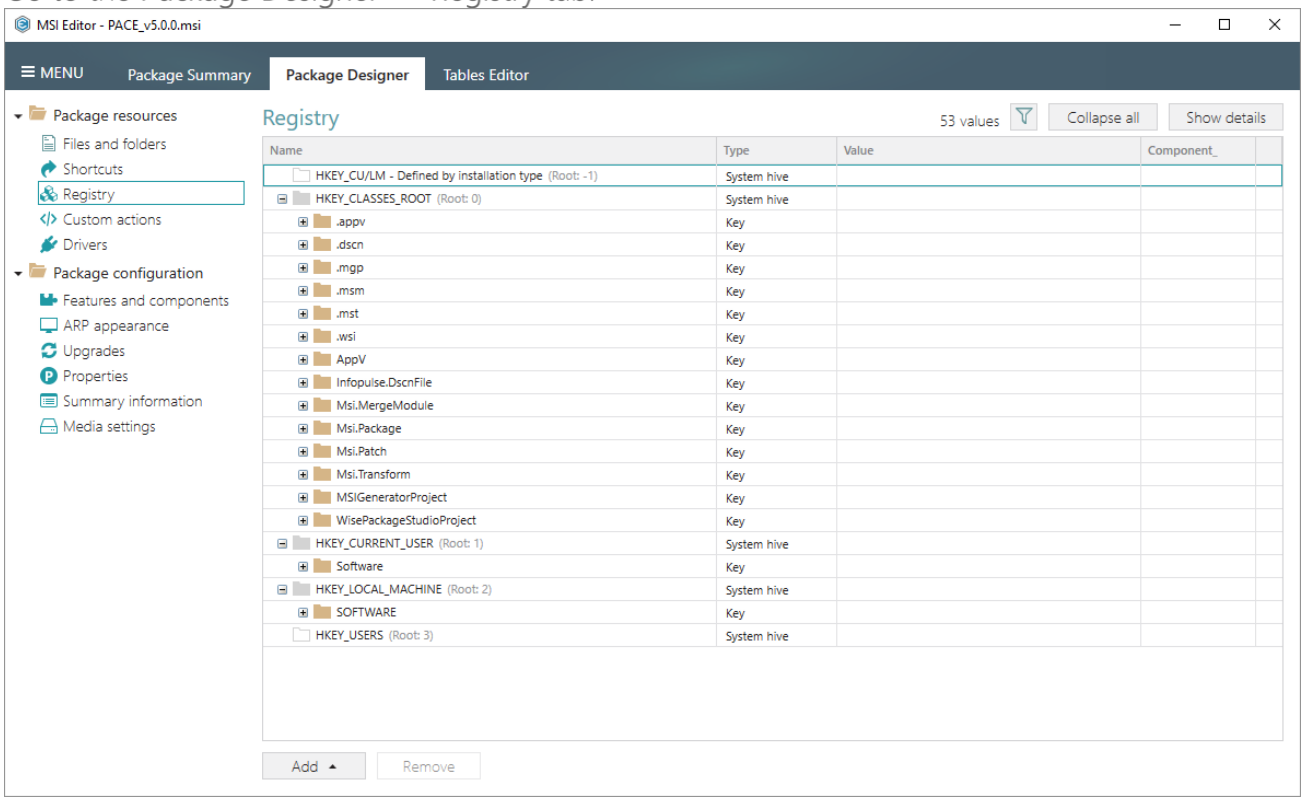
- Specify a name and a destination location of the REG file and click **Save**.



3.7.6.6 Set Permissions

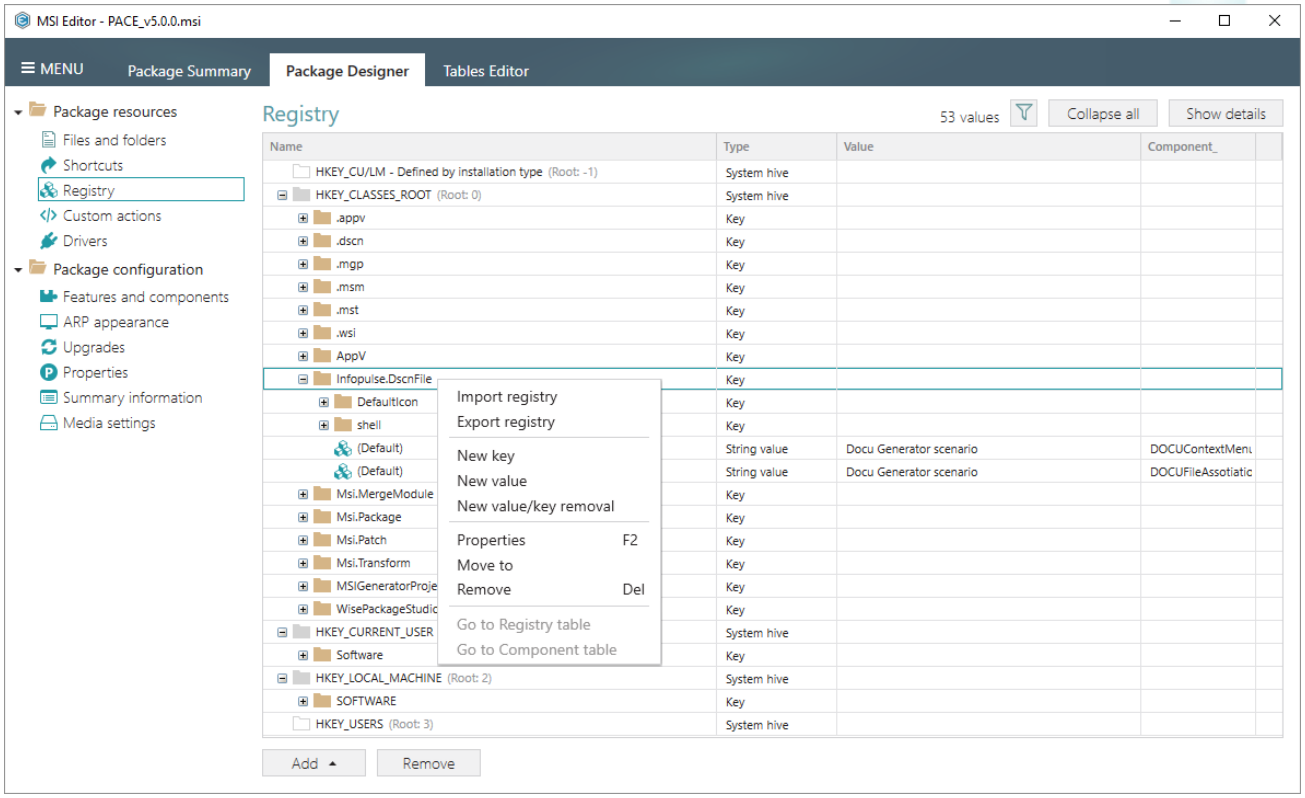
Set custom permissions for the registry key.

- Go to the **Package Designer** -> **Registry** tab.

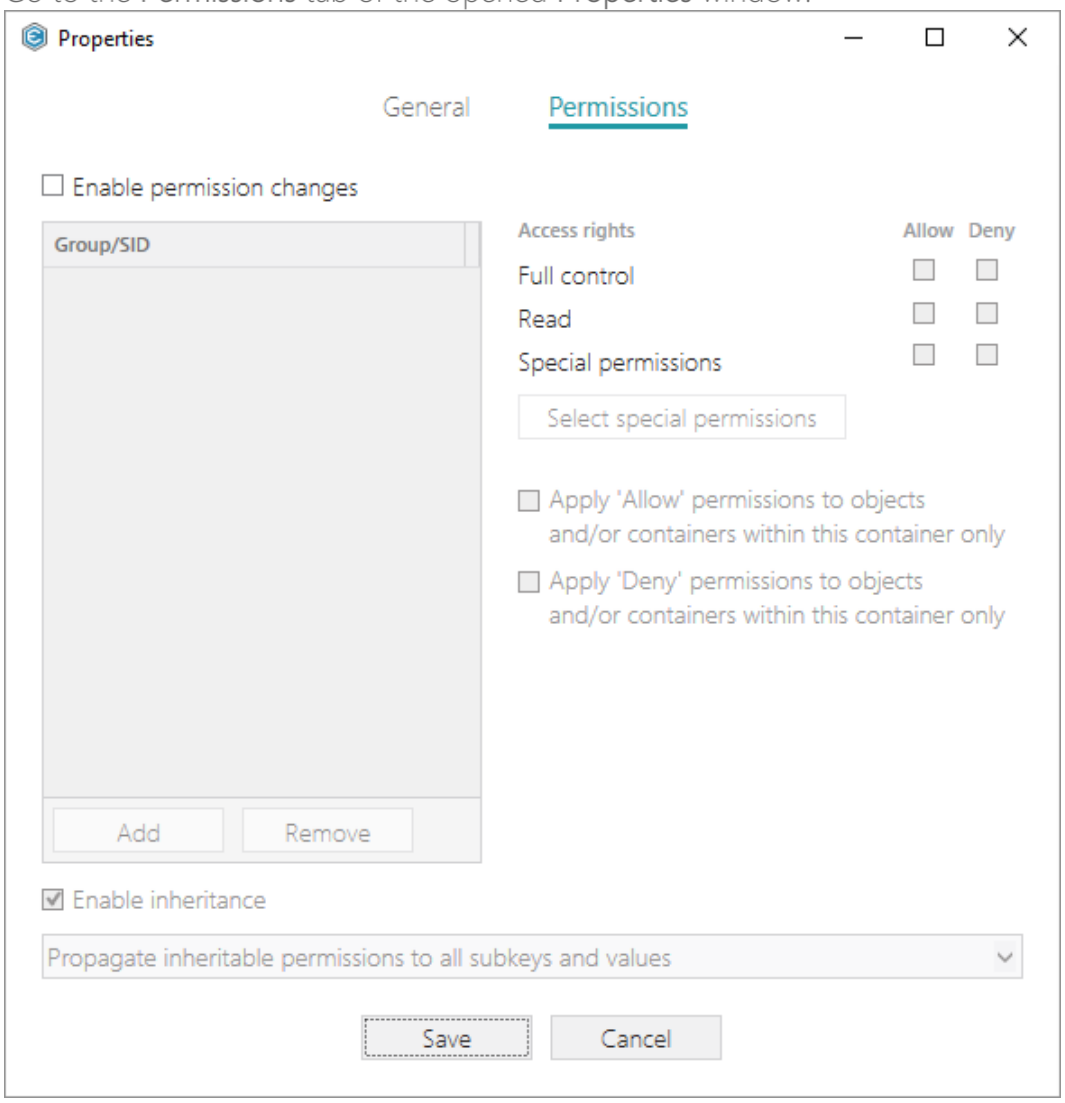


- Select **Properties** from the context menu of a registry key, for which you want to set permissions. Note that permissions could not be changed for the registry values and root

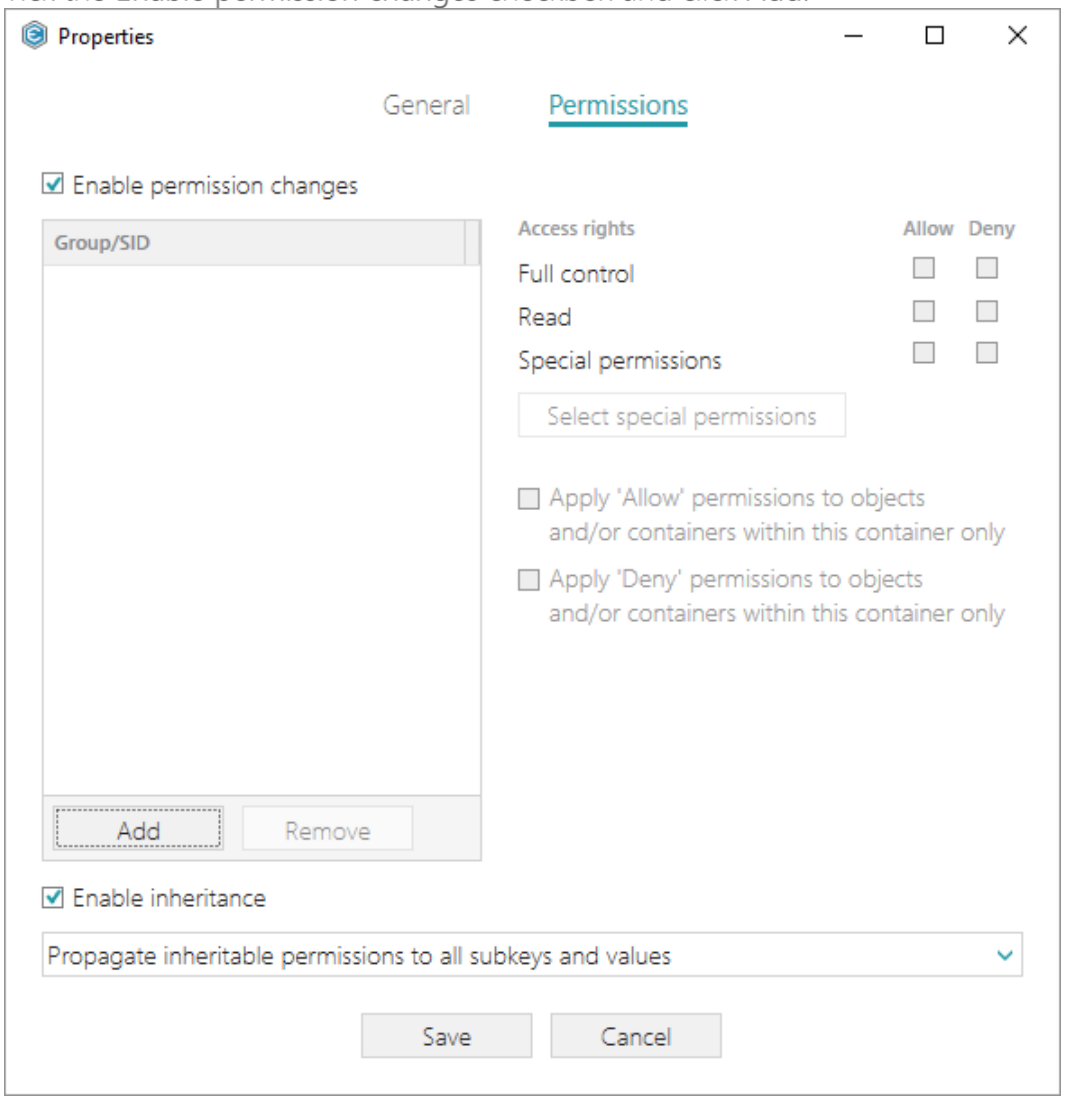
hives.



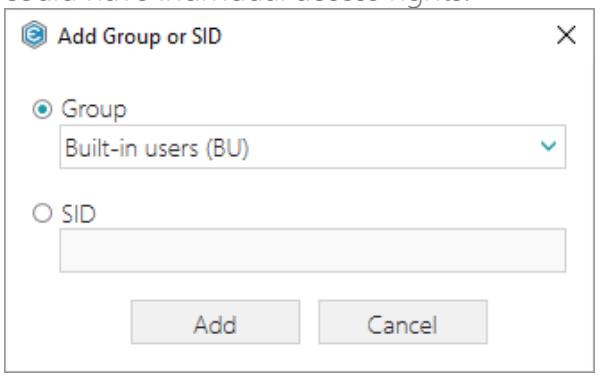
- Go to the **Permissions** tab of the opened **Properties** window.



- Tick the Enable permission changes checkbox and click Add.

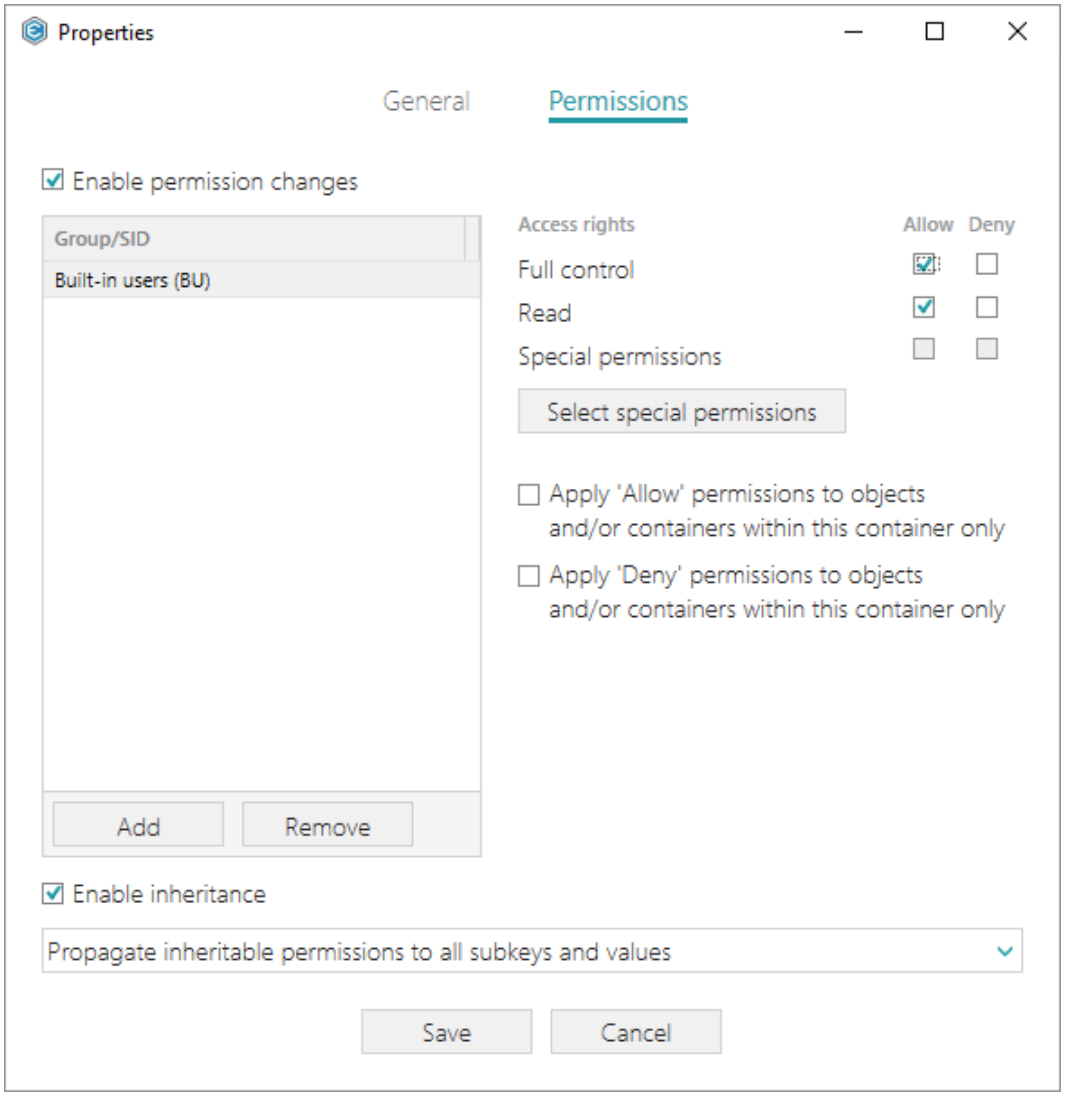


- Select a group from the list or enter SID manually, for which you want to set permissions, and click Add. Note that you can add as many groups and SID as you need. Each group and SID could have individual access rights.

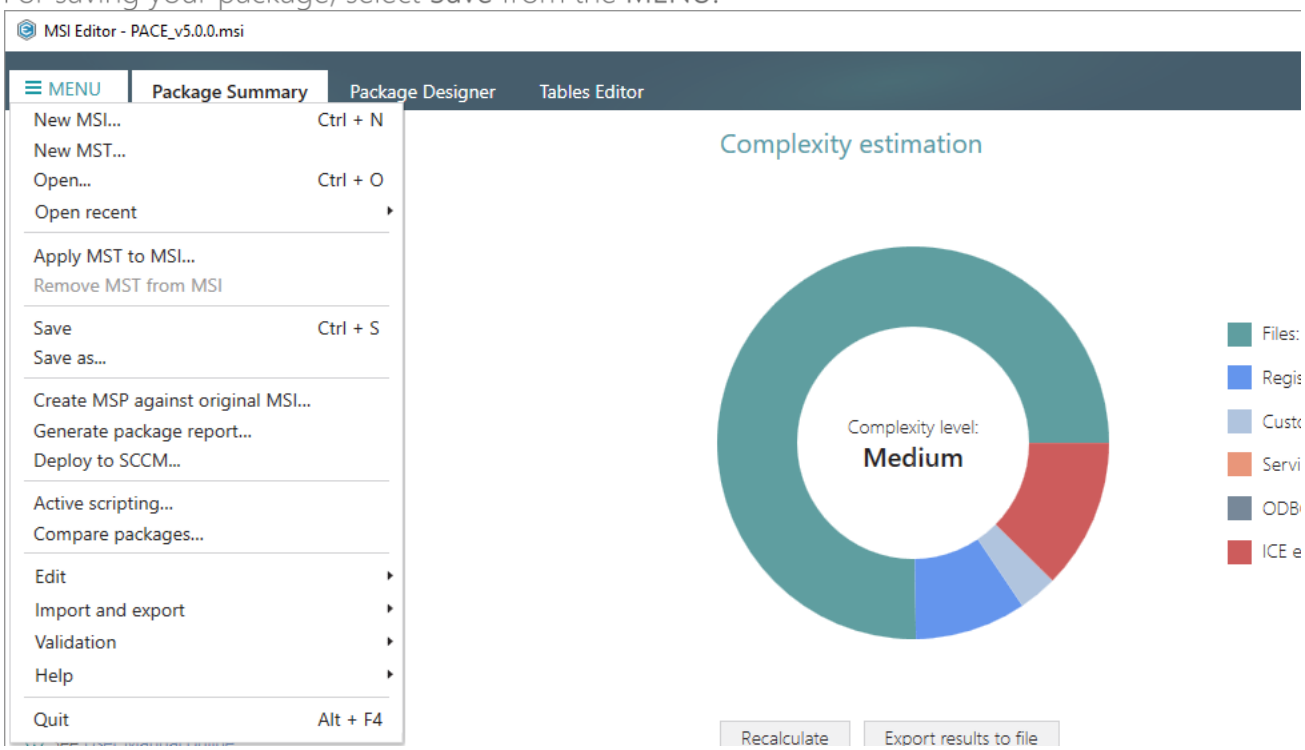


- Tick the necessary access rights for the selected group or SID, change (if needed) special permissions, inheritance options and behavior of applying the permissions. Click Save to save the changed permission settings to MSI. Note that permissions will be propagated using the

SecEdit tool.



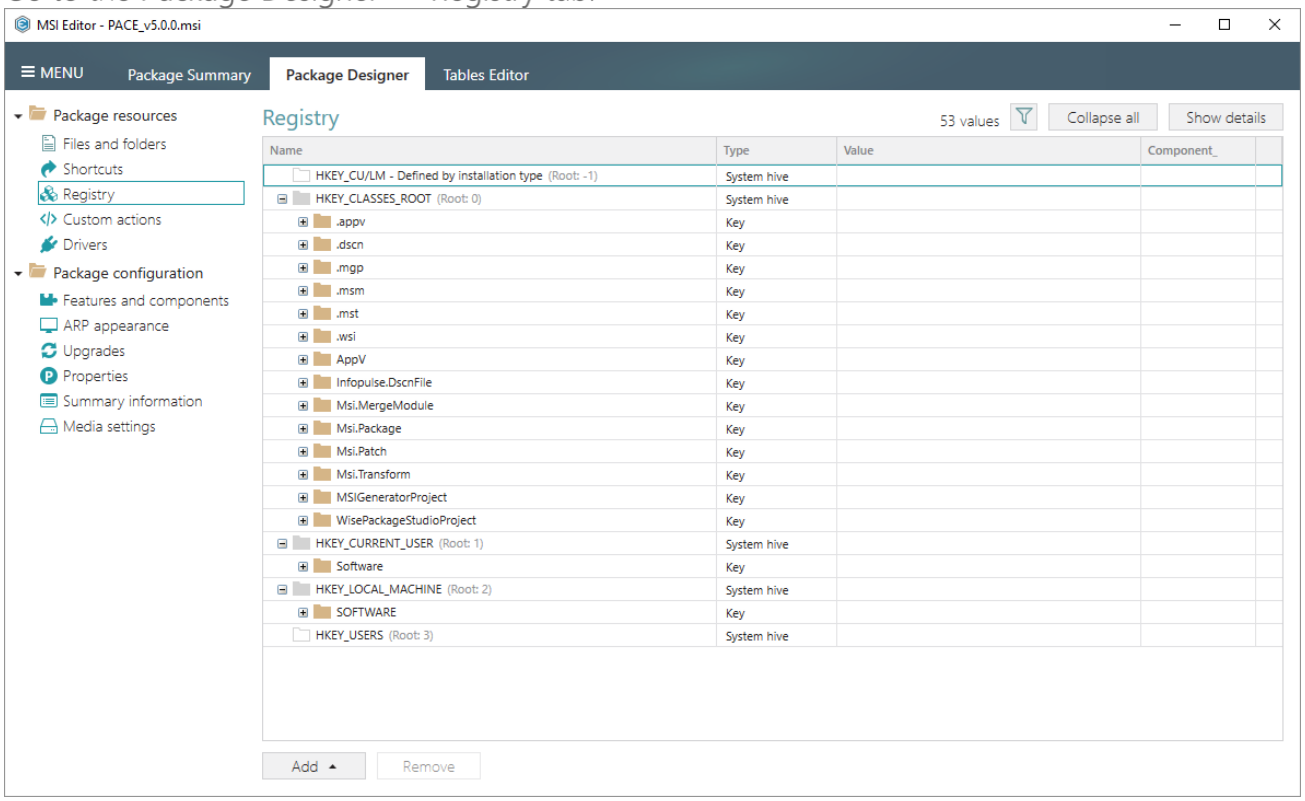
- For saving your package, select **Save** from the **MENU**.



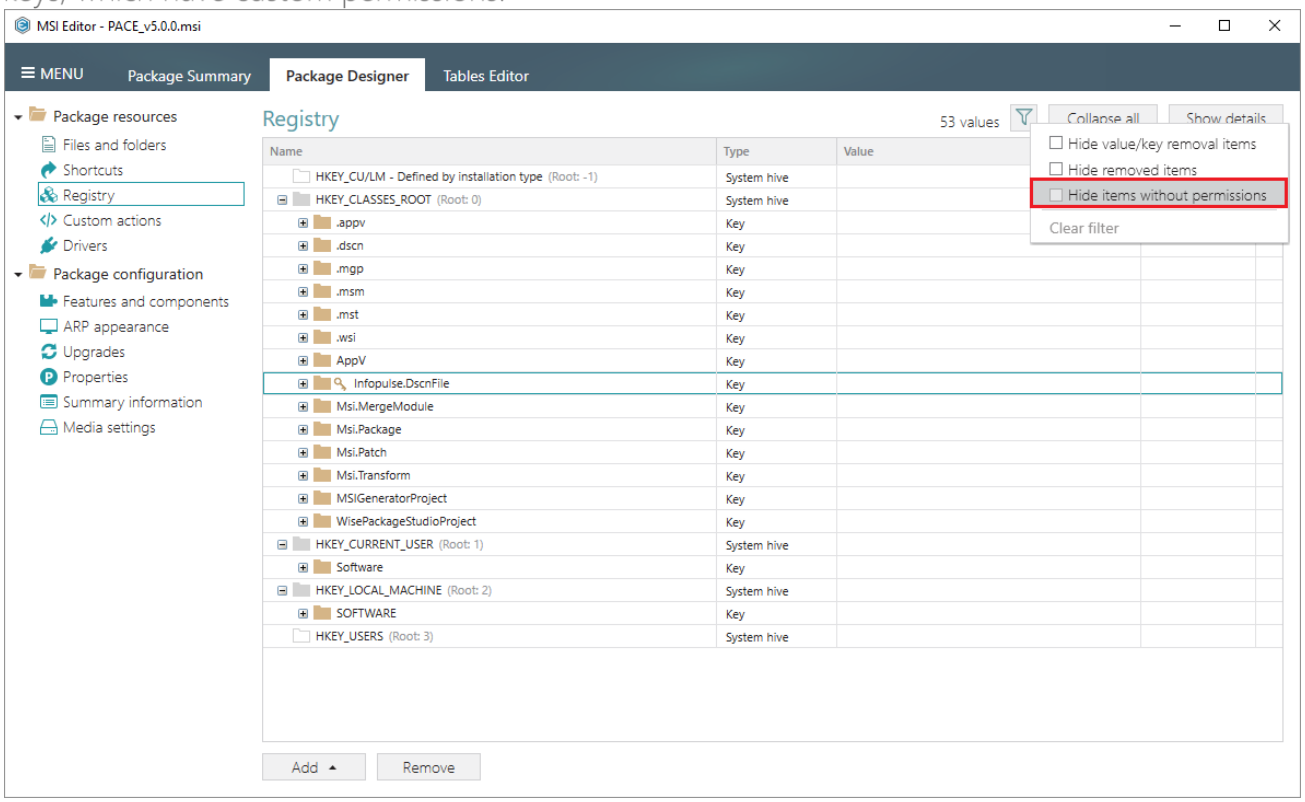
3.7.6.7 Remove Permissions

Locate registry key with custom permissions and remove the unnecessary permissions.

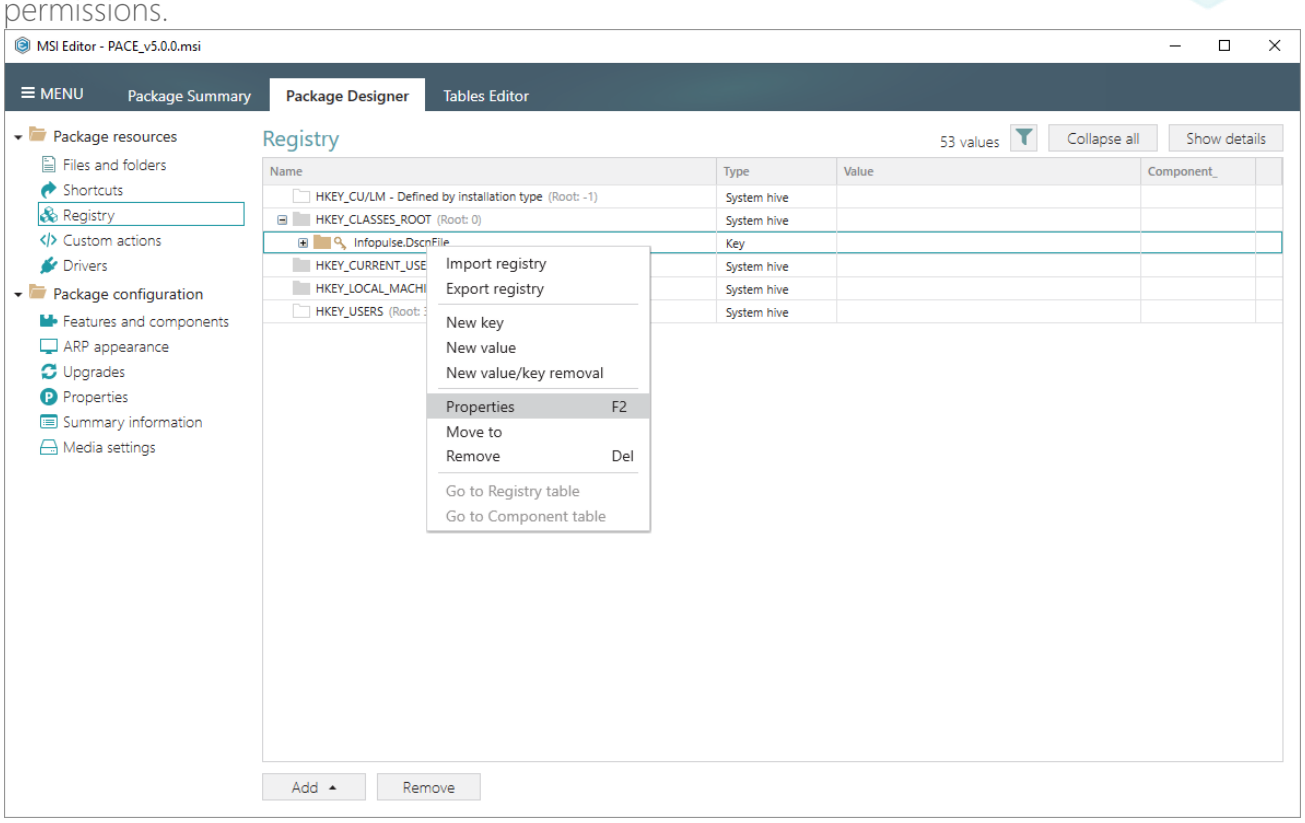
- Go to the Package Designer -> Registry tab.



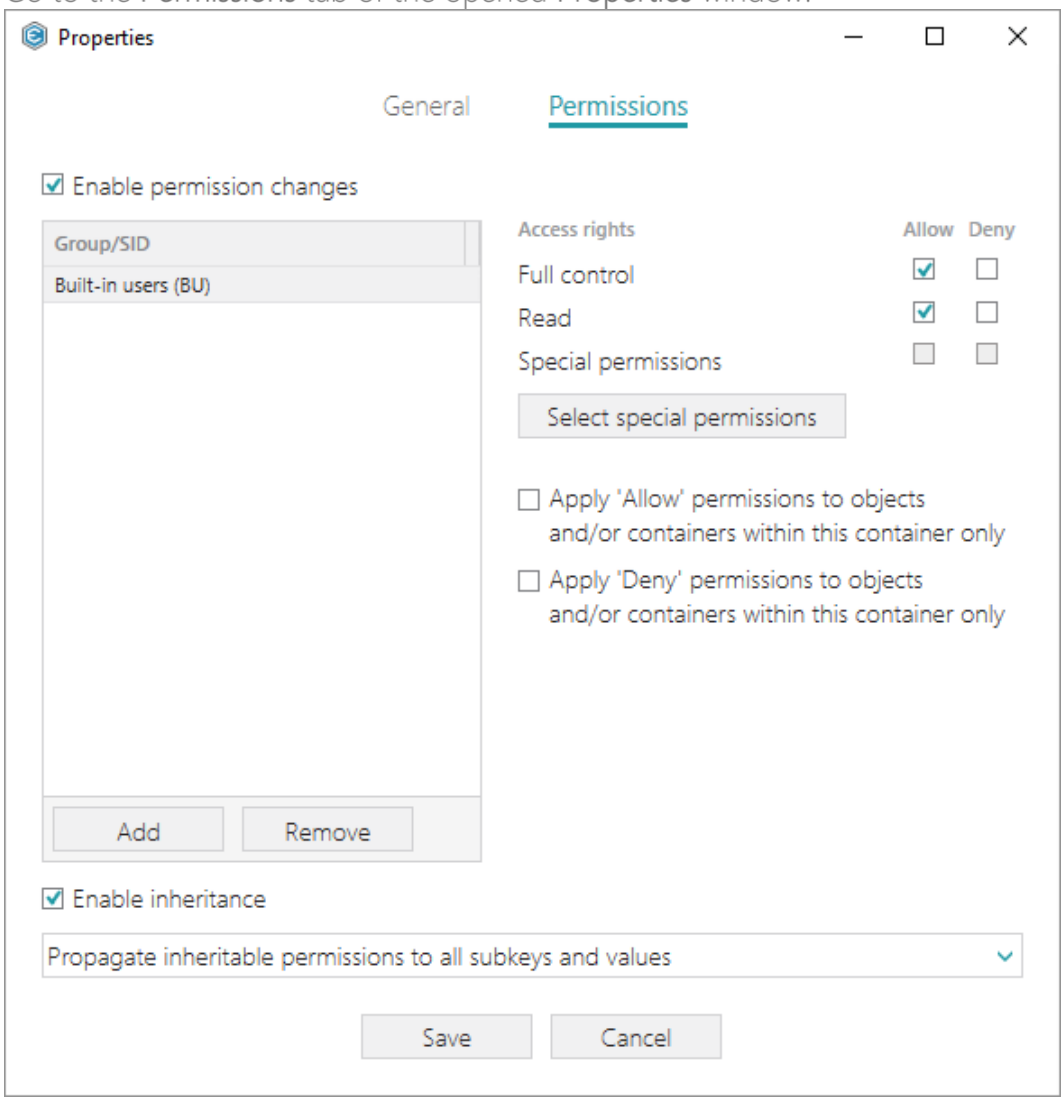
- *(Optional step)* Apply the Hide items without permissions filter to display only those registry keys, which have custom permissions.



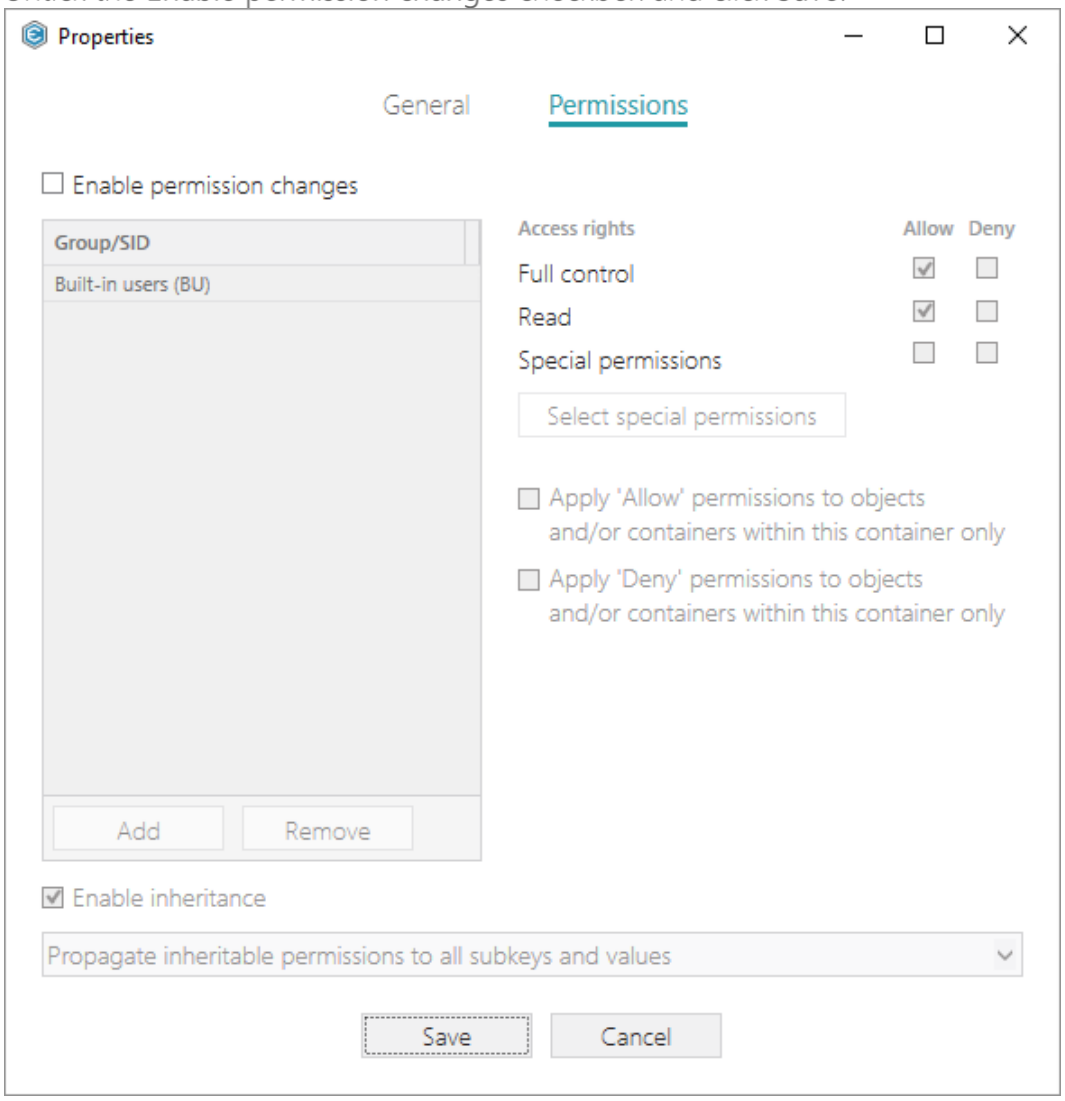
- Select **Properties** from the context menu of a registry key, for which you want to remove permissions.



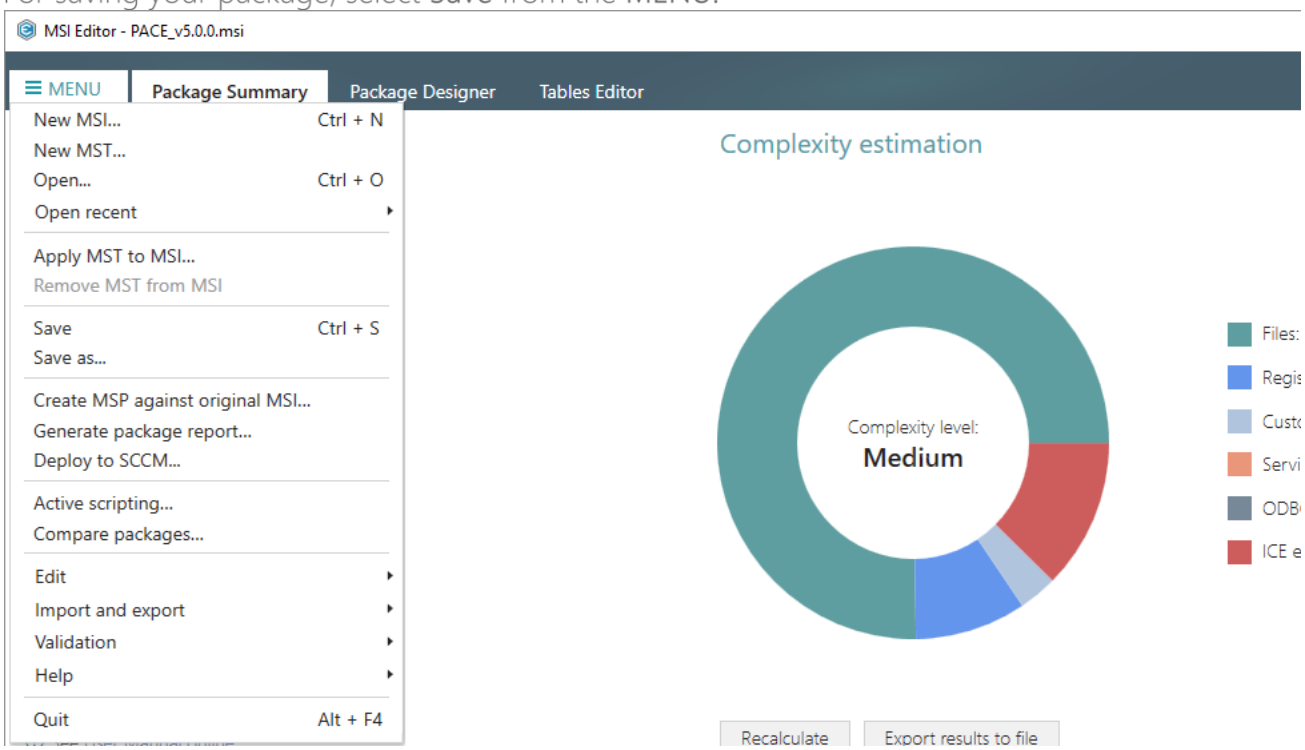
- Go to the **Permissions** tab of the opened **Properties** window.



- Untick the Enable permission changes checkbox and click Save.



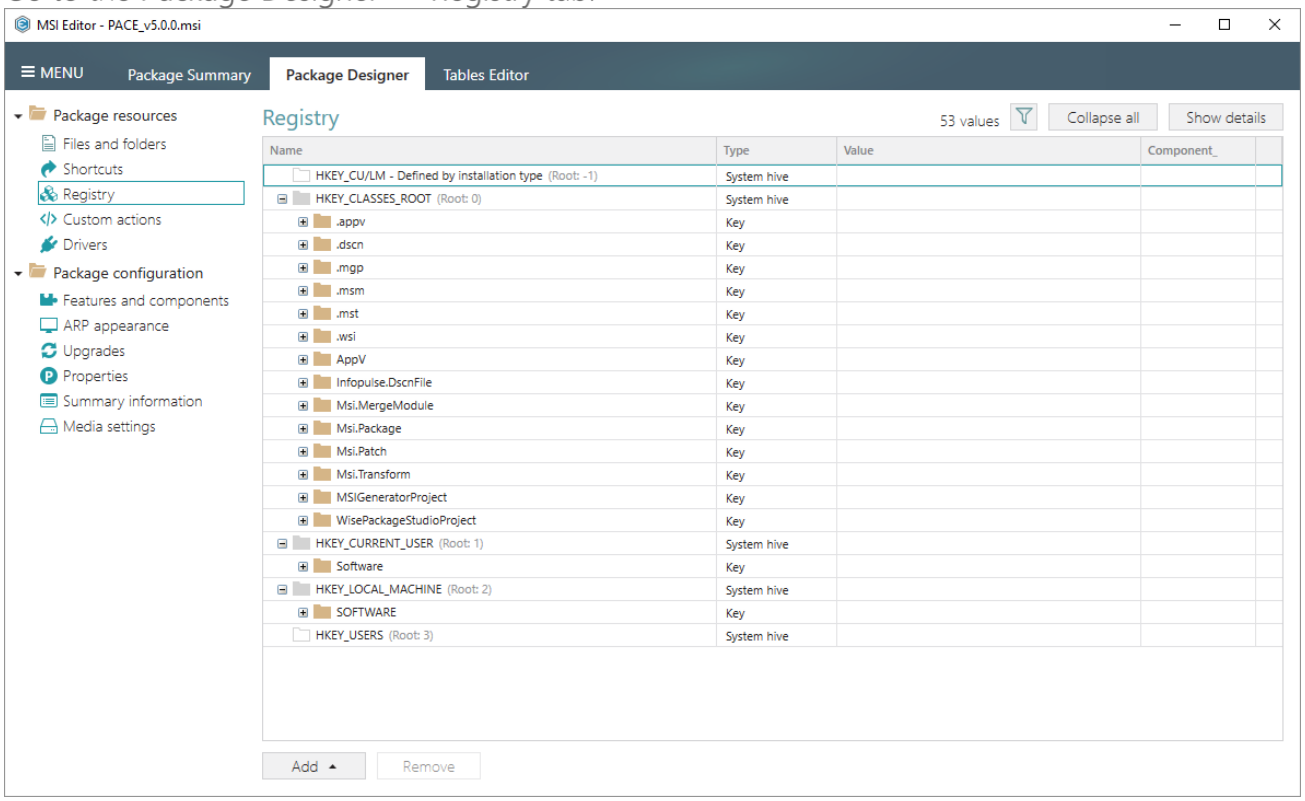
- For saving your package, select Save from the MENU.



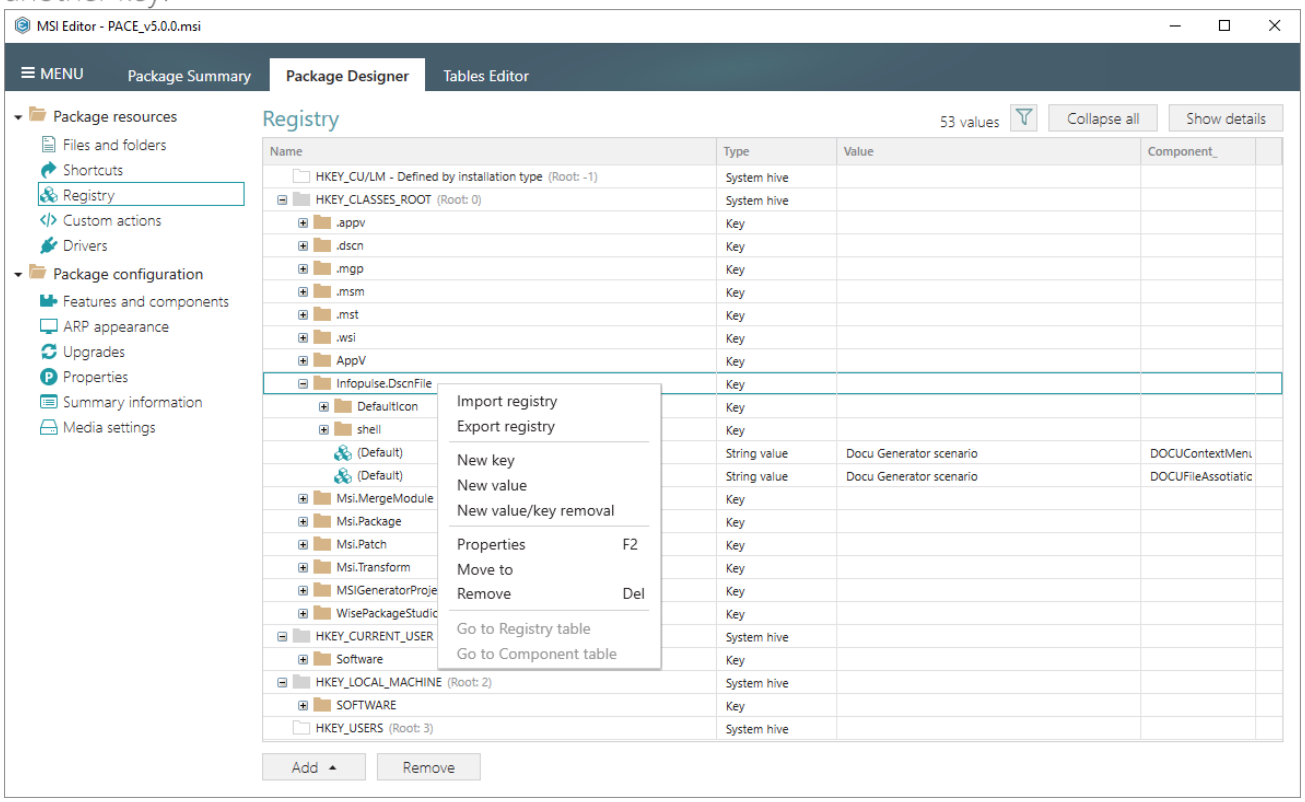
3.7.6.8 Move Value, Key

Move a registry value or a key from one parent key to another one.

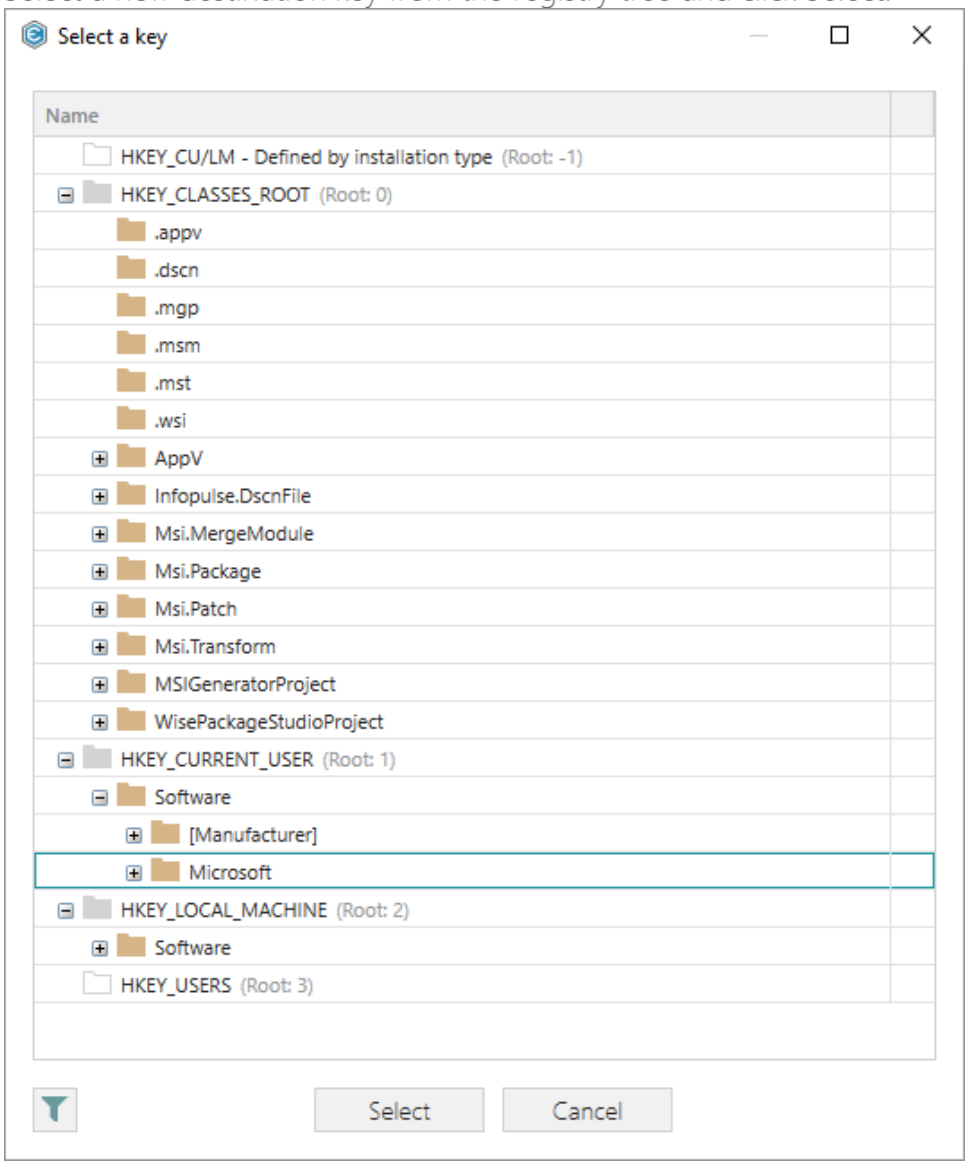
- Go to the Package Designer -> Registry tab.



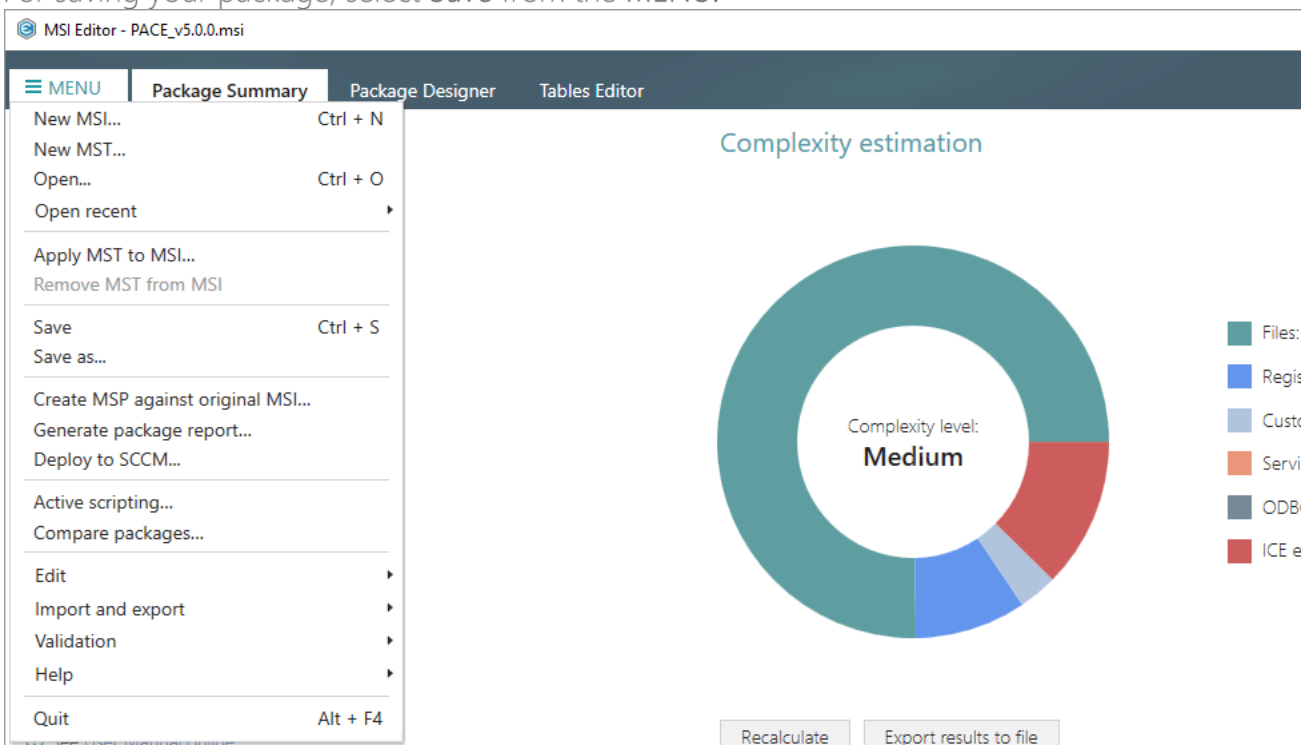
- Select Move to from the context menu of a registry key or a value, which you want to move to another key.



- Select a new destination key from the registry tree and click Select.



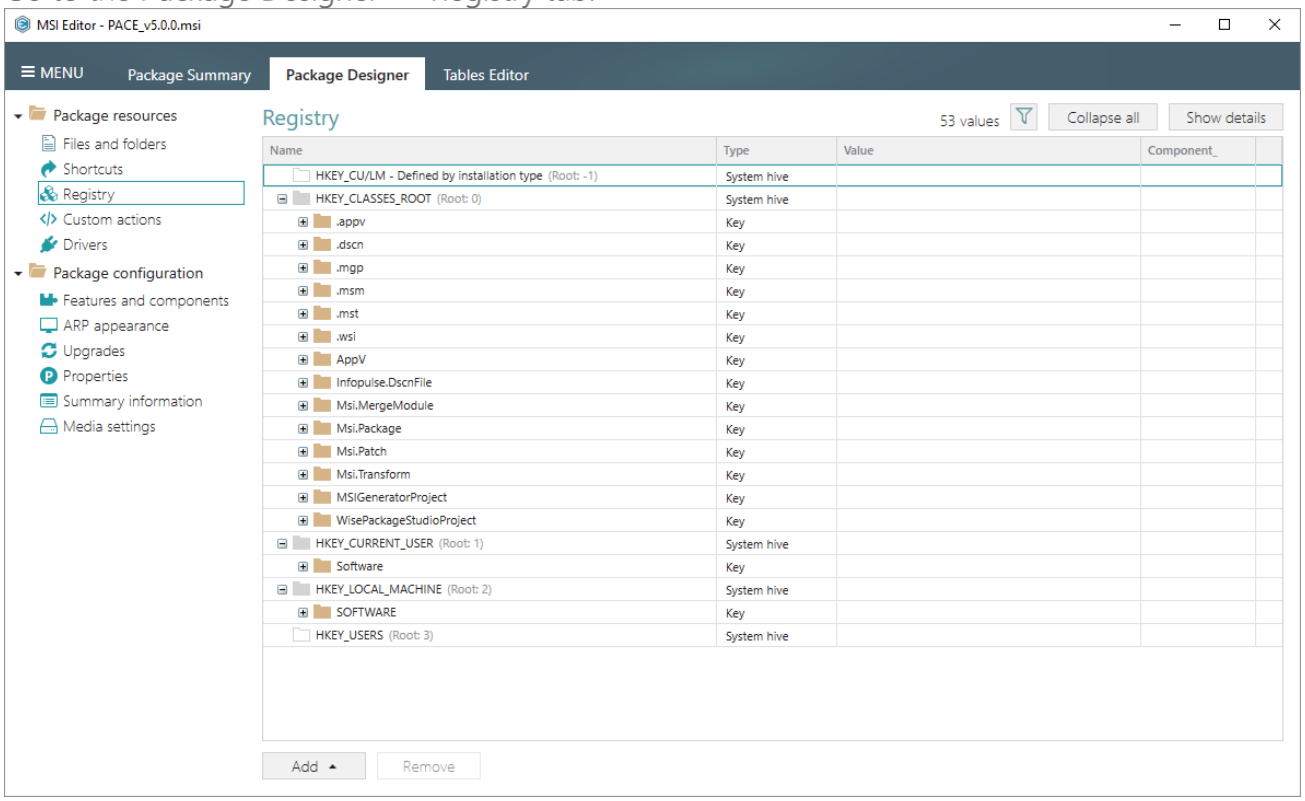
- For saving your package, select Save from the MENU.



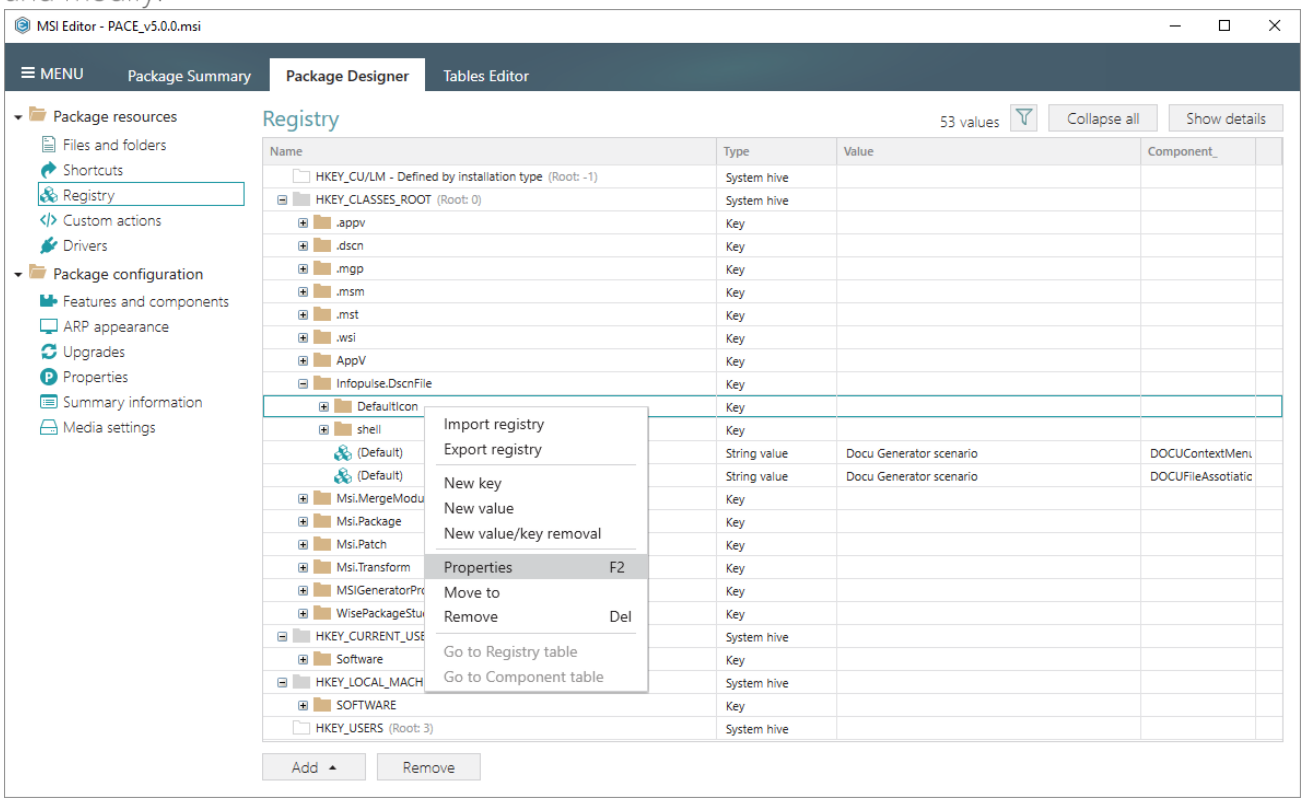
3.7.6.9 Edit Registry Key

Modify details and options of the selected registry key.

- Go to the **Package Designer** -> **Registry** tab.

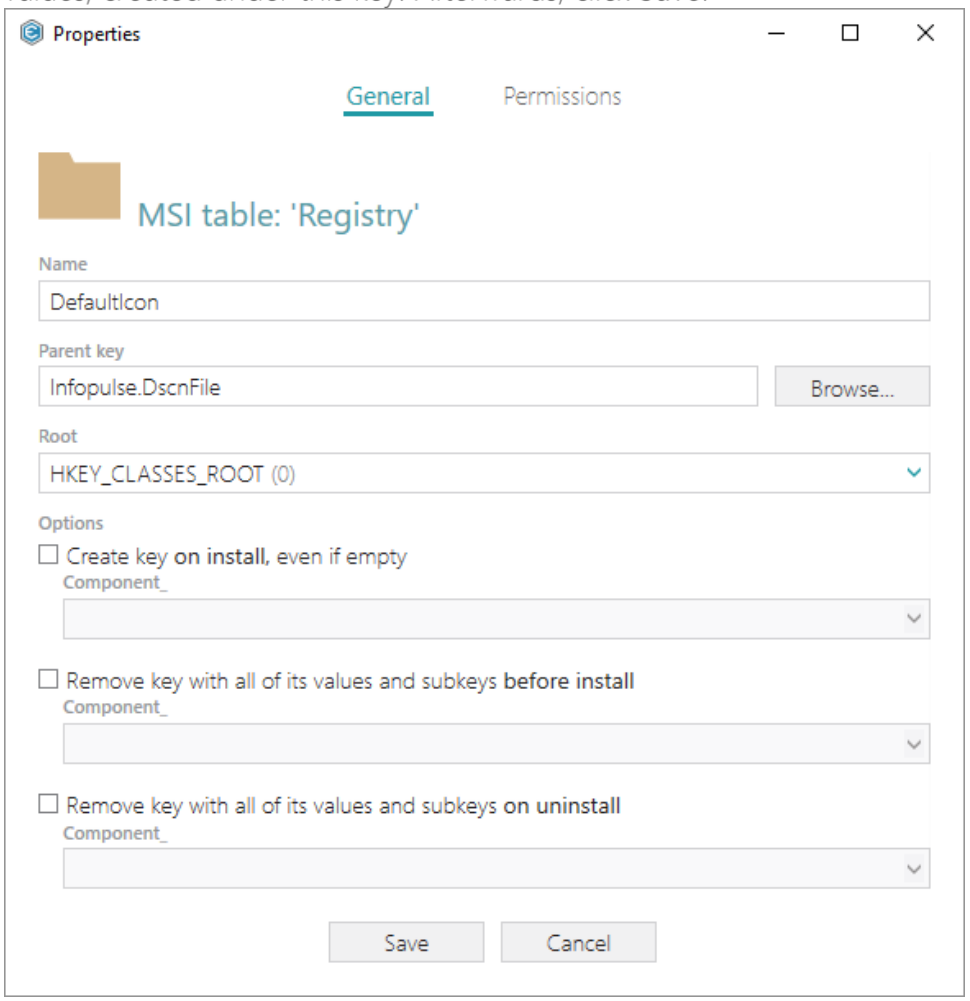


- Select **Properties** from the context menu of a registry key, details of which you want to review and modify.

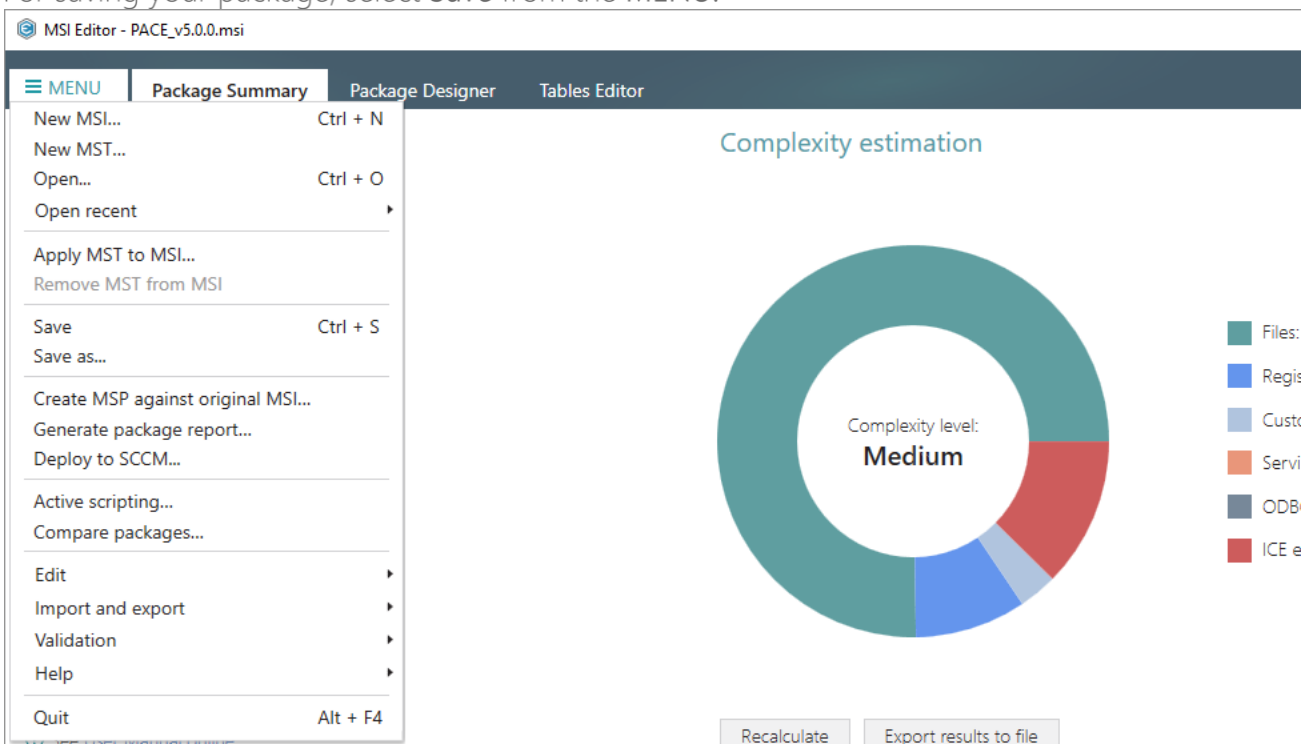


- Using the **Properties** window make the necessary changes to the key **Name**, its **Parent key** and **Root** hive. You can modify the **Parent key** directly in the text field or use **Browse** to choose a new key from the registry tree. Note that changing the **Parent key** and/or the **Root** hive will result in moving the selected key to a new location.

Select the necessary key Options (e.g. 'Create key on install, even if empty') to secure the key creation and/or removal. The selected key Options will be represented as special registry values, created under this key. Afterwards, click Save.



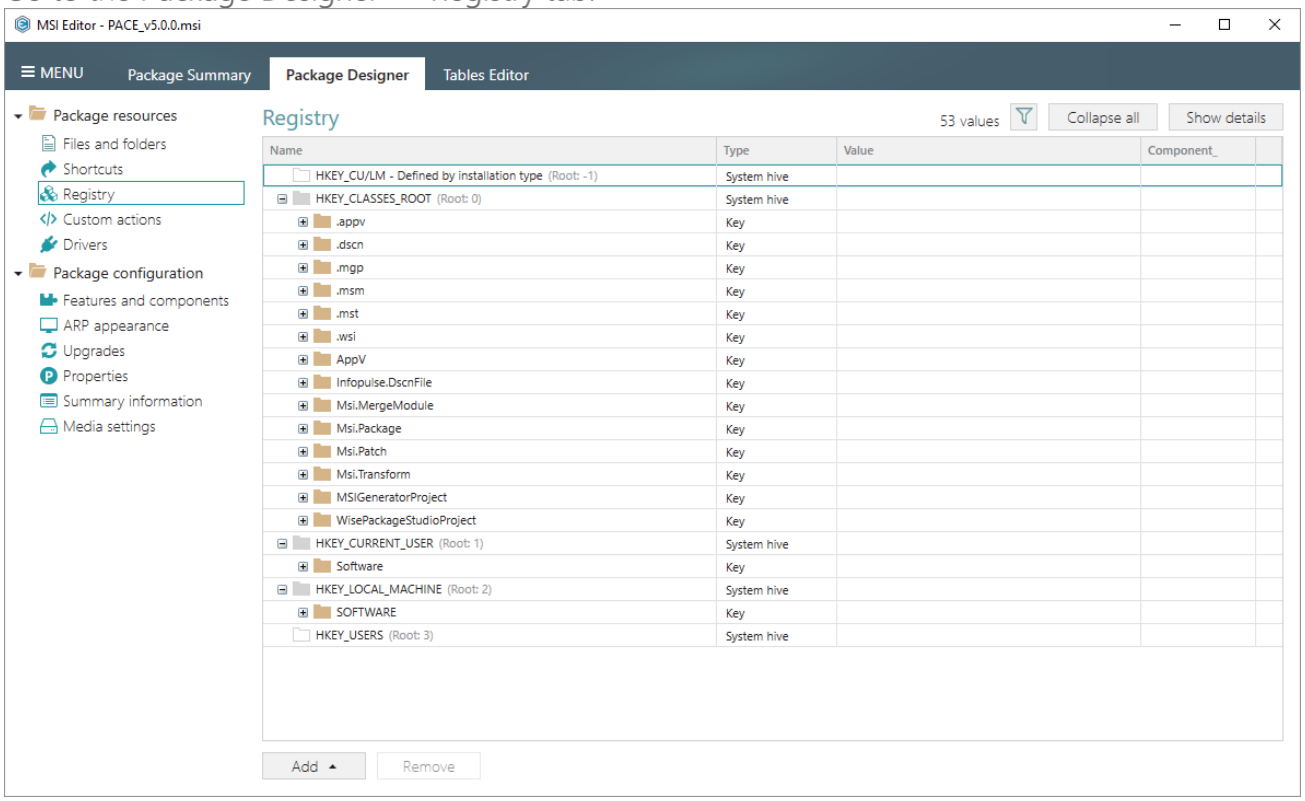
- For saving your package, select Save from the MENU.



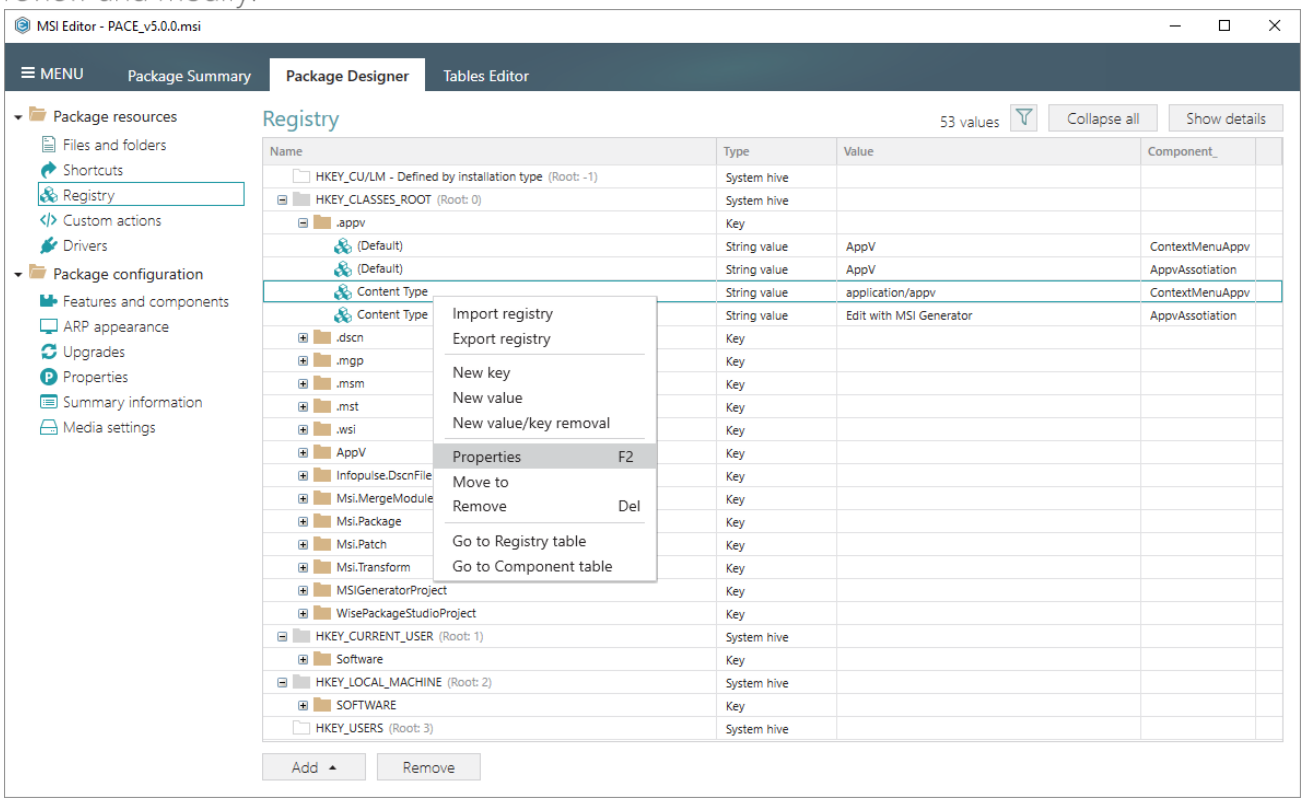
3.7.6.10 Edit Registry Value

Modify details of the selected registry value.

- Go to the Package Designer -> Registry tab.



- Select Properties from the context menu of a registry value, details of which you want to review and modify.



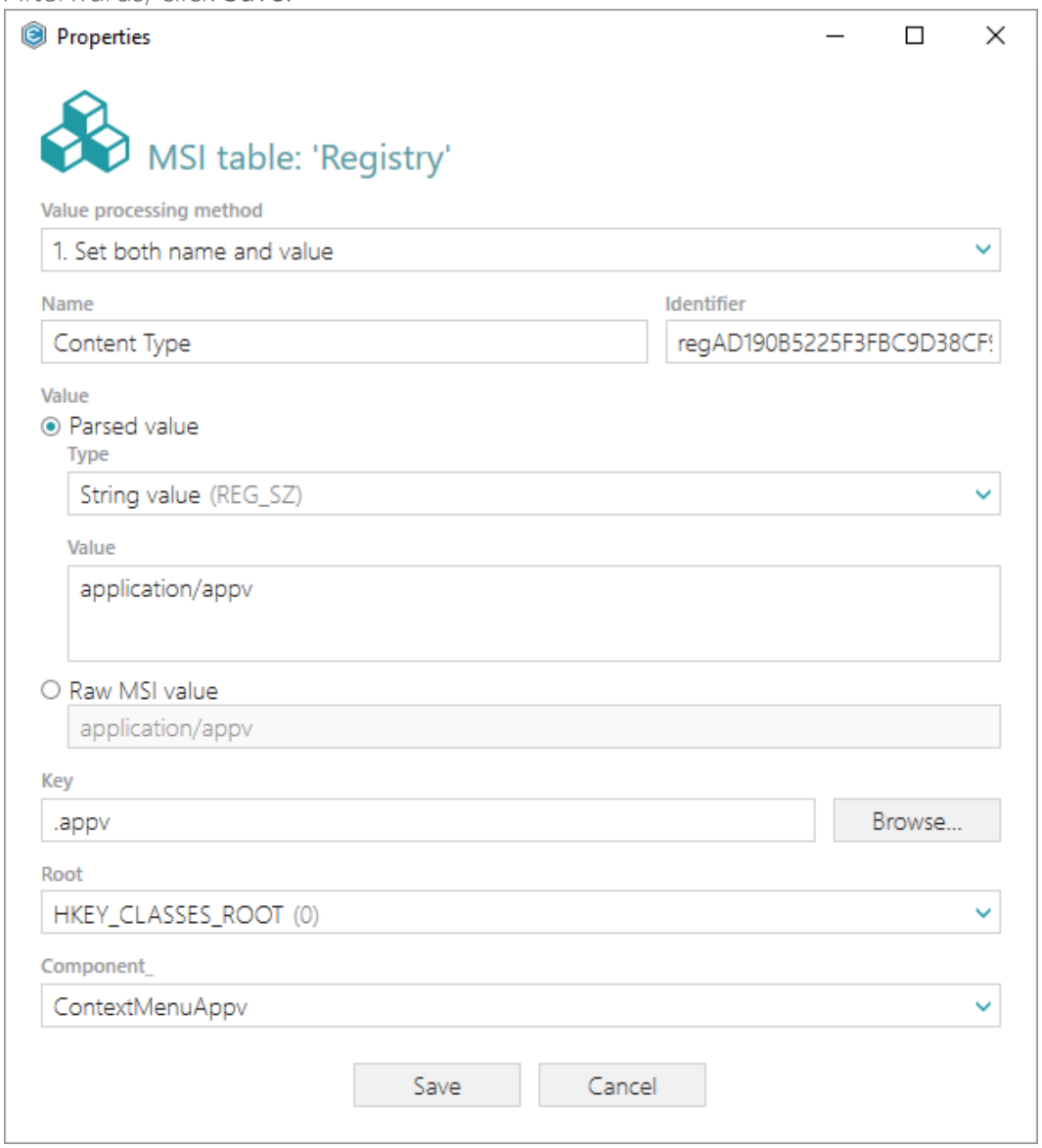
- Using the Properties window make the necessary changes to the value Name, Value itself, and Component that controls the installation of this registry value.

  The Parsed value option serves easier editing of a registry value of the selected value type. The Raw MSI value option displays and gives ability to manage what exactly will be saved to the opened MSI database.

  Note that changing a Key and a Root hive will result in moving this registry value to a new location. You can modify the Key directly in the text field or use Browse to choose a new key from the registry tree.

  The Value processing method options defines whether this value will be an ordinary registry value with the specified Name and Value or will secure a key creation and/or removal. Below, in the tables, you will find the detailed description of these options.

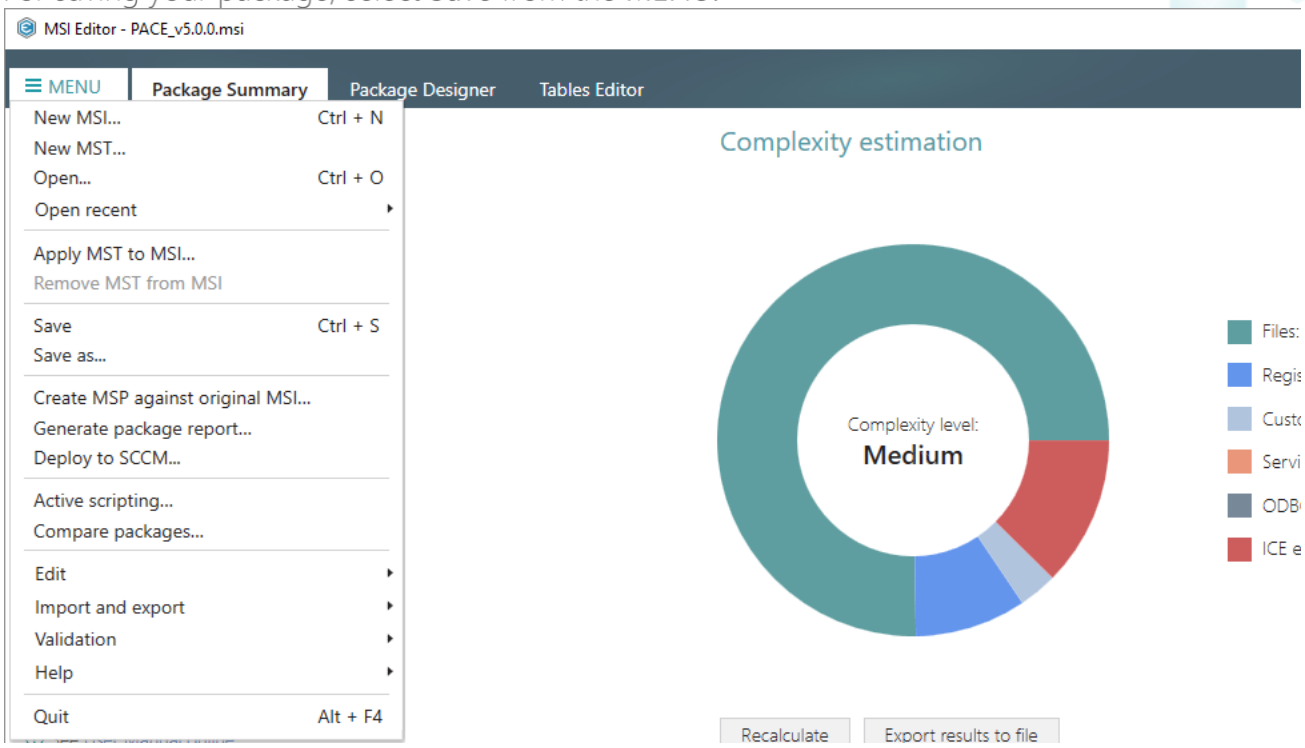
  Afterwards, click Save.



Value processing method	
1. Set both name and value	On install, the package creates an ordinary registry value with the specified name and value itself in the system registry at the selected root and key. On uninstall, the package removes this registry value from the system registry. The containing registry key will be automatically removed only if it is empty.
2. Set <Default> key value	On install, the package creates the Default key value with the specified value in the system registry at the selected root and key. In the system registry, this value is represented as "(Default)": On uninstall, the package cleans-up the Default key registry value in the system registry. The containing registry key will be automatically removed only if it is empty.
3. Create the key on install, even if empty	On install, the package creates the specified key in the system registry at the selected root, even if this key is empty. By default, empty keys are not created. On uninstall, no additional actions take place.
4. Remove the key with all of its values and subkeys on uninstall	On install, no additional actions take place. On uninstall, the package removes this key completely along with all of its values and subkeys.
5. Both 3 and 4	On install, the package creates the specified key in the system registry at the selected root, even if this key is empty. By default, empty keys are not created. On uninstall, the package removes this key completely along with all of its values and subkeys.

NOTE For removing registry values or keys from the system registry before install, add the Value/key removal item, described in section 3.7.6.3

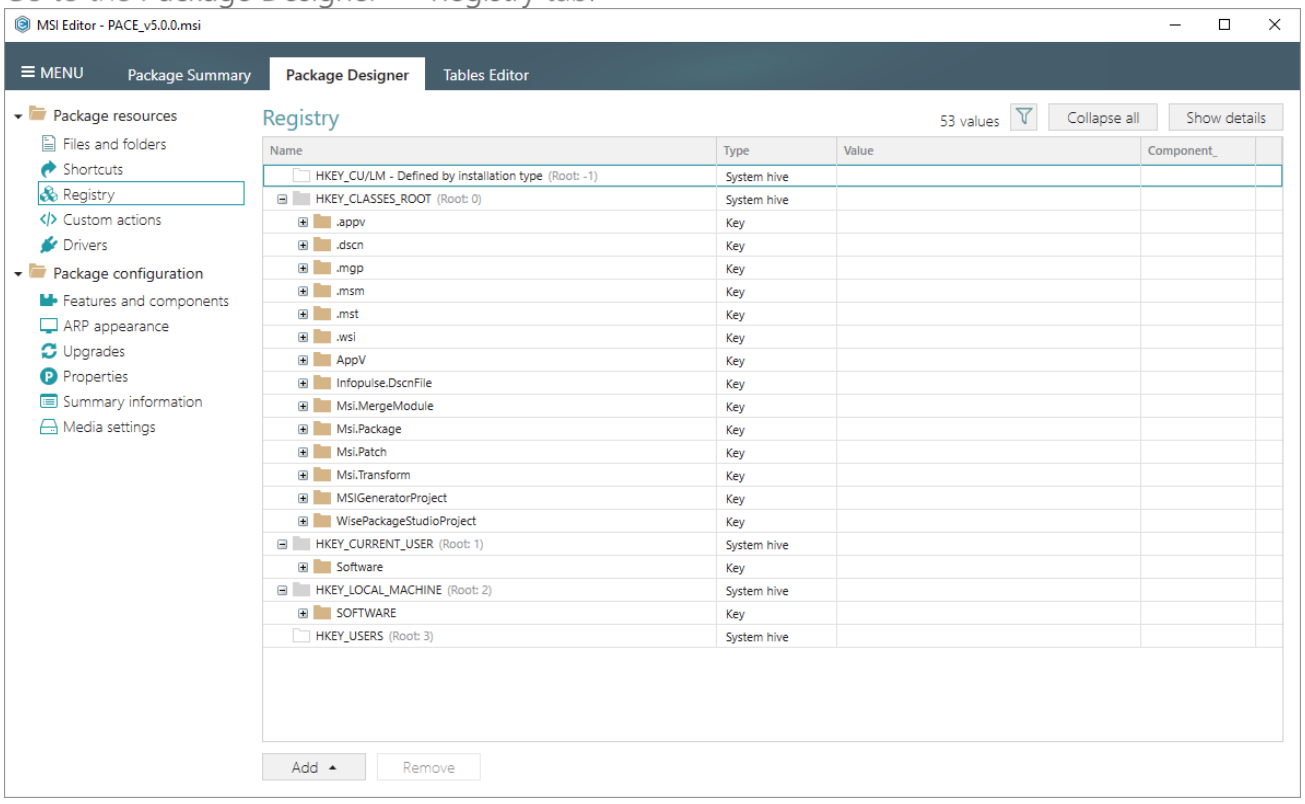
- For saving your package, select Save from the MENU.



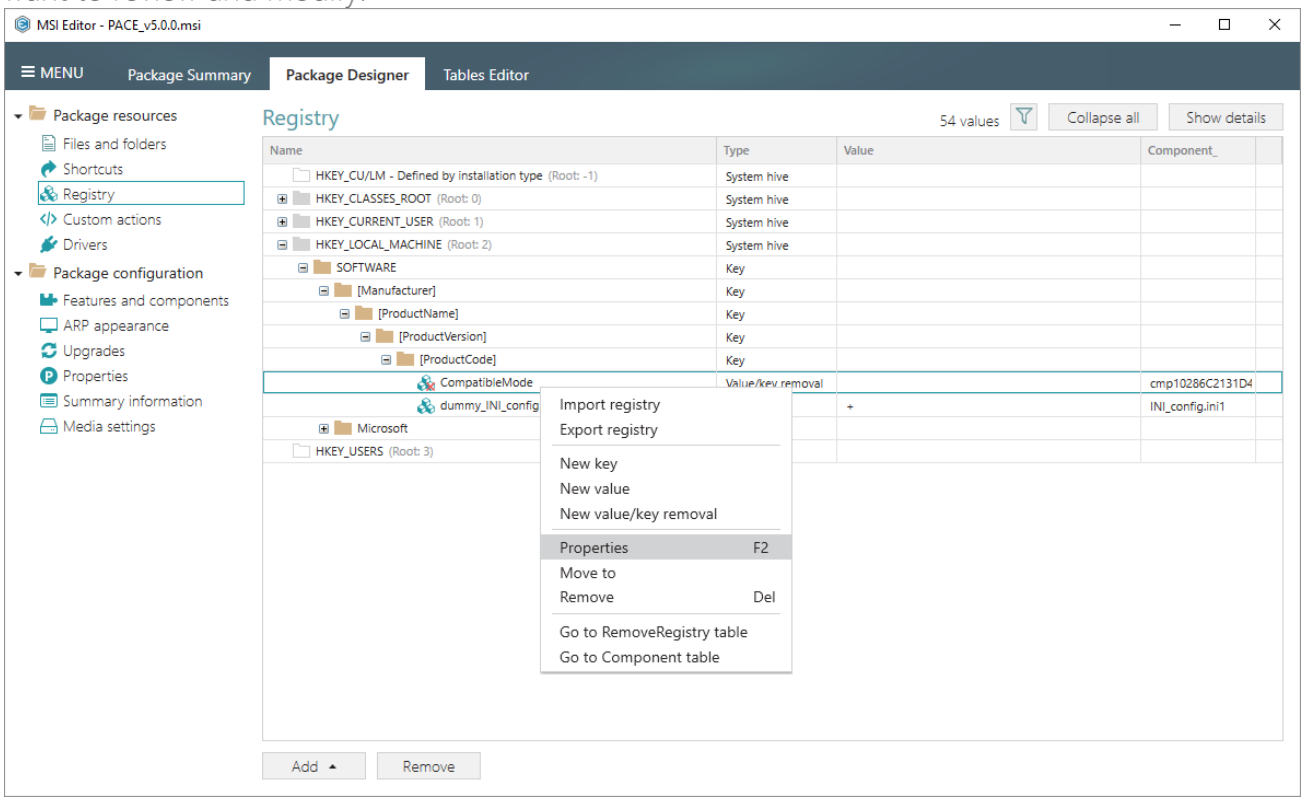
3.7.6.11 Edit Value/Key Removal

Modify details of the selected value/key removal item (RemoveRegistry) to secure a value or a key removal during the package installation.

- Go to the Package Designer -> Registry tab.

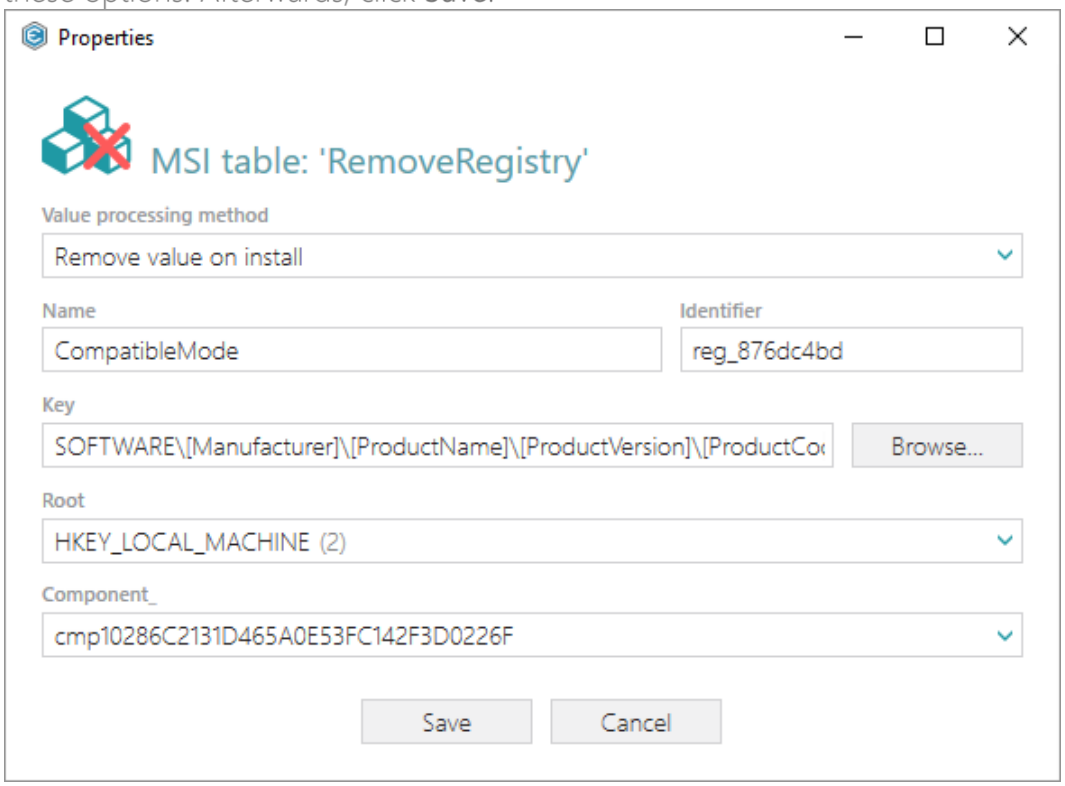


- Select **Properties** from the context menu of a 'Value/key removal' item, details of which you want to review and modify.



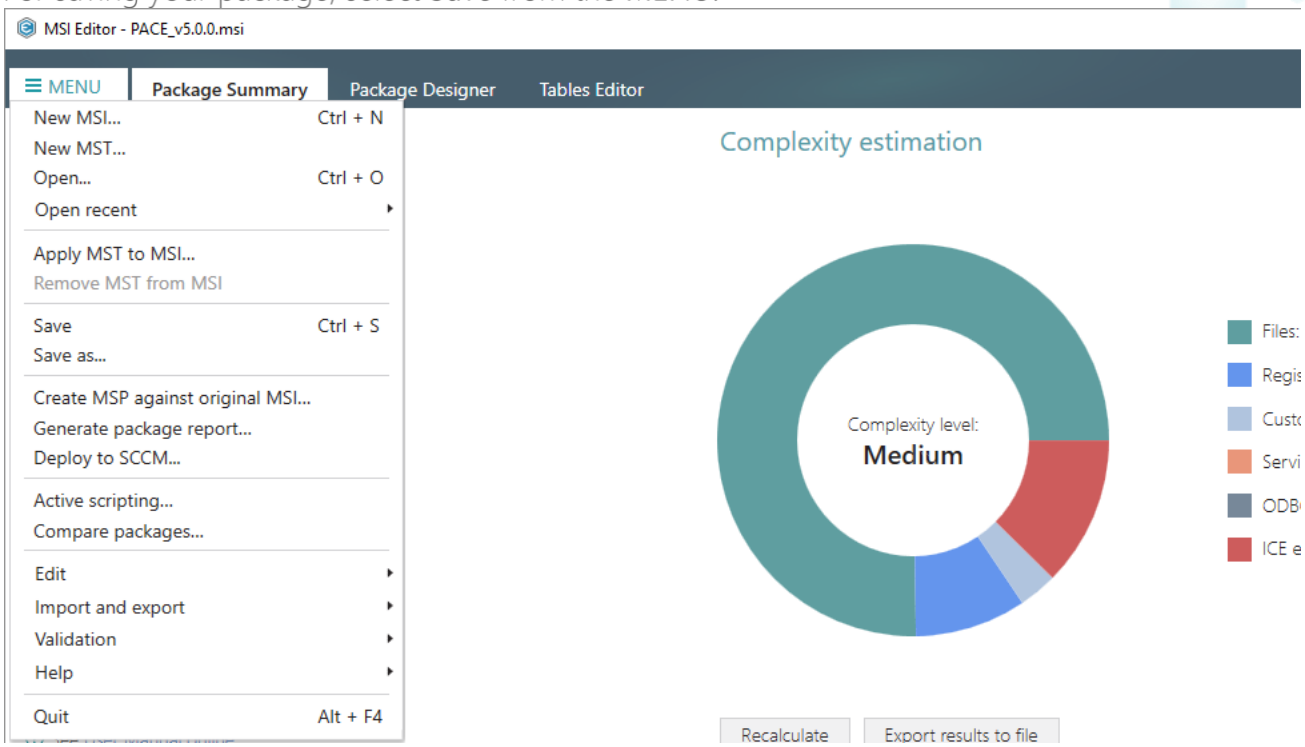
- Using the **Properties** window make the necessary changes to a value **Name**, its **Key**, **Root** hive, and **Component** that controls the deletion of this registry value. Note that the specified registry value will be removed from the system registry at the selected root and key during the package installation, right before new registry values are created.

  The **Value processing method** options defines what will be removed: an ordinary value, a Default key value, or whole key. Below, in the tables, you will find the detailed description of these options. Afterwards, click **Save**.



Value processing method	
Remove value on install	On install, right before new registry values are created, the package removes an ordinary registry value of the specified name from the system registry at the selected root and key. On uninstall, no additional actions take place.
Clean-up <Default> key value on install	On install, right before new registry values are created, the package cleans-up the Default key value with the specified value in the system registry at the selected root and key. In the system registry, this value is represented as "(Default)": Name: (Default) | Type: REG_SZ | Data: (value not set) On uninstall, no additional actions take place.
Remove the key with all of its values and subkeys on install	On install, right before new registry values are created, the package removes this key completely along with all of its values and subkeys. On uninstall, no additional actions take place.

- For saving your package, select Save from the MENU.



3.7.7 Custom Actions

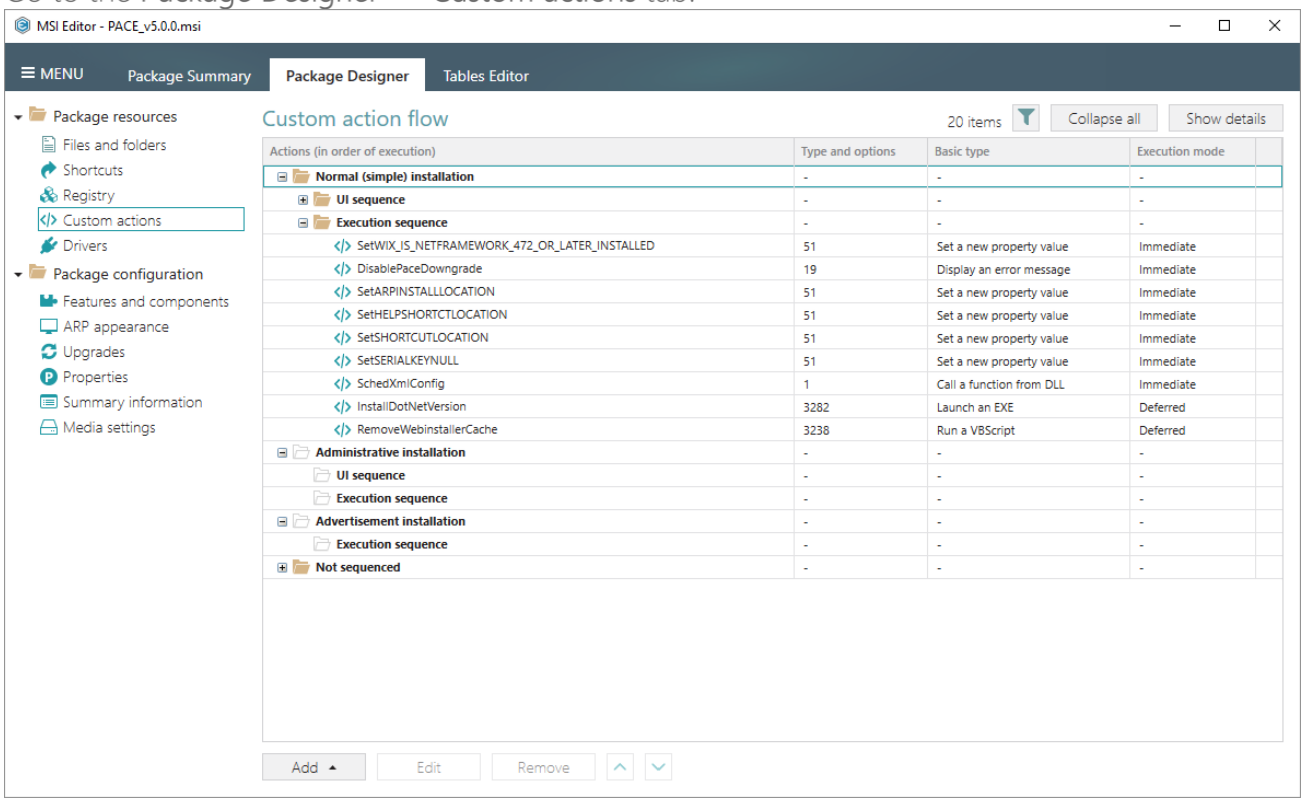
Choose a scenario that better suits your needs:

- Add Standard Actions, described in section 3.7.7.1
  Add new standard actions to the necessary sequences.
- Add Custom Action, described in section 3.7.7.2
  Add a new custom action with the aim of launching EXE, calling functions from DLL, running VBScript, Jscript, setting a new property value, folder path, and showing an error message.
- Run PowerShell script, described in section 3.7.7.2.1
  Run a PowerShell script from MSI on install or uninstall.
- Edit Custom Action, described in section 3.7.7.3
  Edit basic type, resources, sequence and other settings of the existing custom action.

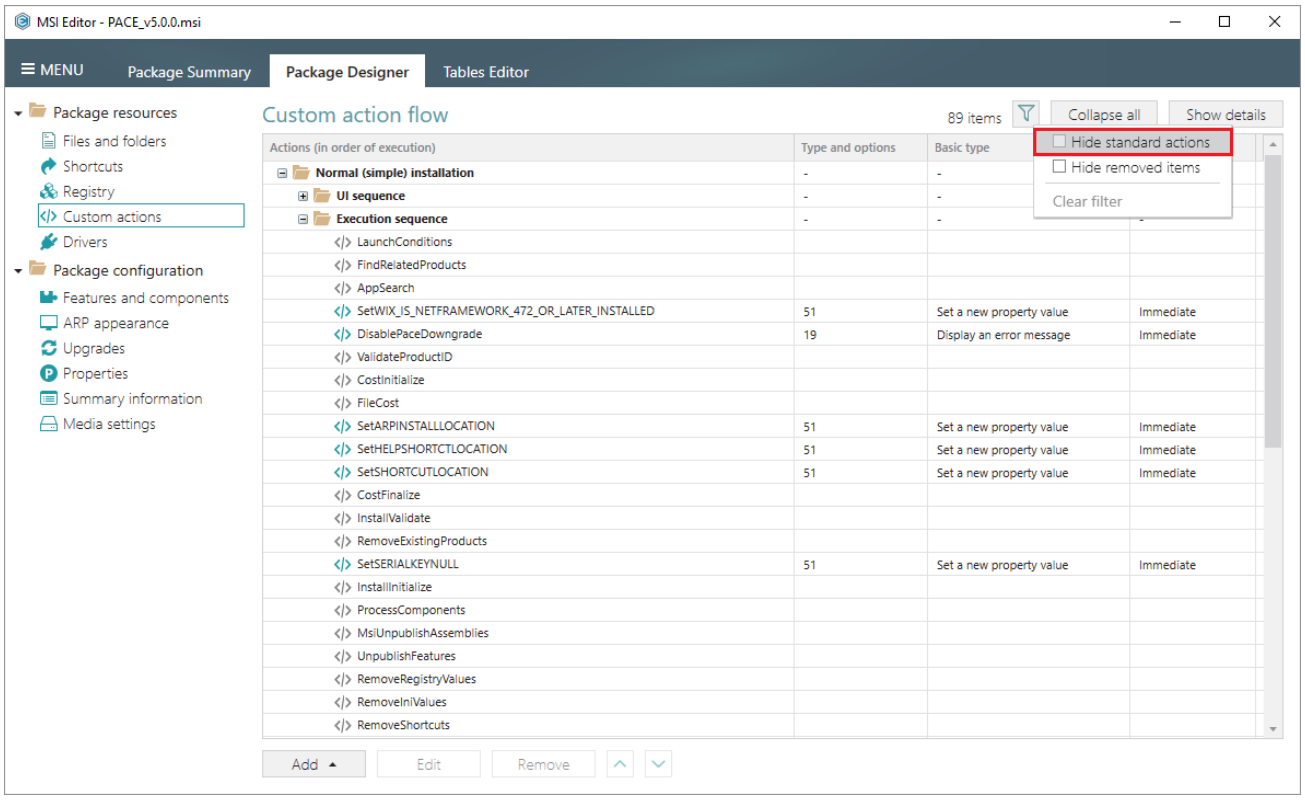
3.7.7.1 Add Standard Actions

Add new standard actions to the necessary sequences of the opened package.

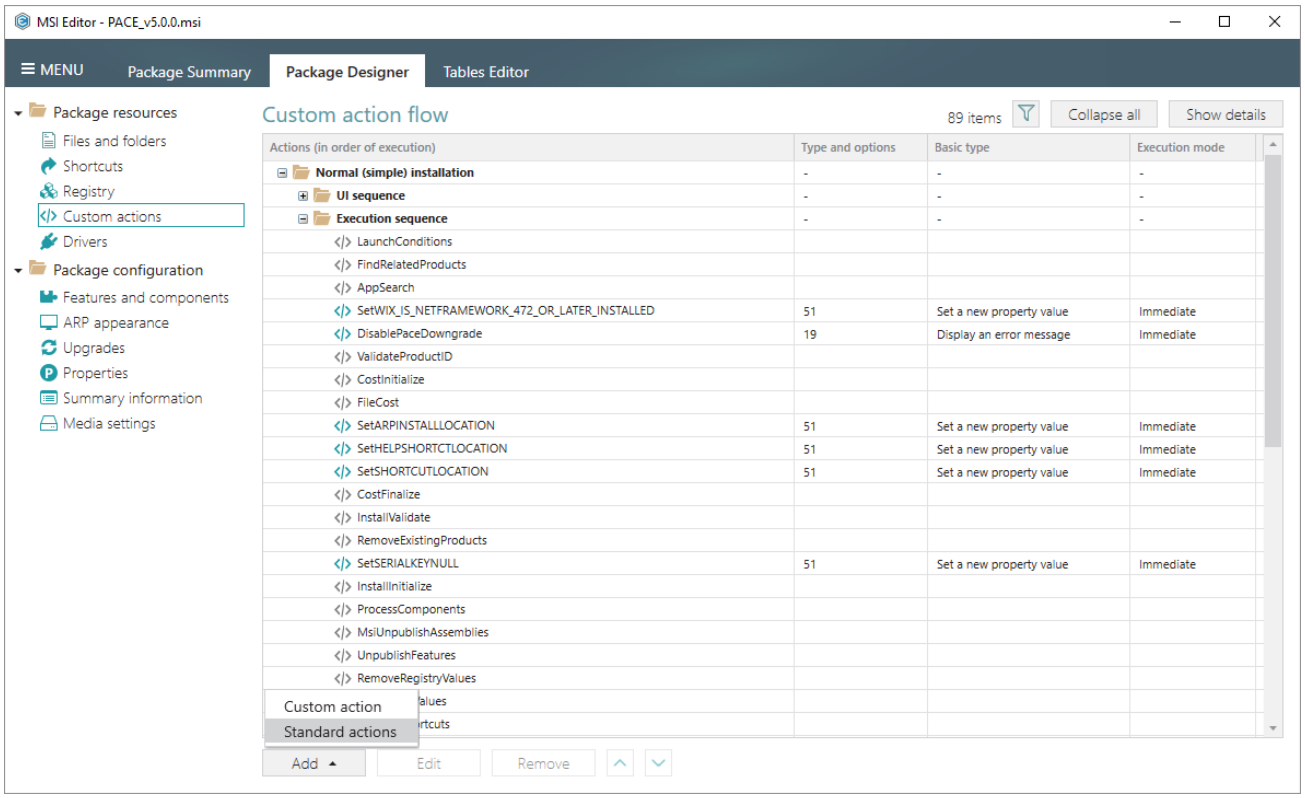
- Go to the Package Designer -> Custom actions tab.



- (Optional) To review the existing standard actions, hidden by default, click on the and untick the '**Hide standard actions**' filter.

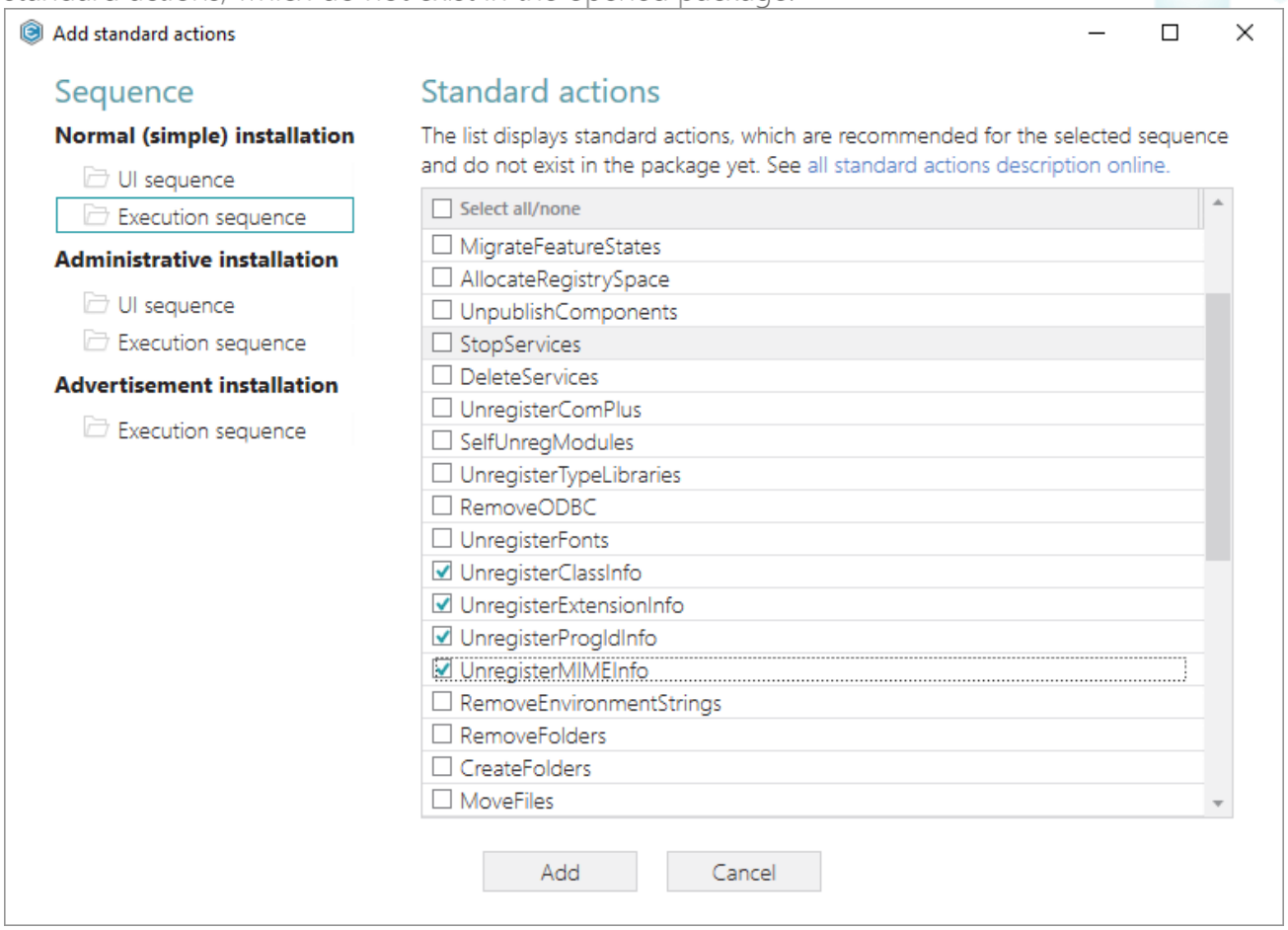


- Select **Add** -> **Standard actions** from the toolbox.

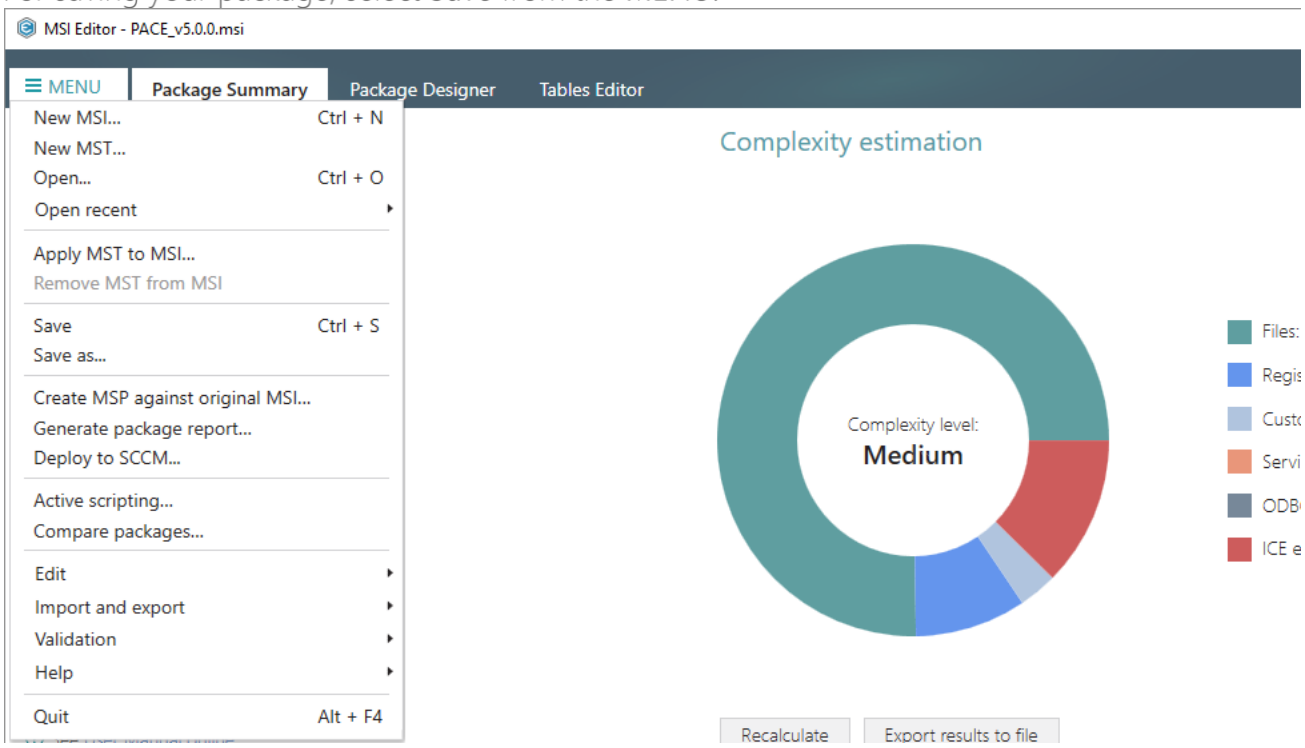


- Select the standard actions, which you want to add, and click **Add**. Note that you can add many actions to the different installation sequences at once. The list displays only those

standard actions, which do not exist in the opened package.



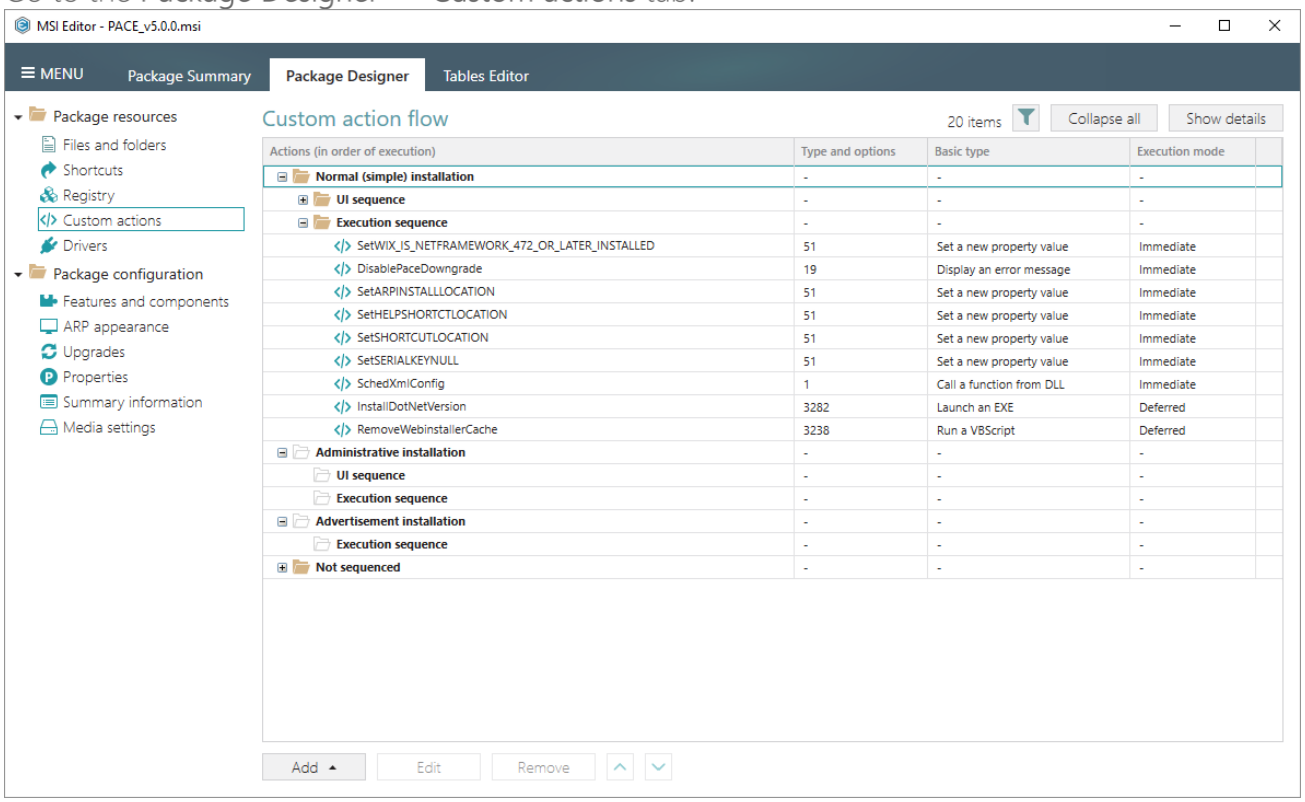
- For saving your package, select Save from the MENU.



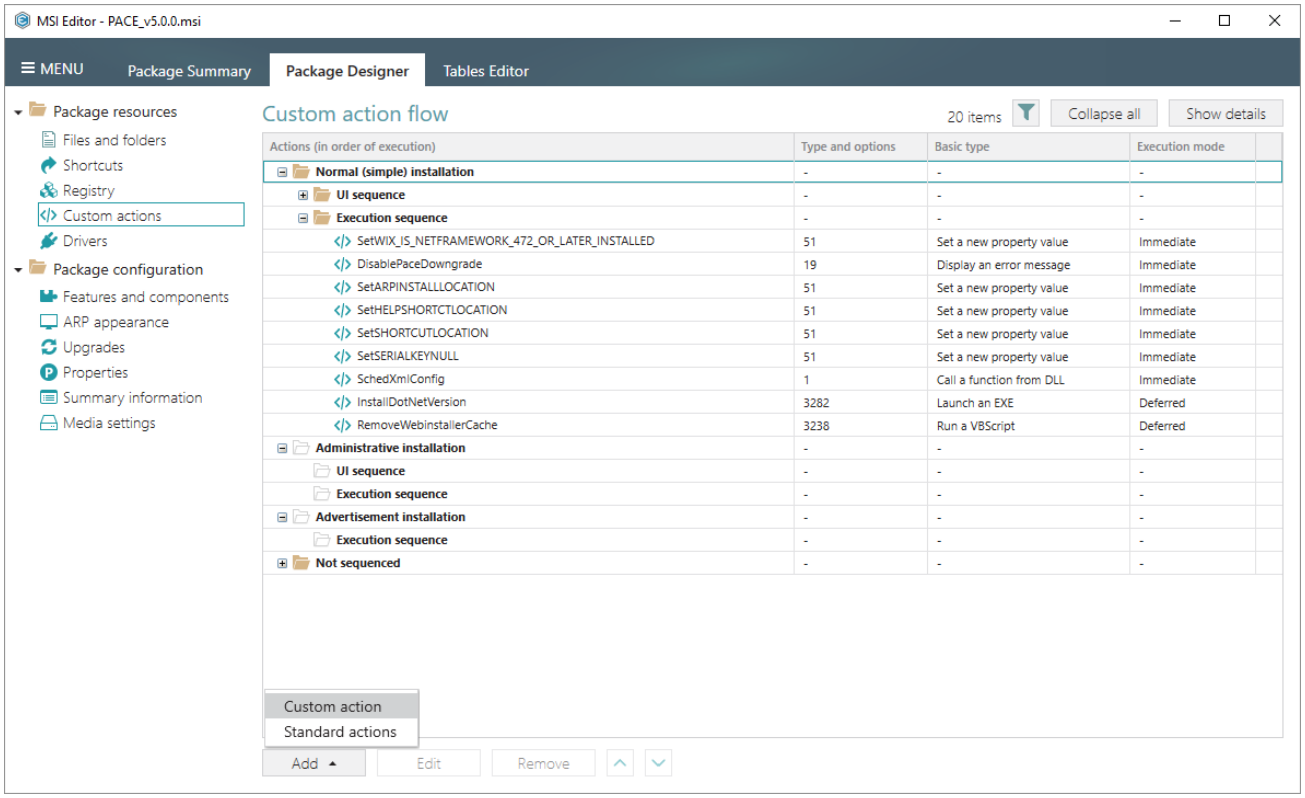
3.7.7.2 Add Custom Action

Add a new custom action with the aim of launching EXE, calling functions from DLL, running VBScript, Jscript, setting a new property value, folder path, and showing an error message.

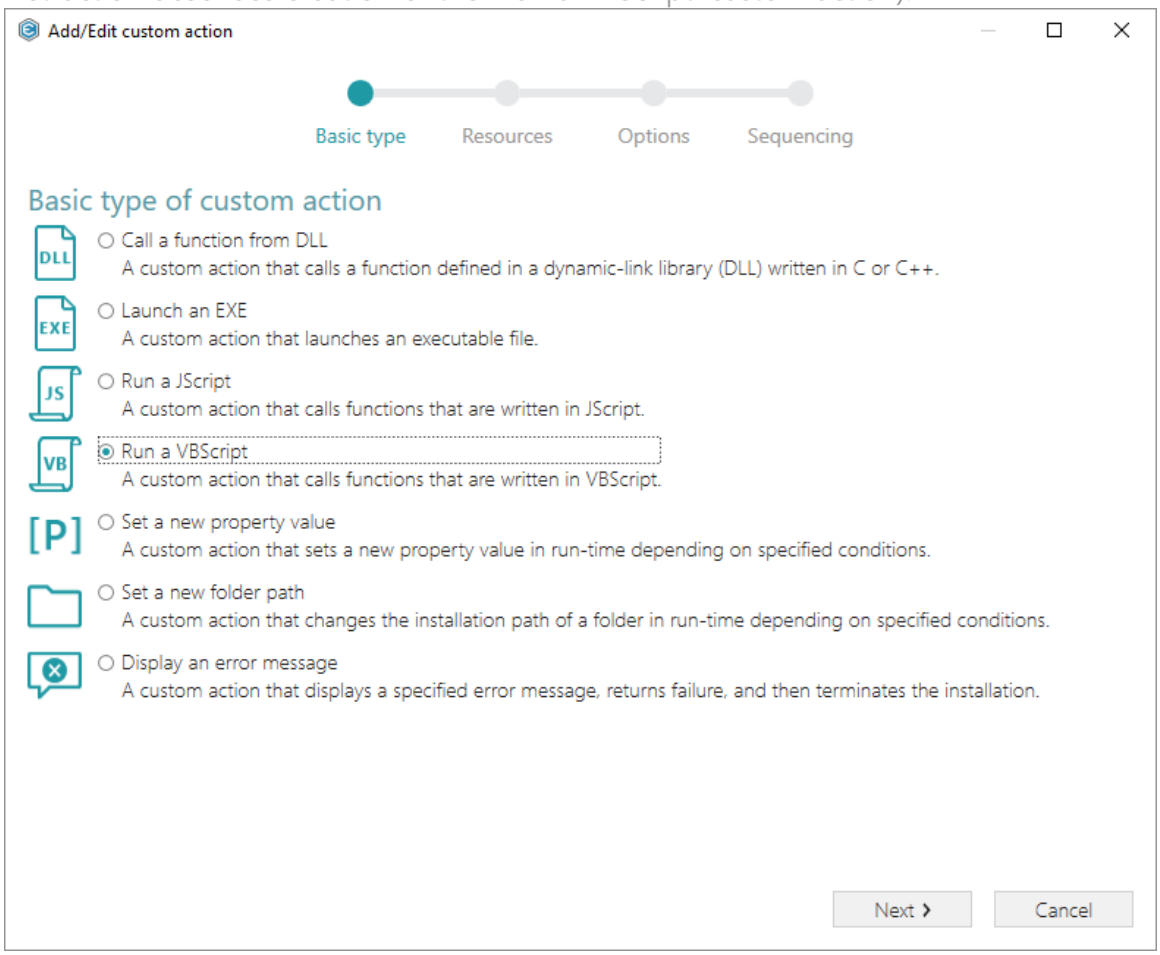
- Go to the Package Designer -> Custom actions tab.



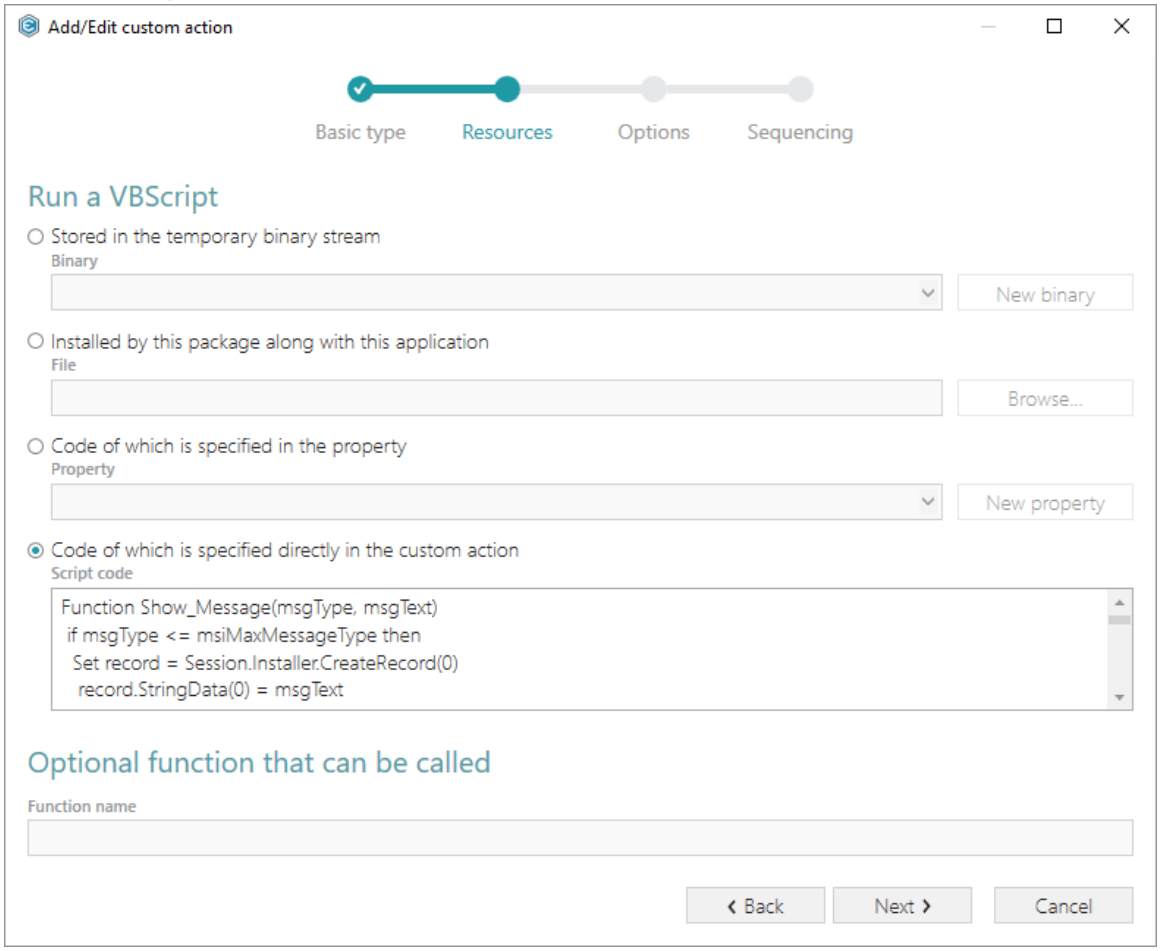
- Select Add -> Custom action from the toolbox.



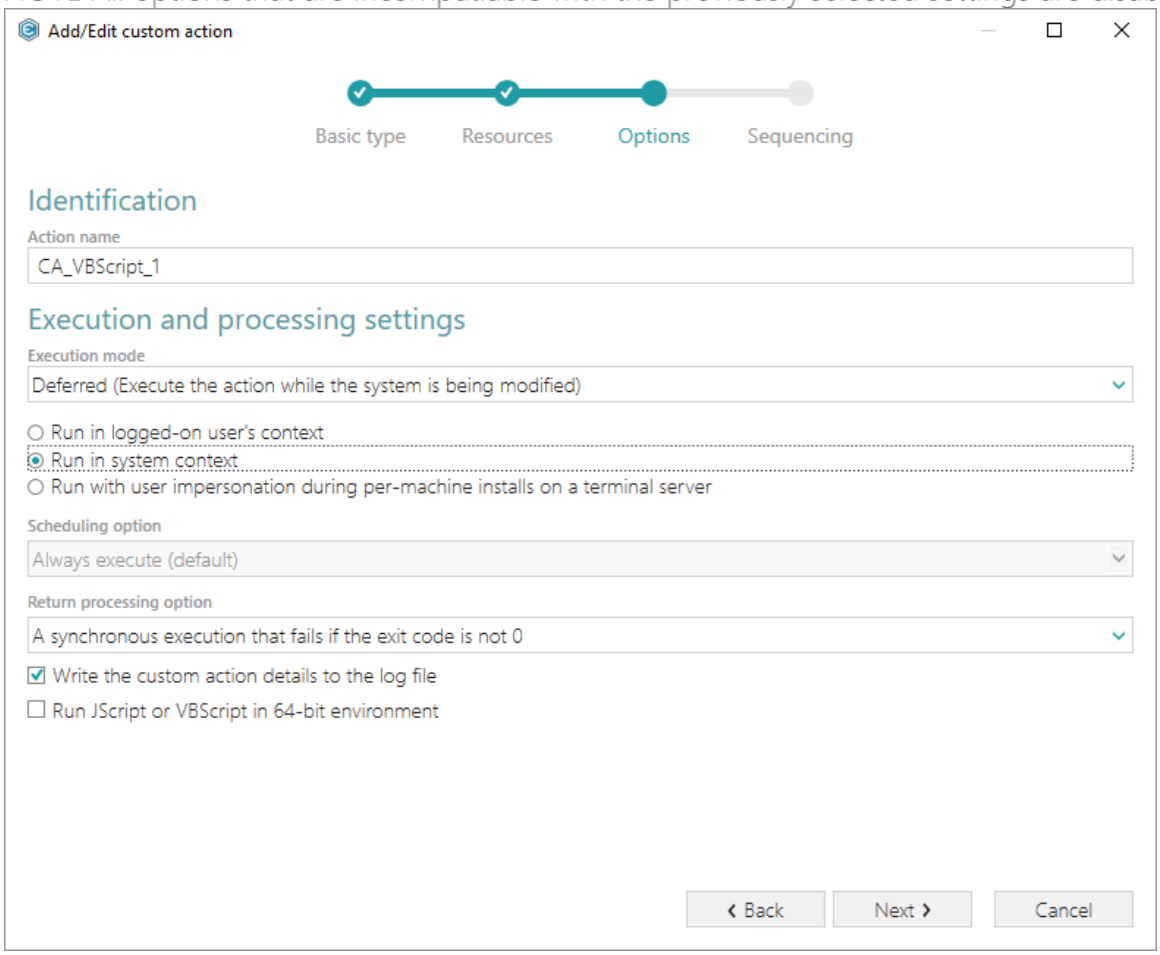
- Select a basic type of your custom action (e.g. Run a VBScript), and click **Next**. (The following instruction describes creation of the 'Run a VBScript' custom action).



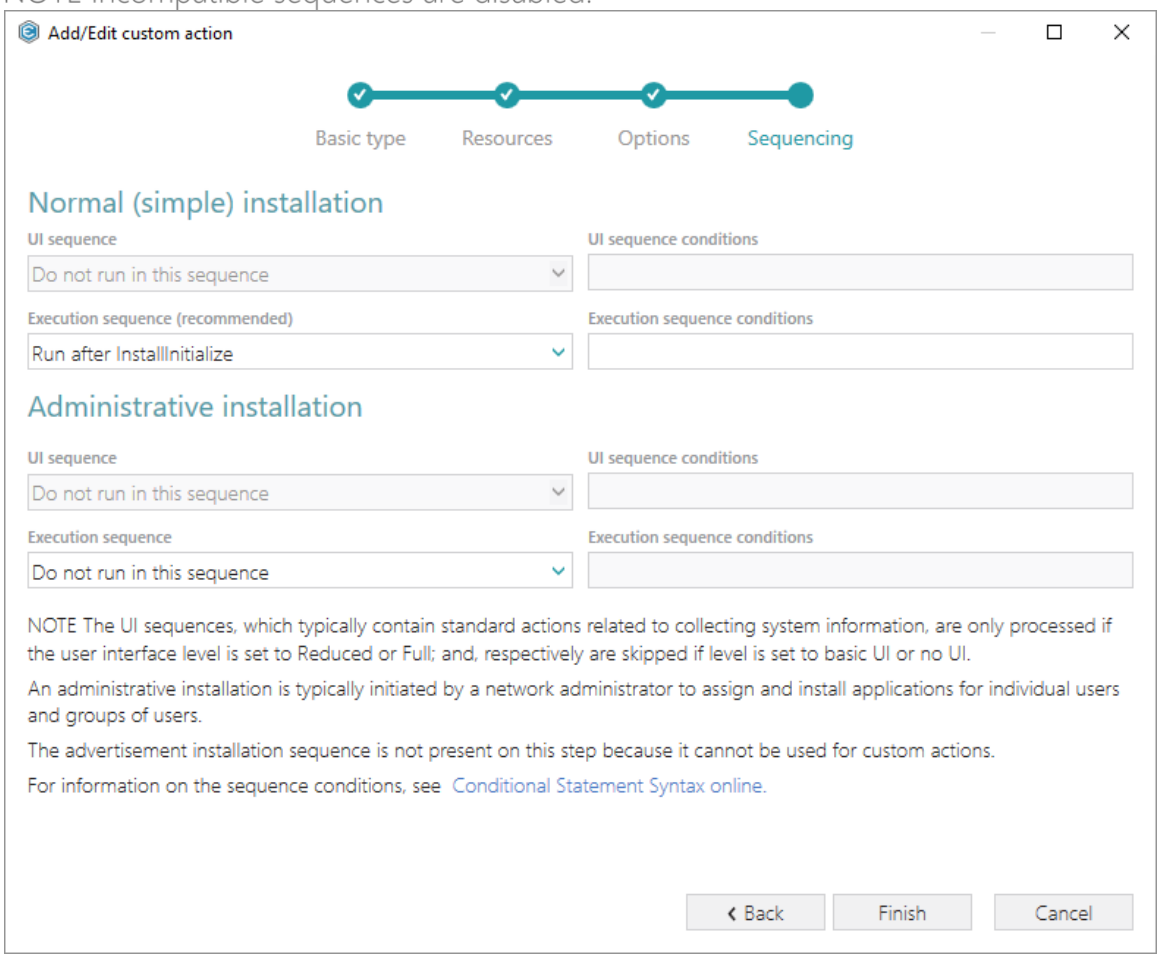
- Specify where the VBScript file is located or embed a script body directly into the custom action. Then, click Next.



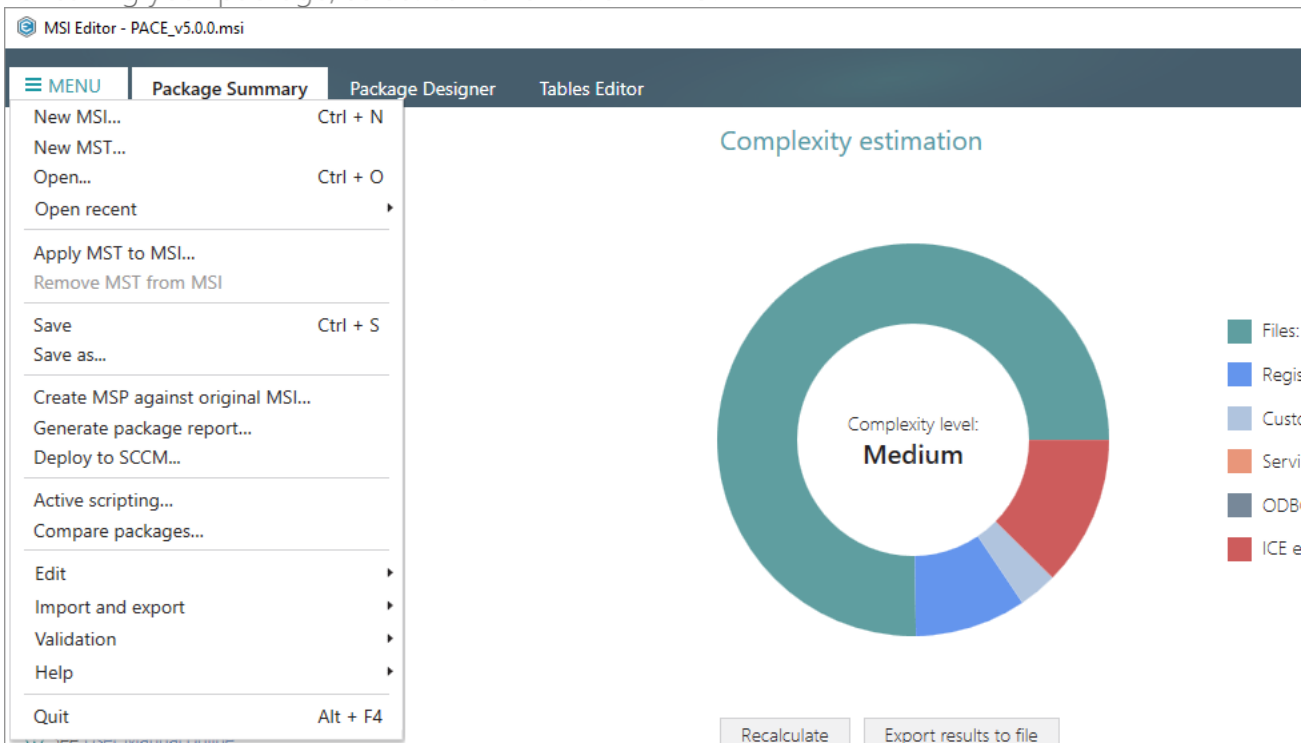
- At the **Options** step you can set an **Action name**, select **Execution mode** (Immediate, Deferred, Rollback and Commit), execution context, scheduling, return processing and other options. Then, click **Next**.
  NOTE All options that are incompatiable with the previously selected settings are disabled.



- The last step allows selecting secuences, where your custom action will be runned. The sequence drop-down list allows you to select only compatible positions, taking into account your previously selected options. Also, you are free to specify execution conditions either by entering them manually or choosing from the drop-down list. Click Finish to add your custom action.
NOTE Incompatible sequences are disabled.



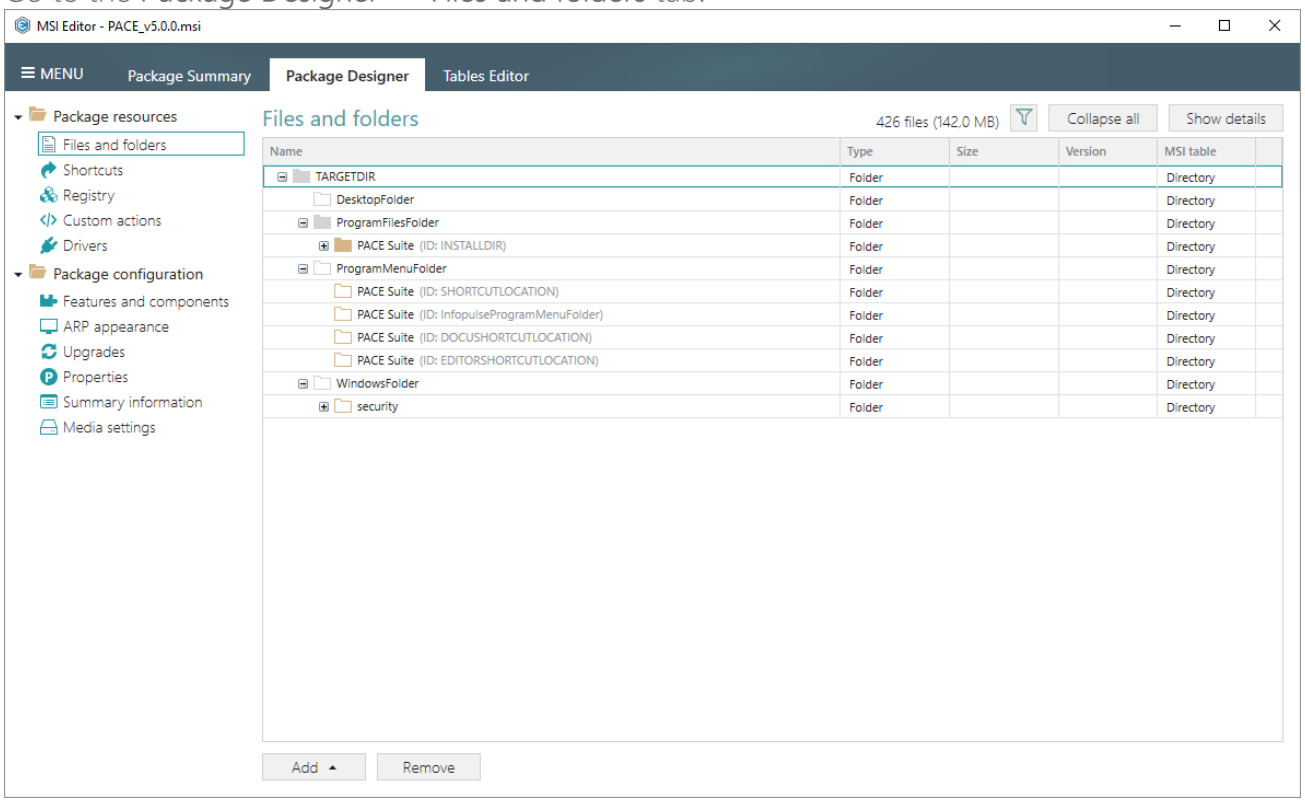
- For saving your package, select Save from the MENU.



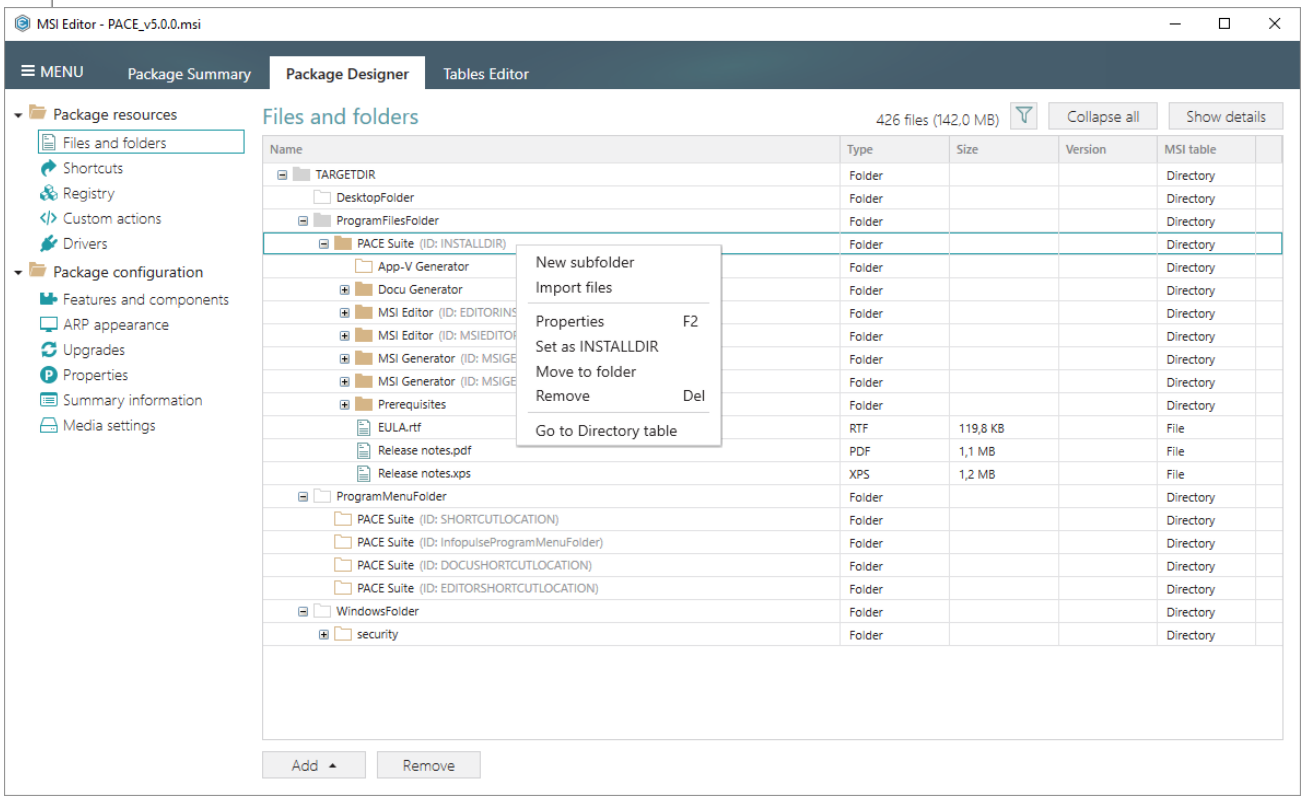
3.7.7.2.1 Run PowerShell script

Run a PowerShell script from MSI on install or uninstall. This solution consists of two parts: (1) importing PowerShell script to the package; (2) adding a Custom action for running this script.

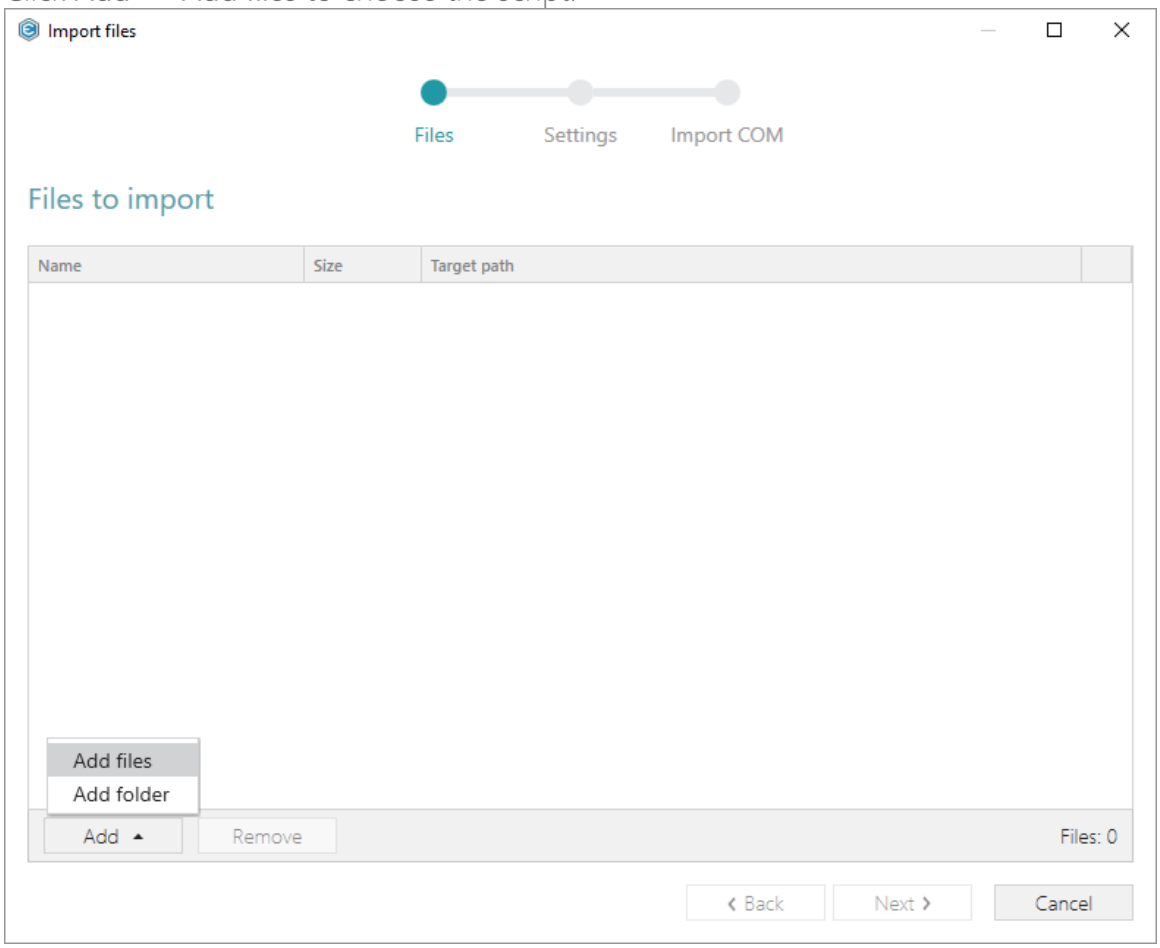
- Go to the Package Designer -> Files and folders tab.



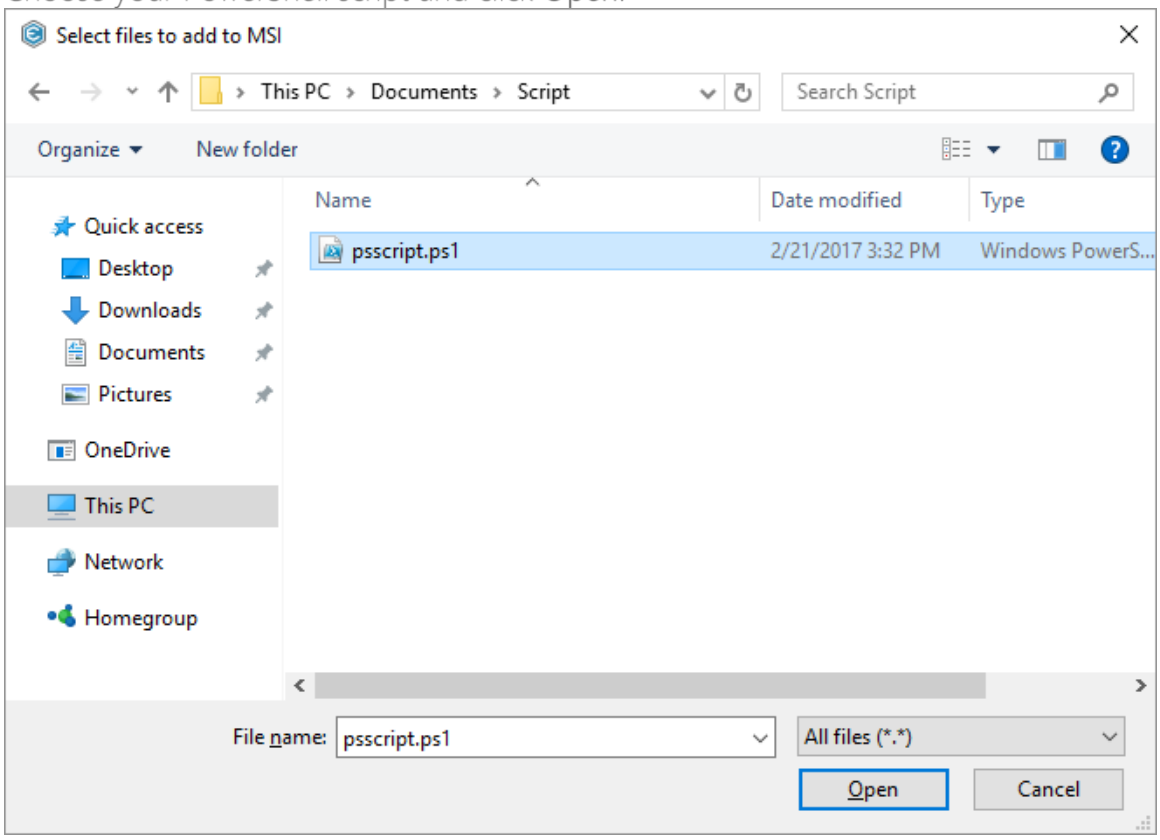
- Select Import files from the context menu of a folder, to which you want to import PowerShell script.



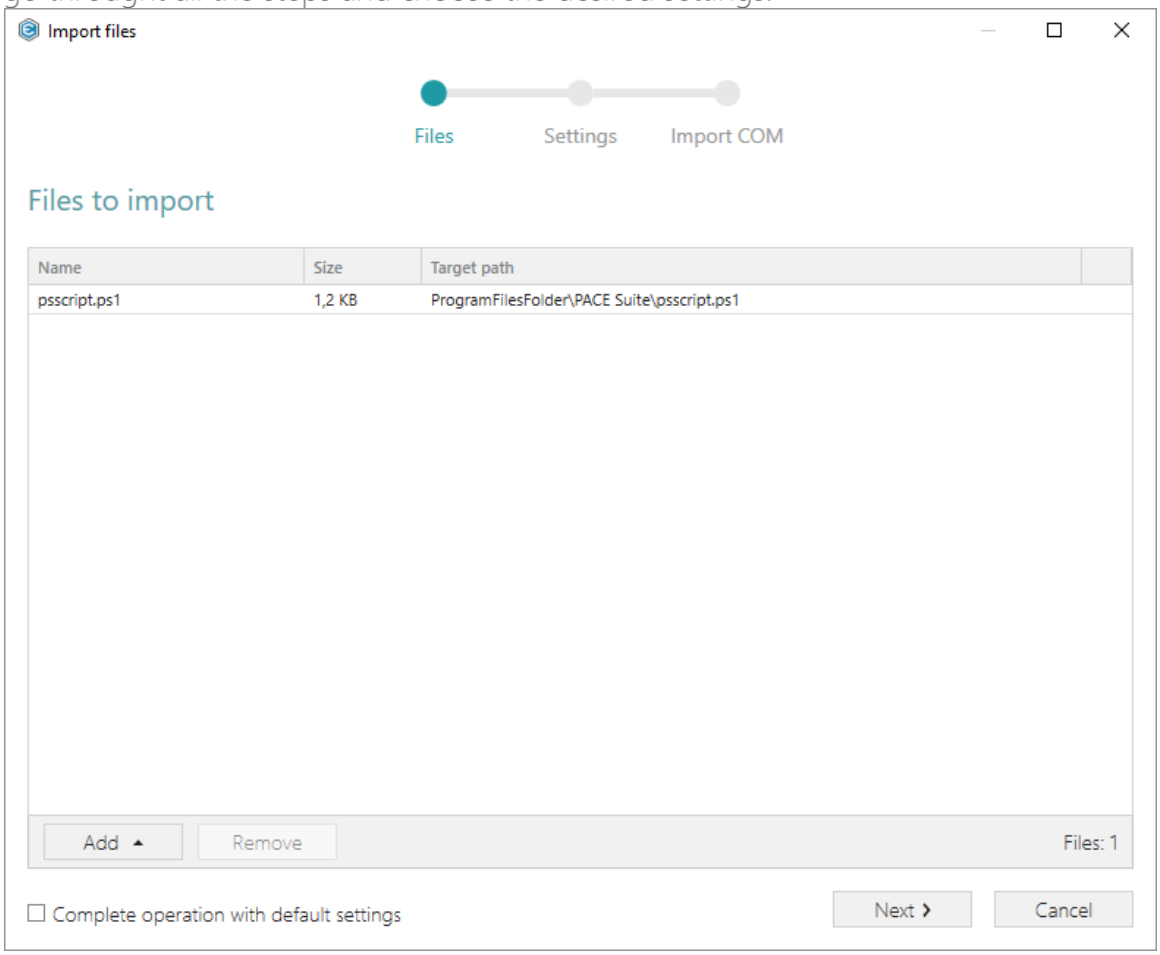
- Click Add -> Add files to choose the script.



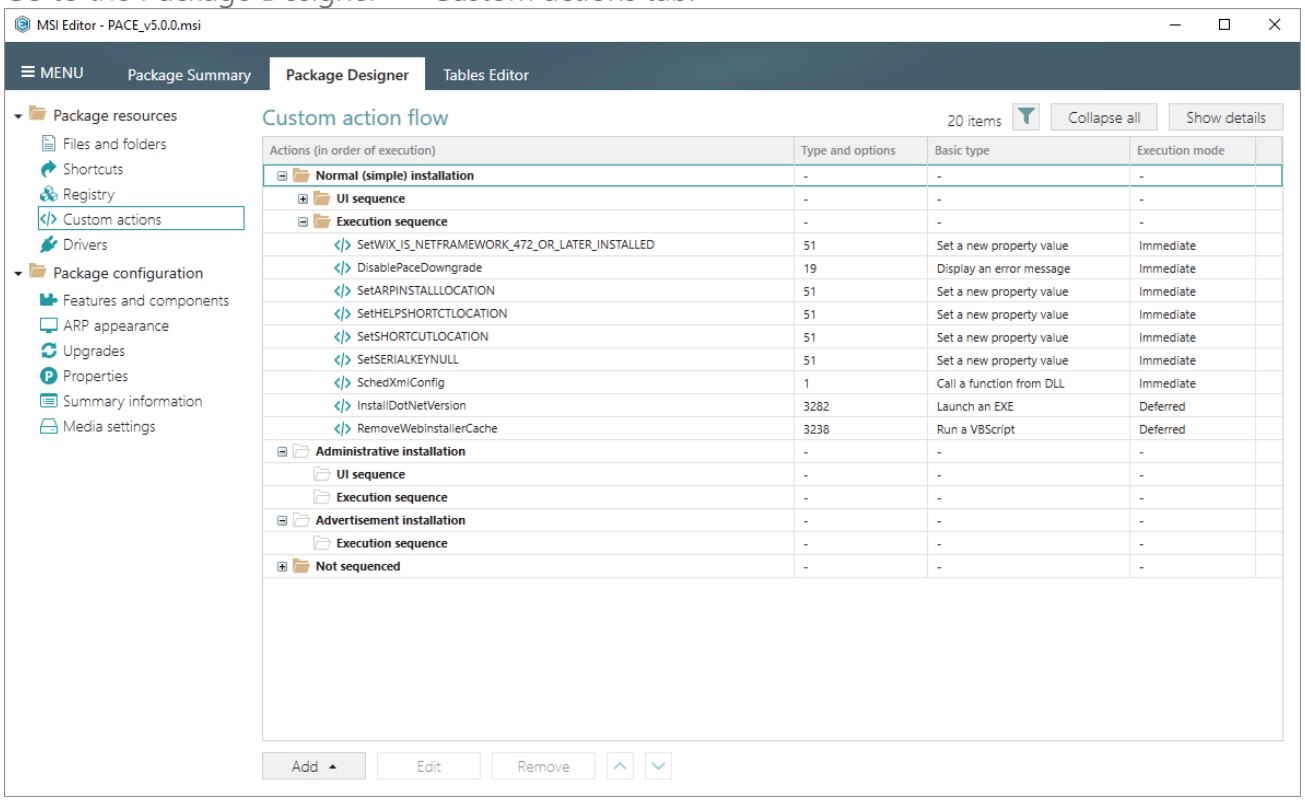
- Choose your PowerShell script and click Open.



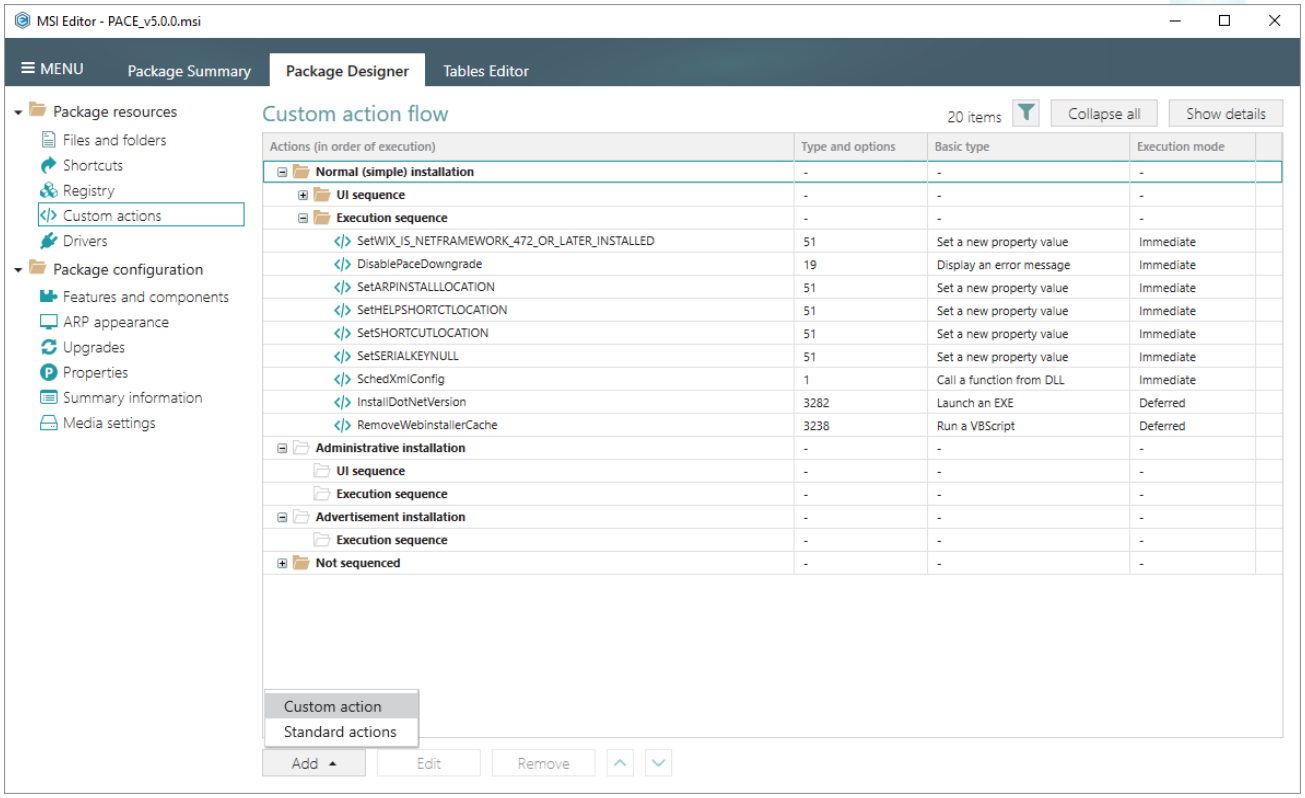
- Tick the **Complete operation with default settings** option and click **Finish**. Alternatively, you can go throught all the steps and choose the desired settings.



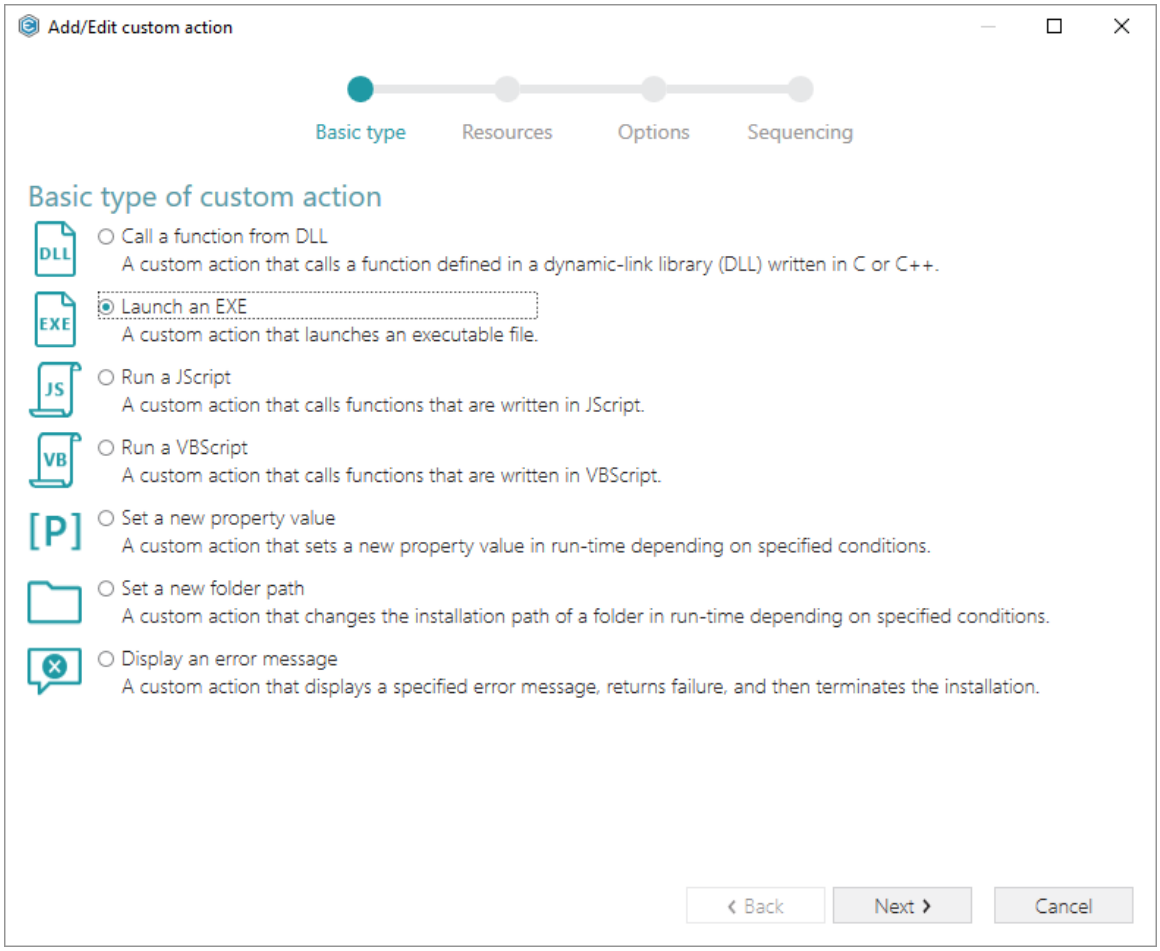
- Go to the **Package Designer** -> **Custom actions** tab.



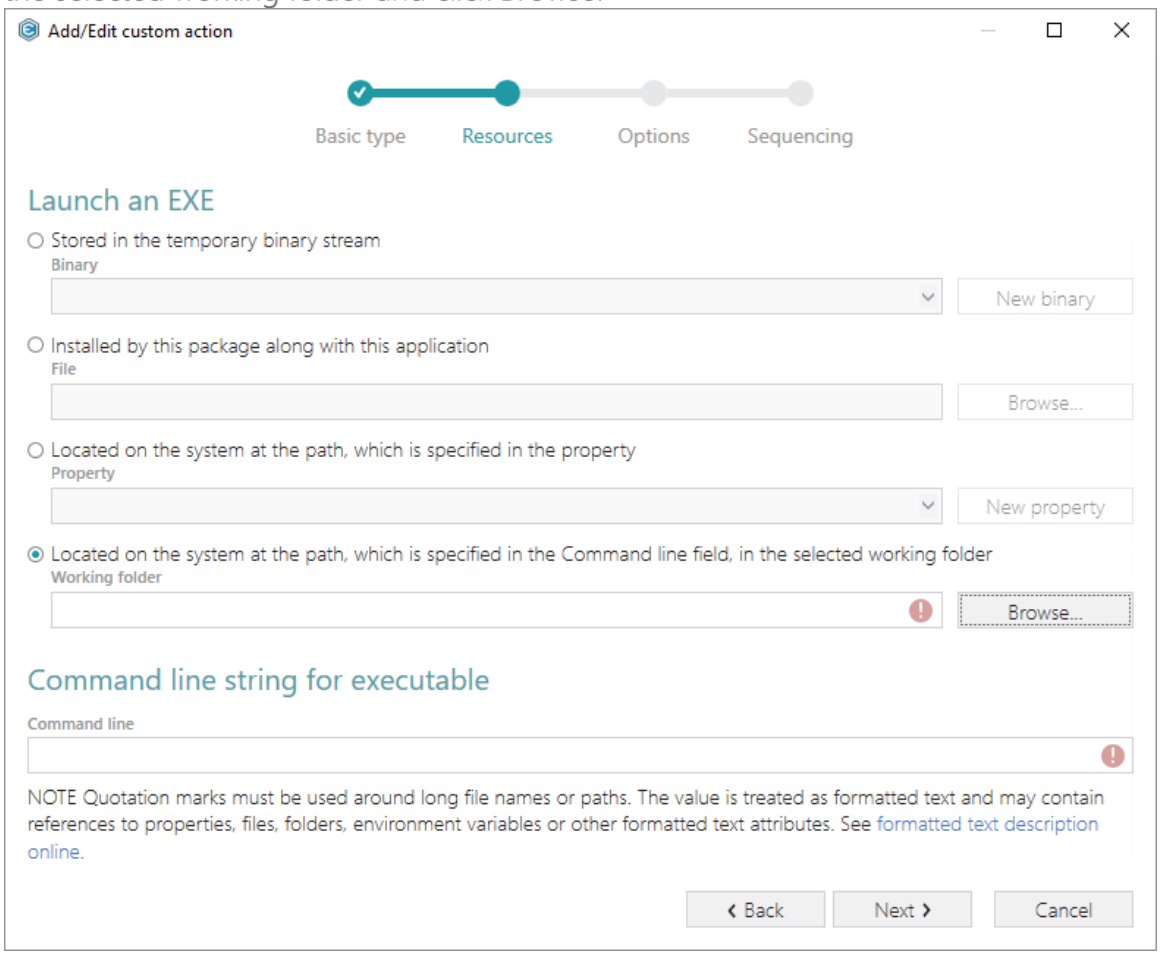
- Select Add -> Custom action from the toolbox.



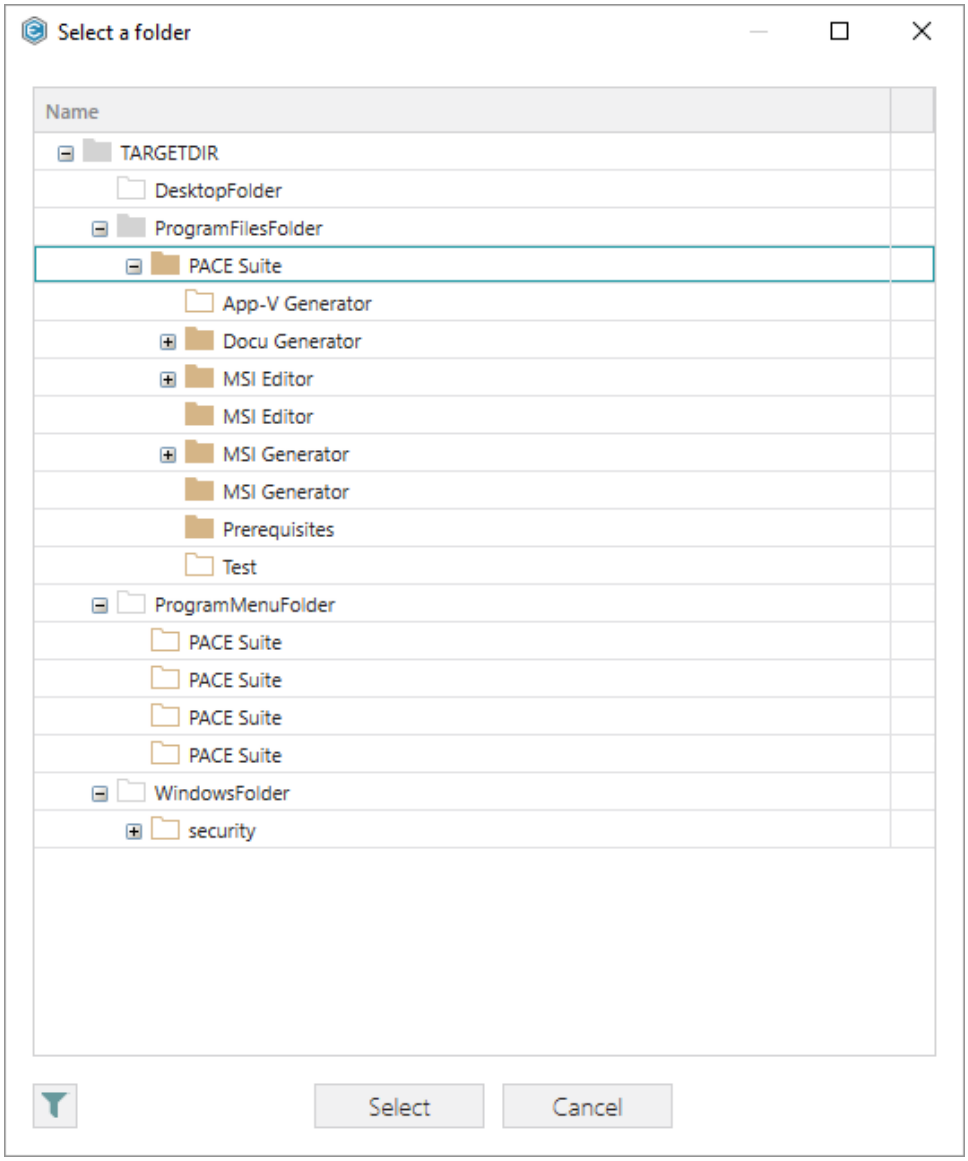
- Select Launch an EXE and click Next.



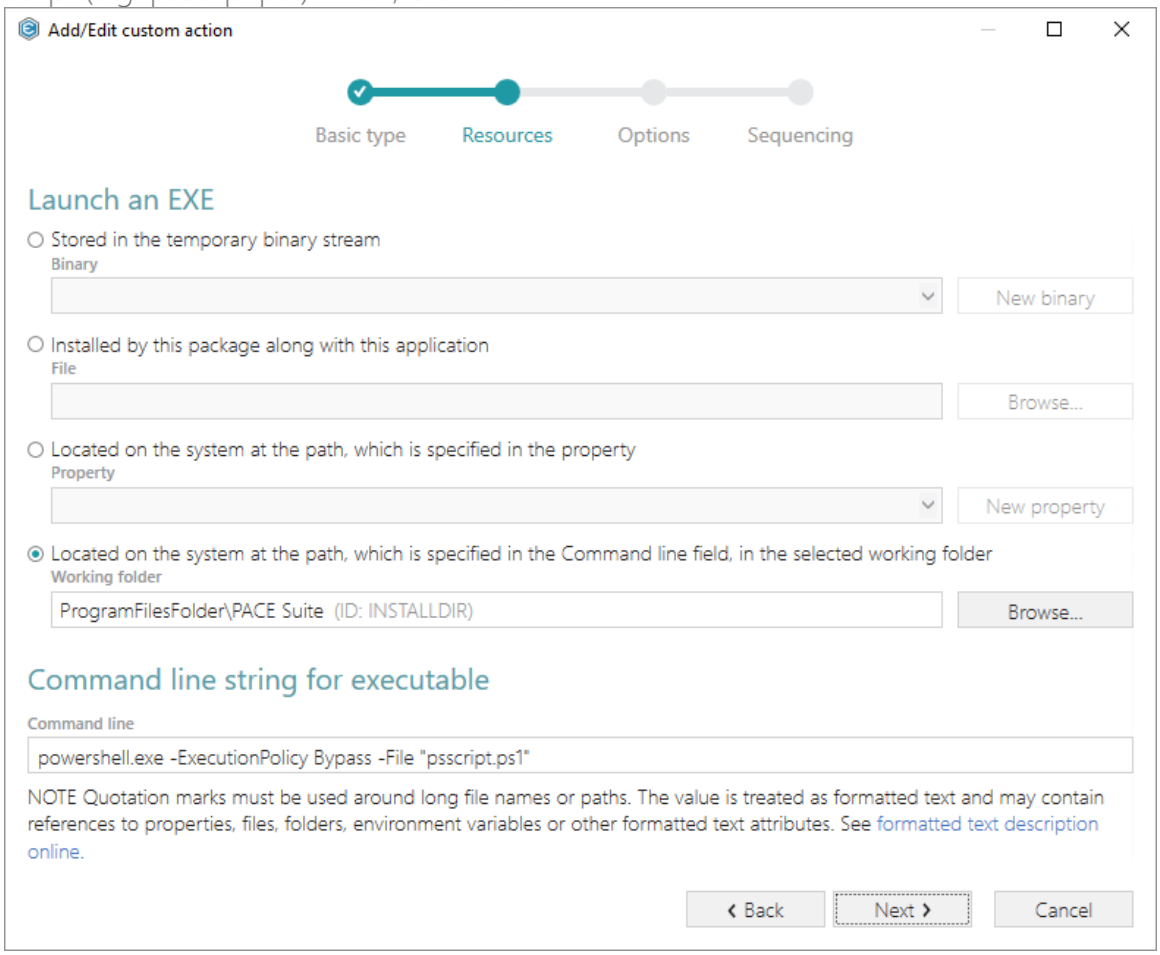
- Select the Located on the system at the path, which is specified in the Command line field, in the selected working folder and click Browse.



- Select a script-containing folder, to which you have imported the PowerShell script and then, click Select.

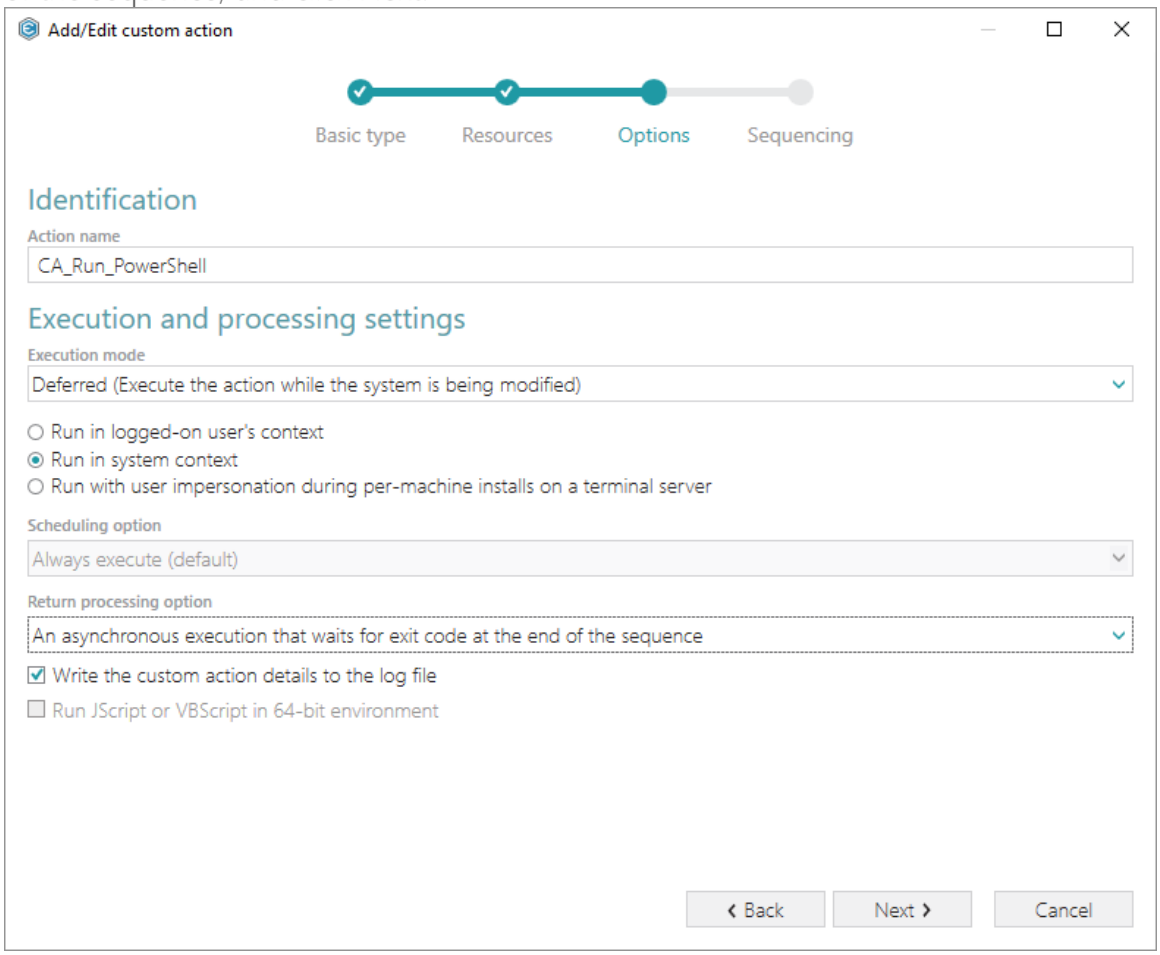


- Enter the *powershell.exe -ExecutionPolicy Bypass -File "<script file name>.ps1"* to the **Command line** field, where *<script file name>* is replaced with a name of your PowerShell script (e.g. psscript.ps1). Then, click **Next**.

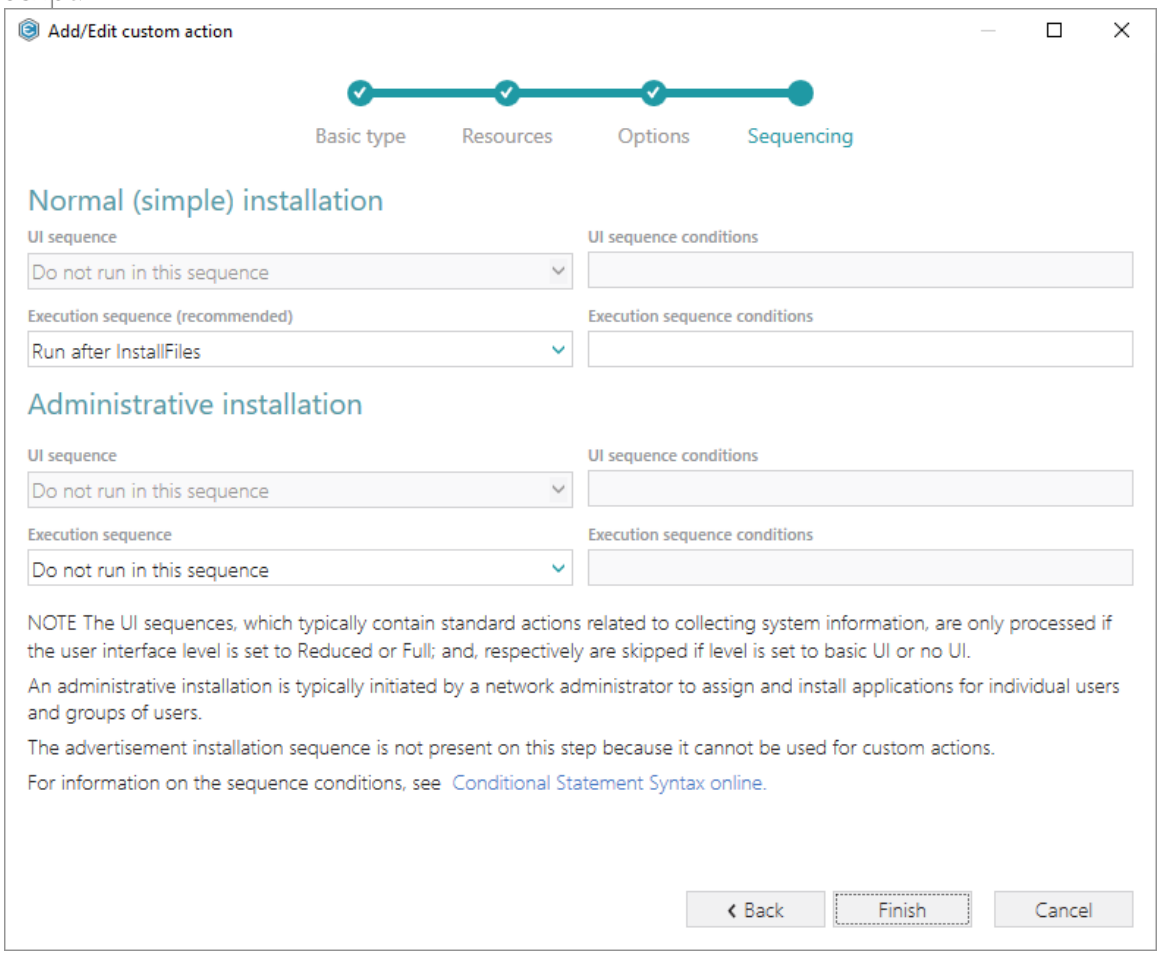


NOTE **-ExecutionPolicy Bypass** parameter was added to make sure the custom action will always execute regardless of the current execution policy on the system.

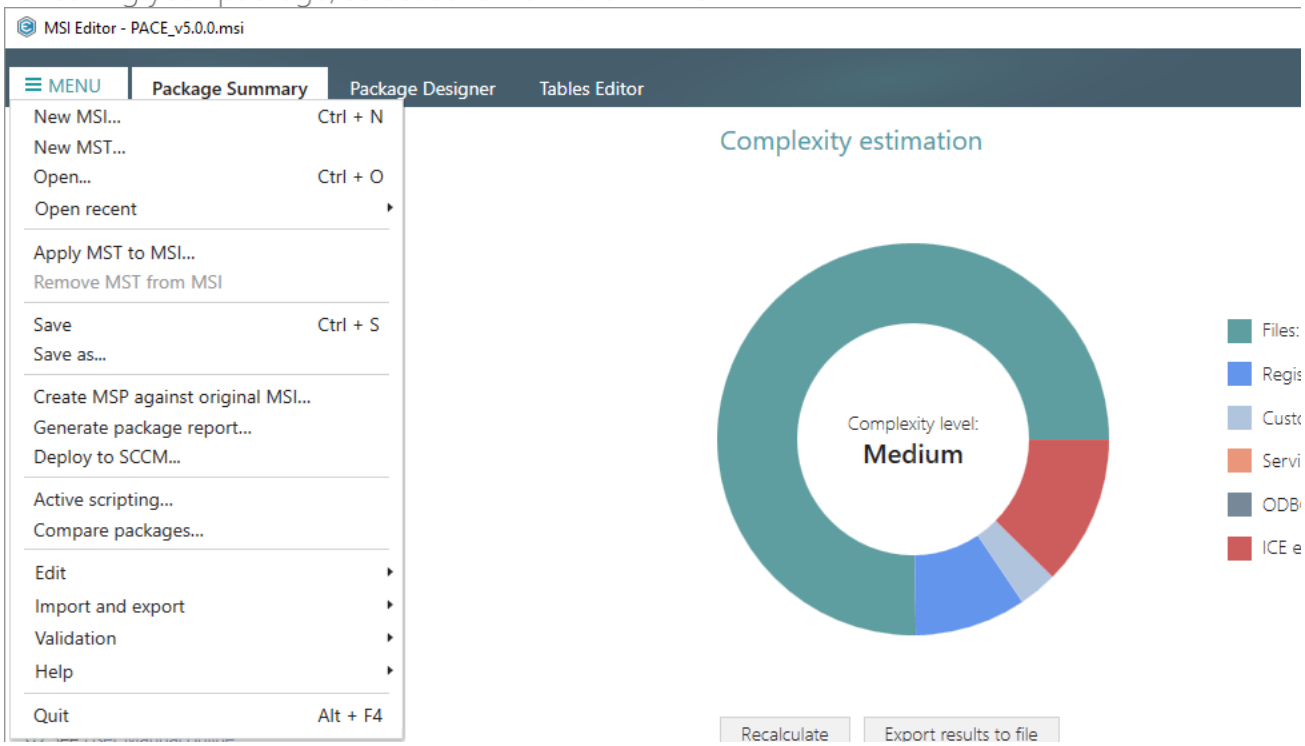
- Select the **Deferred** execution mode, and the **Run in system context** option. From the **Return processing options** list, choose **An asynchronous execution that waits for exit code at the end of the sequence**, and click **Next**.



- From the **Execution sequence (recommended)** list, select **Run after InstallFiles** (or any other position, located below). If needed, specify the execution conditions - enter them manually or choose from the dropped list. Click **Finish** to add your custom action for running PowerShell script.



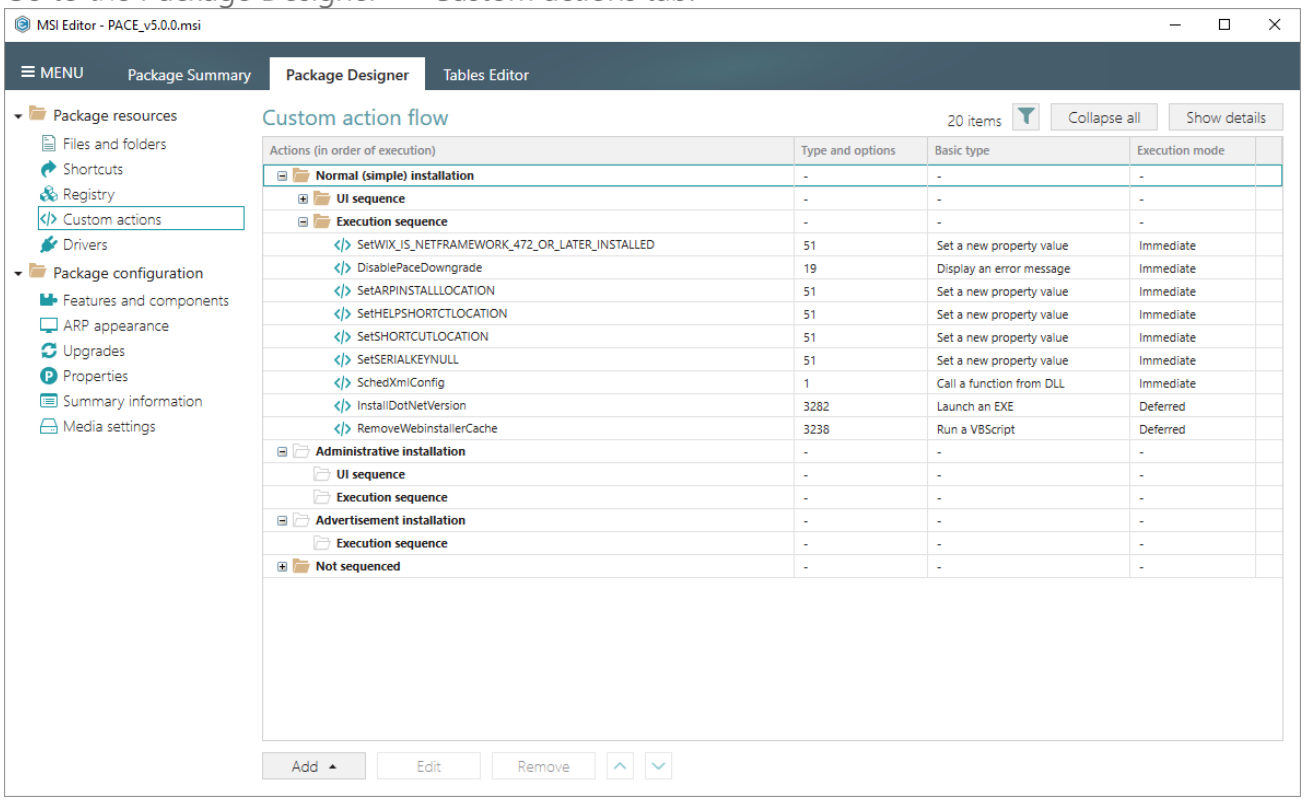
- For saving your package, select **Save** from the **MENU**.



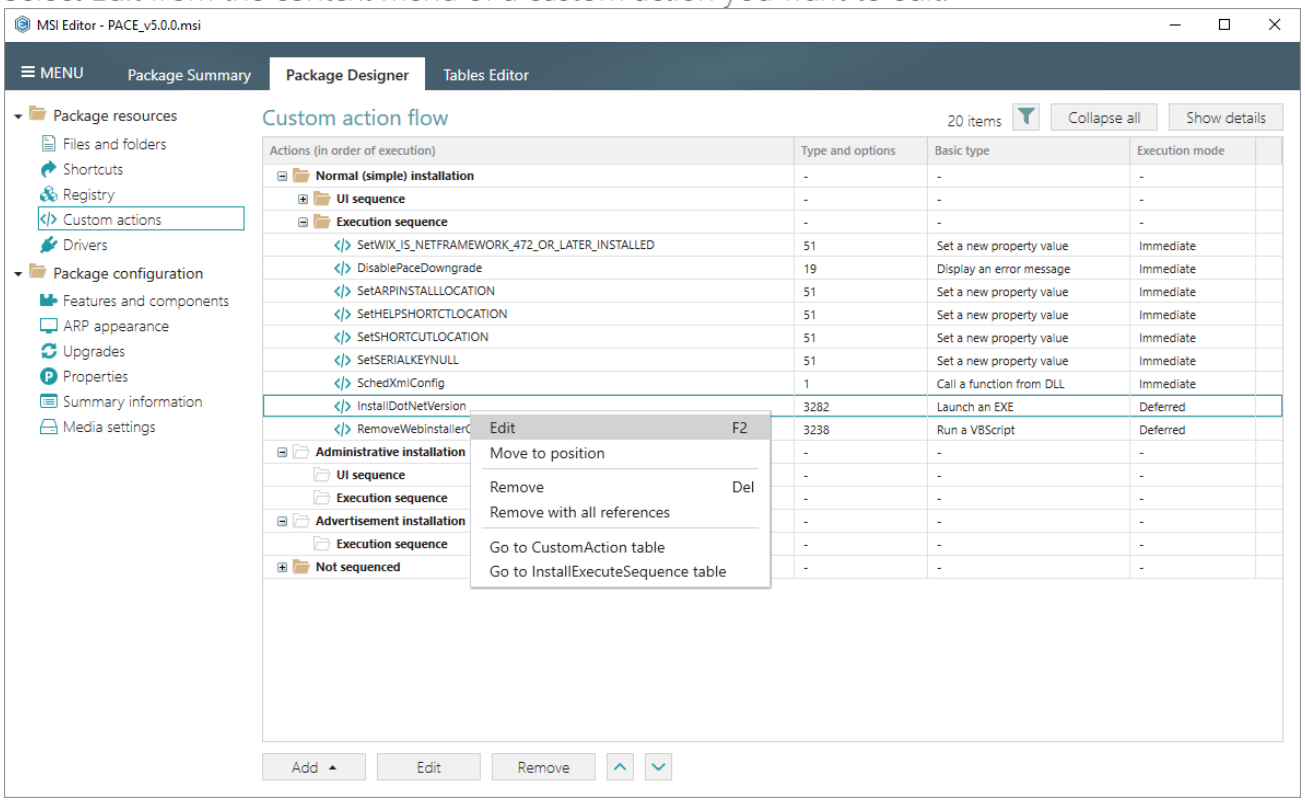
3.7.7.3 Edit Custom Action

Edit basic type, resources, sequence and other settings of the existing custom action.

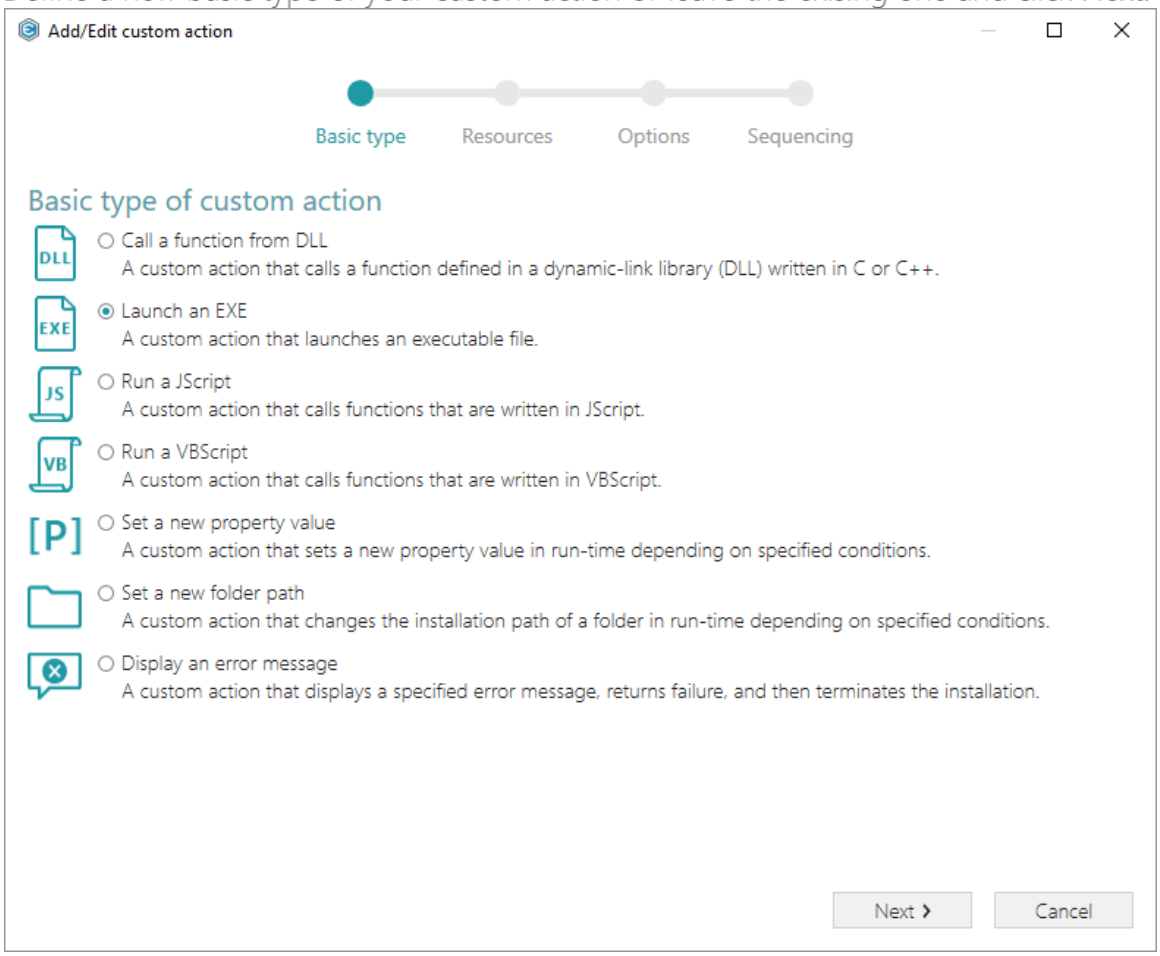
- Go to the Package Designer -> Custom actions tab.



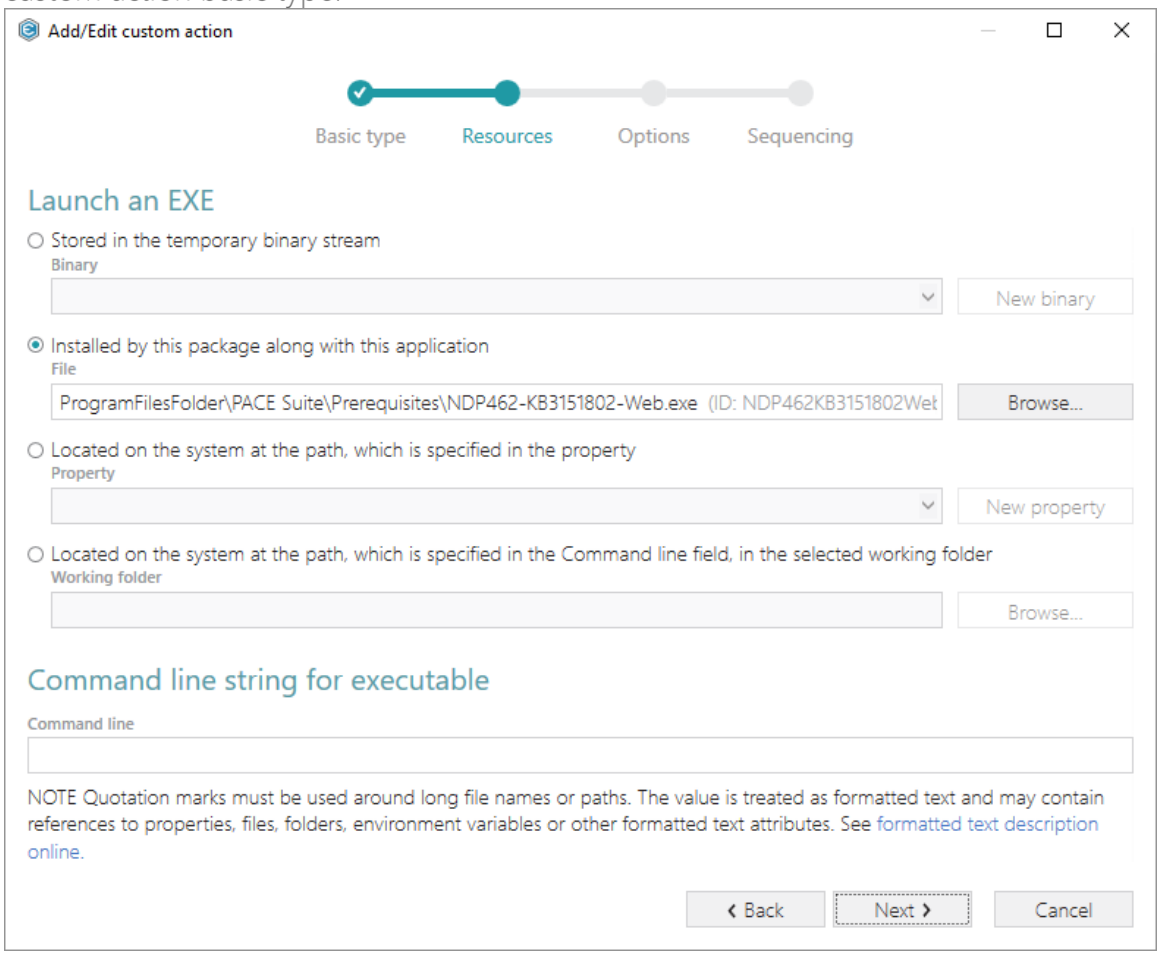
- Select Edit from the context menu of a custom action you want to edit.



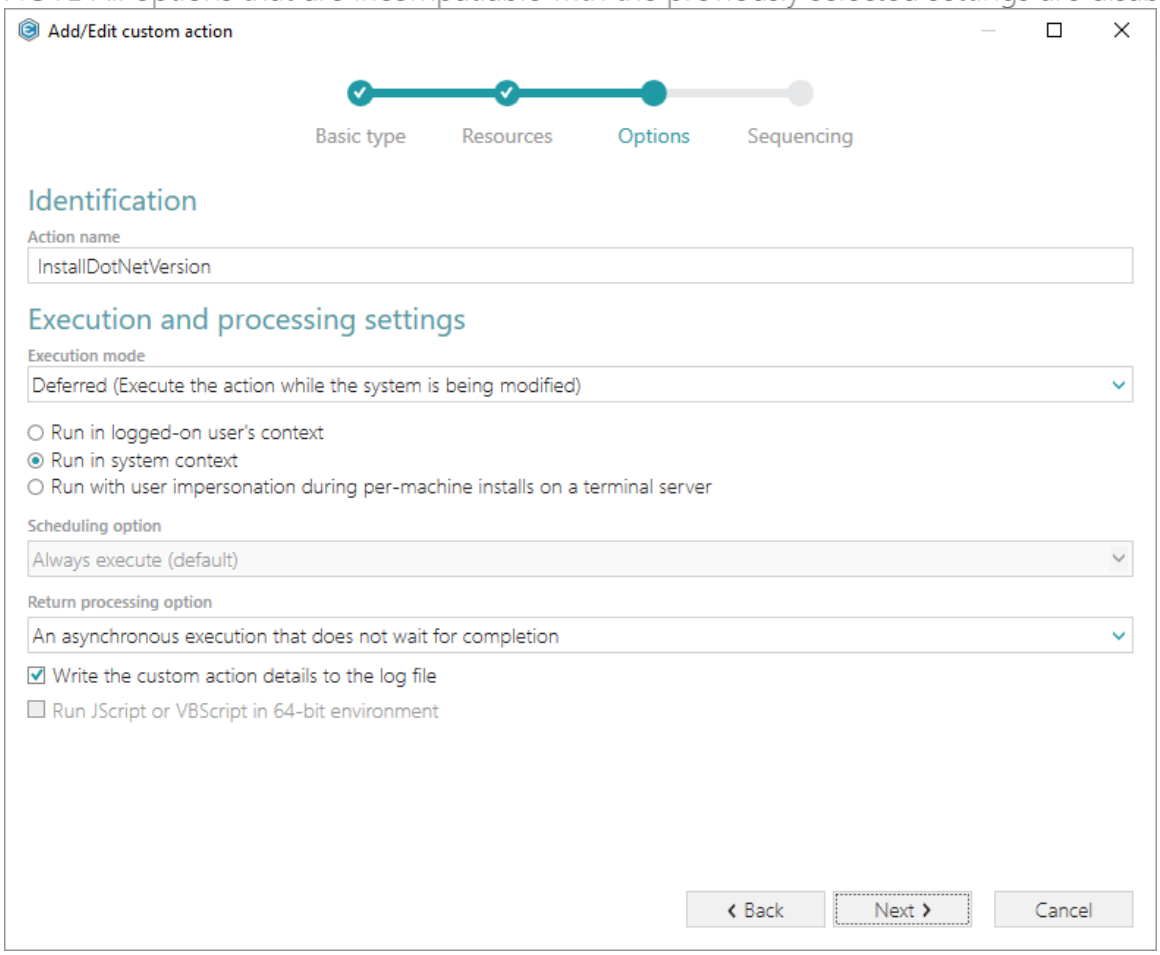
- Define a new basic type of your custom action or leave the exising one and click Next.



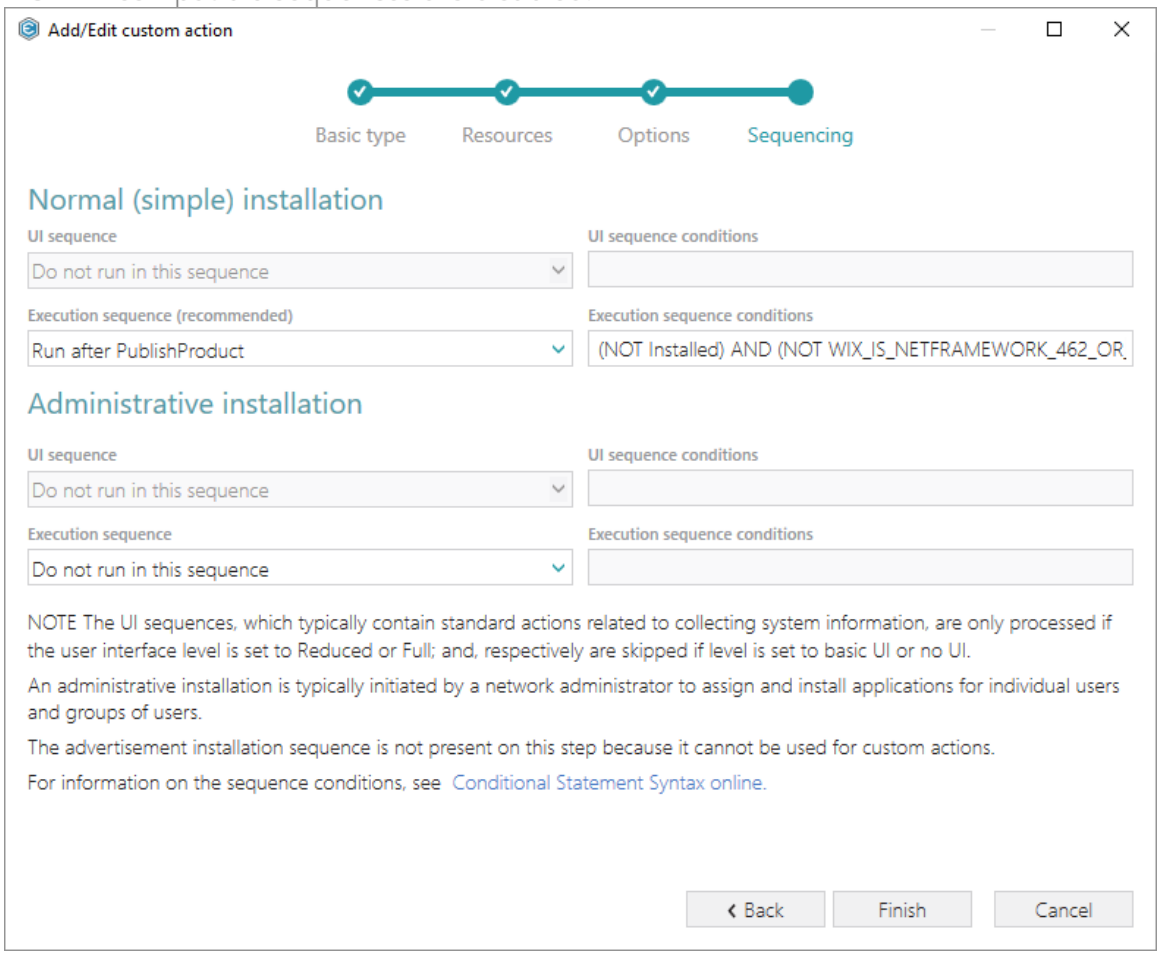
- On the second step you can specify a new file, launched by the custom action. Click Next to continue.
  NOTE The wizard displays only compatible fields and options for the previously selected custom action basic type.



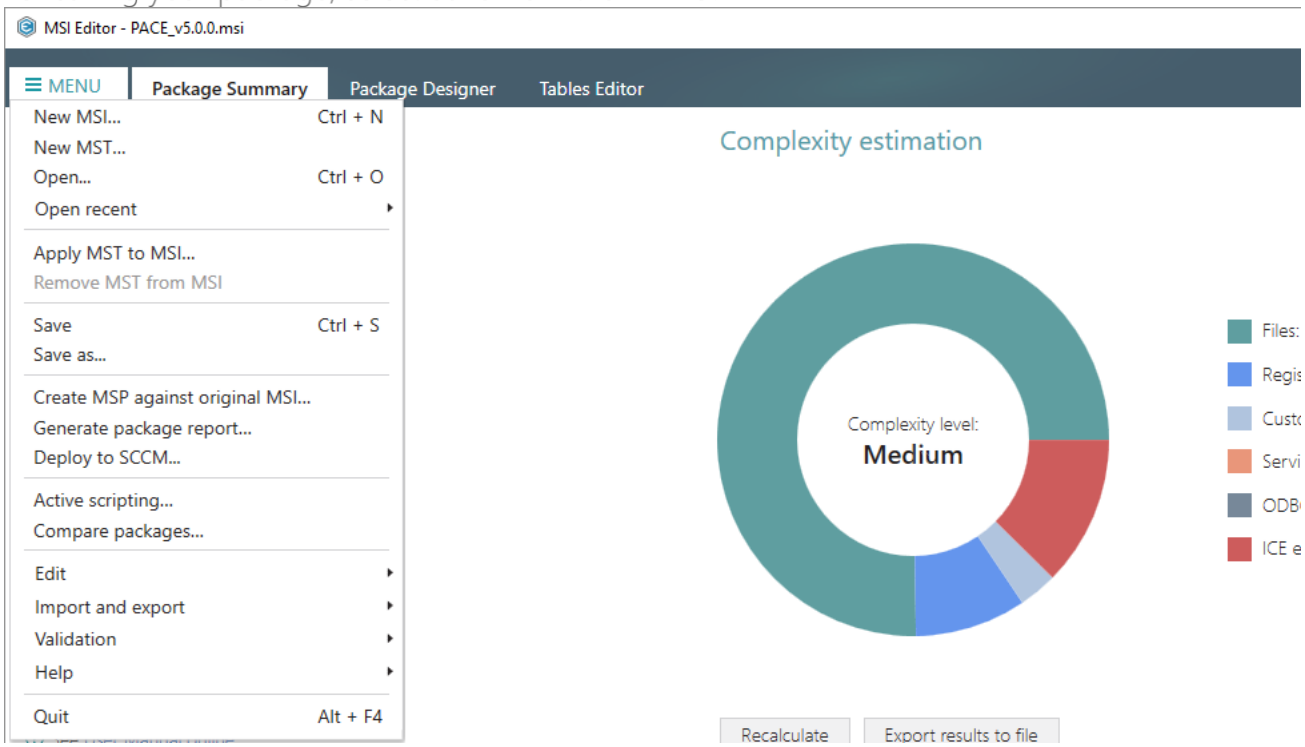
- At the **Options** step you can set an **Action name**, select **Execution mode** (Immediate, Deferred, Rollback and Commit), execution context, scheduling, return processing and other options. Then, click **Next**.
  NOTE All options that are incompatiable with the previously selected settings are disabled.



- At the last step, you can change a position, where you would like to run your custom action. The sequence drop-down list allows you to select only compatible positions, taking into account your previously selected options. Also, you free to update the execution conditions either by entering them manually or choosing from the drop-down list. Click **Finish** to update your custom action.
  NOTE Incompatible sequences are disabled.



- For saving your package, select **Save** from the **MENU**.



3.7.8 Drivers

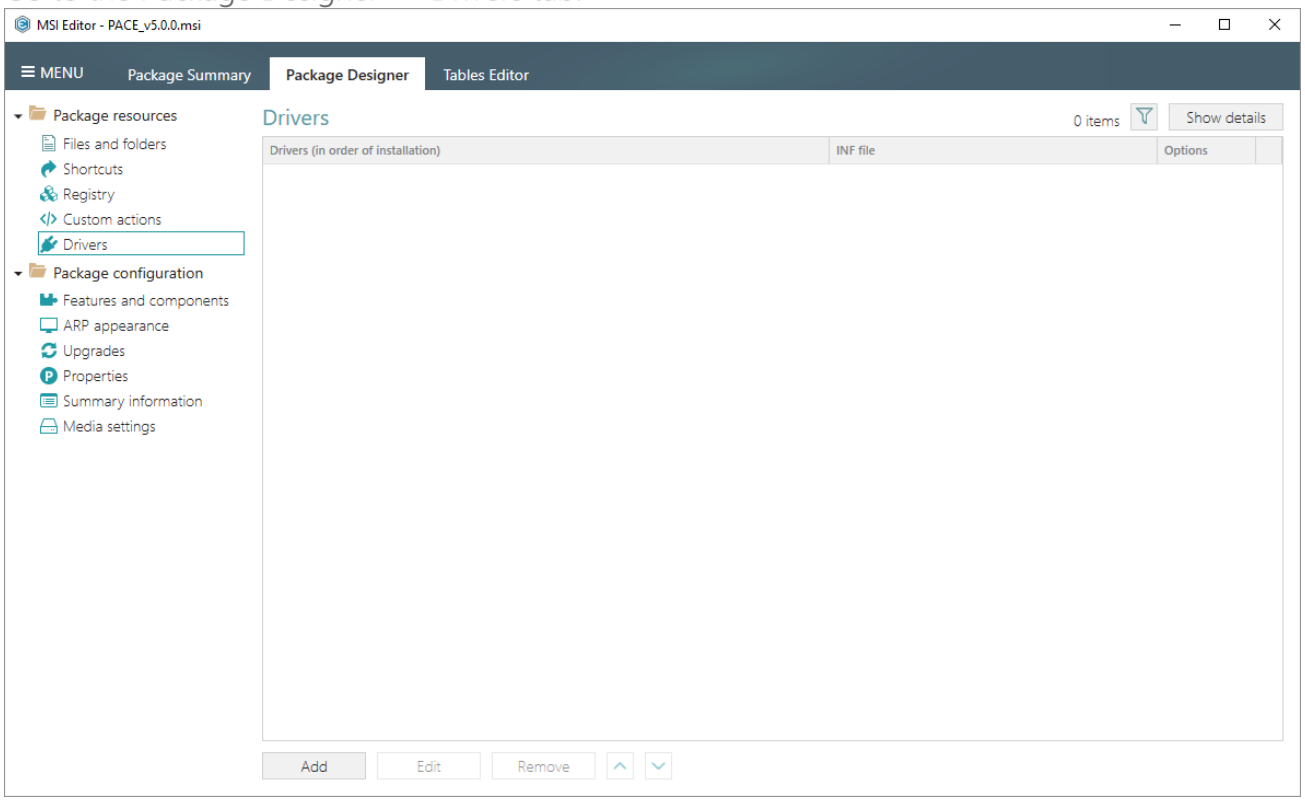
Choose a scenario that better suits your needs:

- Add Driver, described in section 3.7.8.1
  Add a new device driver with the help of Driver Install Frameworks (DIFx) tools, a standard solution recommended by Microsoft.
- Edit Driver, described in section 3.7.8.2
  Edit installation options, conditions and target architecture of the existing device driver.

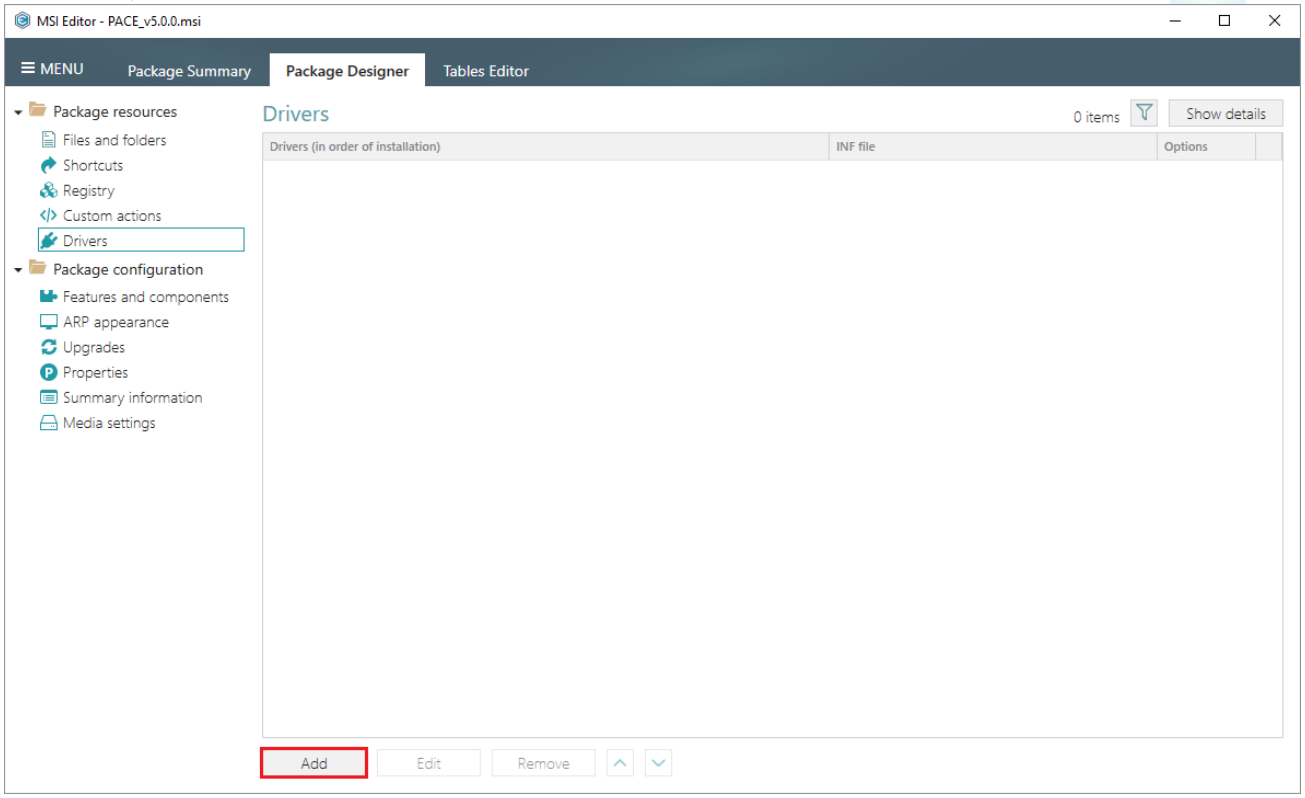
3.7.8.1 Add Driver

Add a new device driver with the help of Driver Install Frameworks (DIFx) tools, a standard solution recommended by Microsoft.

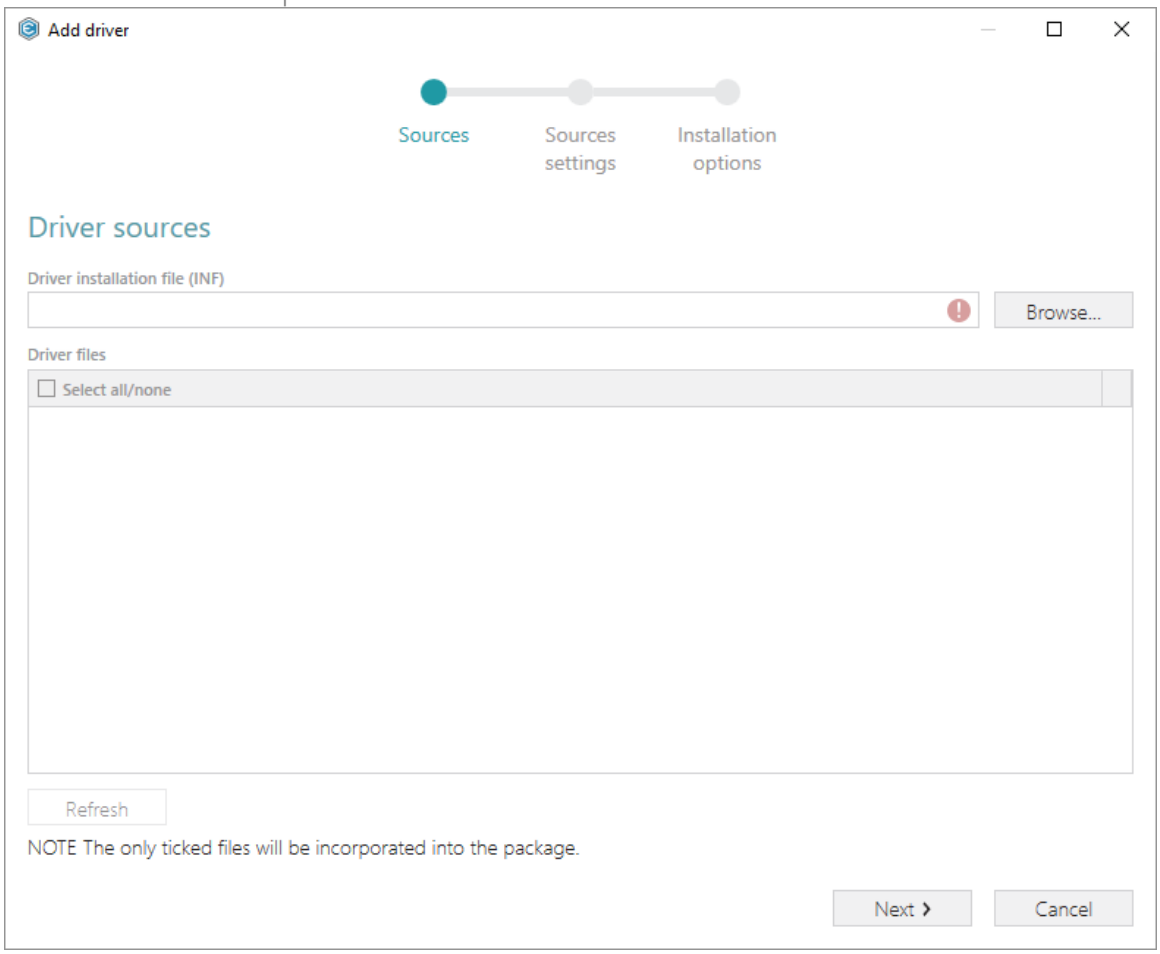
- Go to the Package Designer -> Drivers tab.



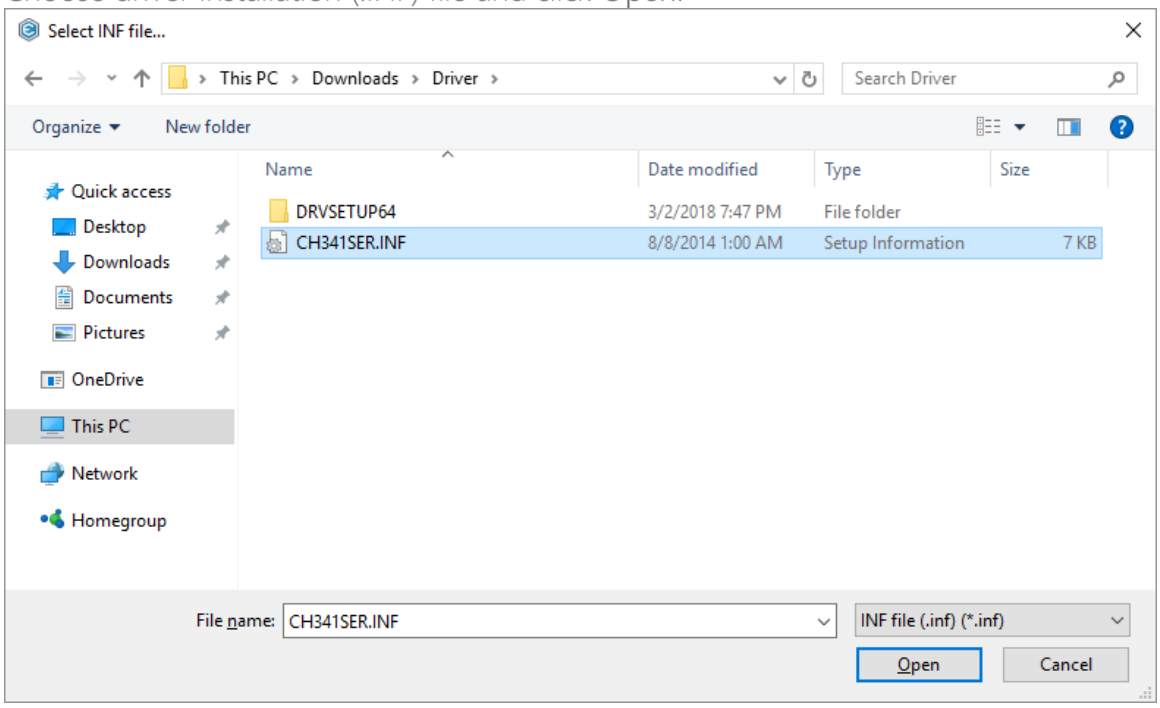
- Click **Add**, located in the toolbox.



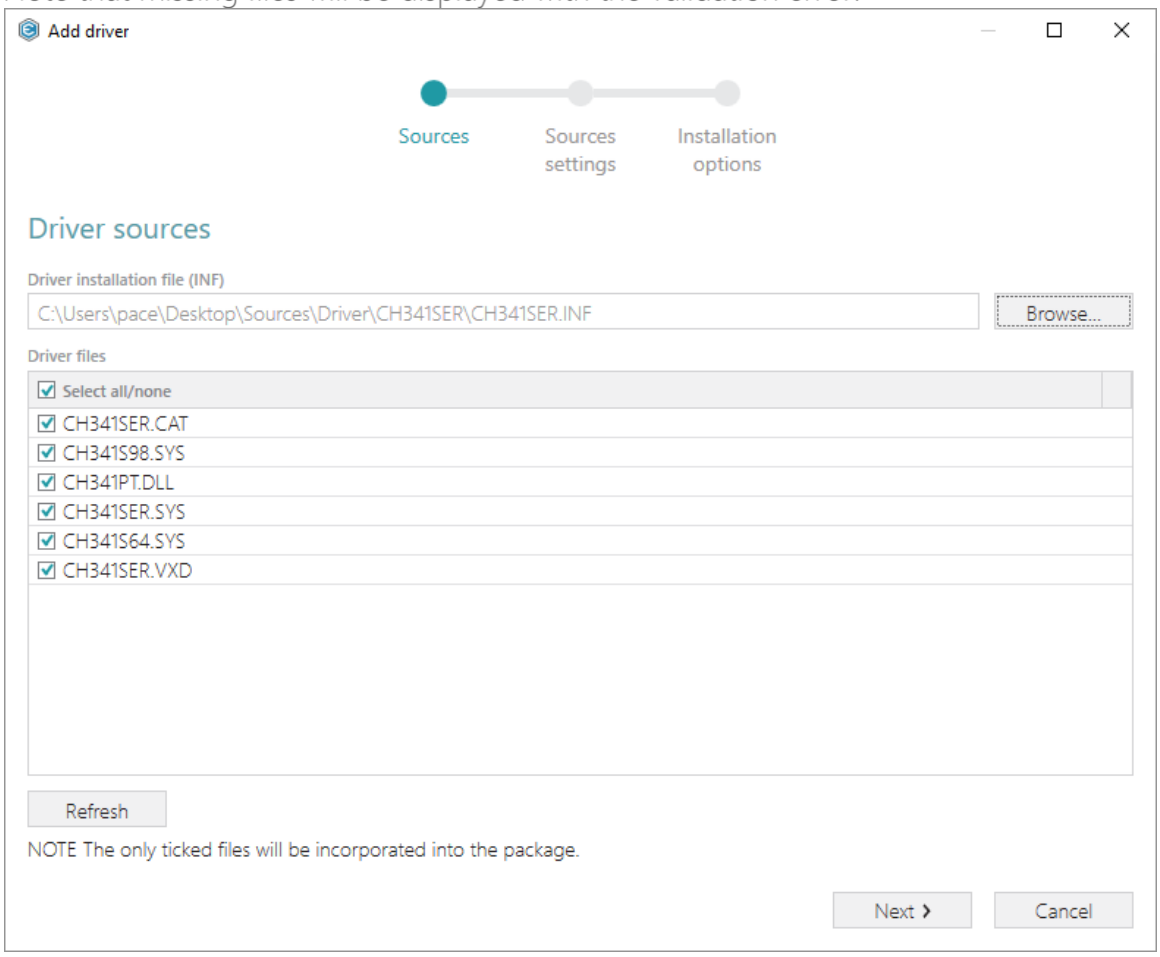
- Click **Browse** in the opened wizard.



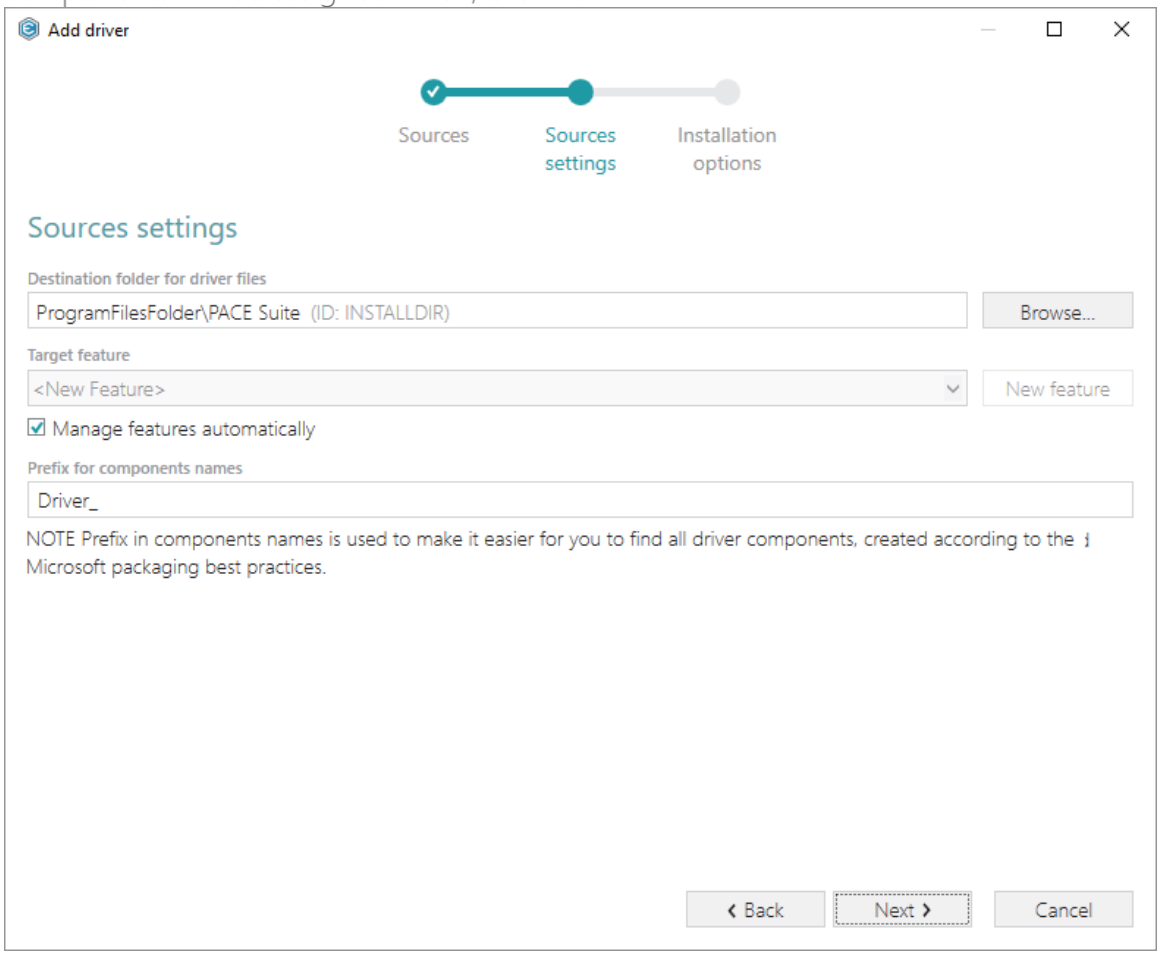
- Choose driver installation (.INF) file and click **Open**.



- Ensure that all files, required by driver, exist on the file system, and click **Next**.
Note that missing files will be displayed with the validation error.

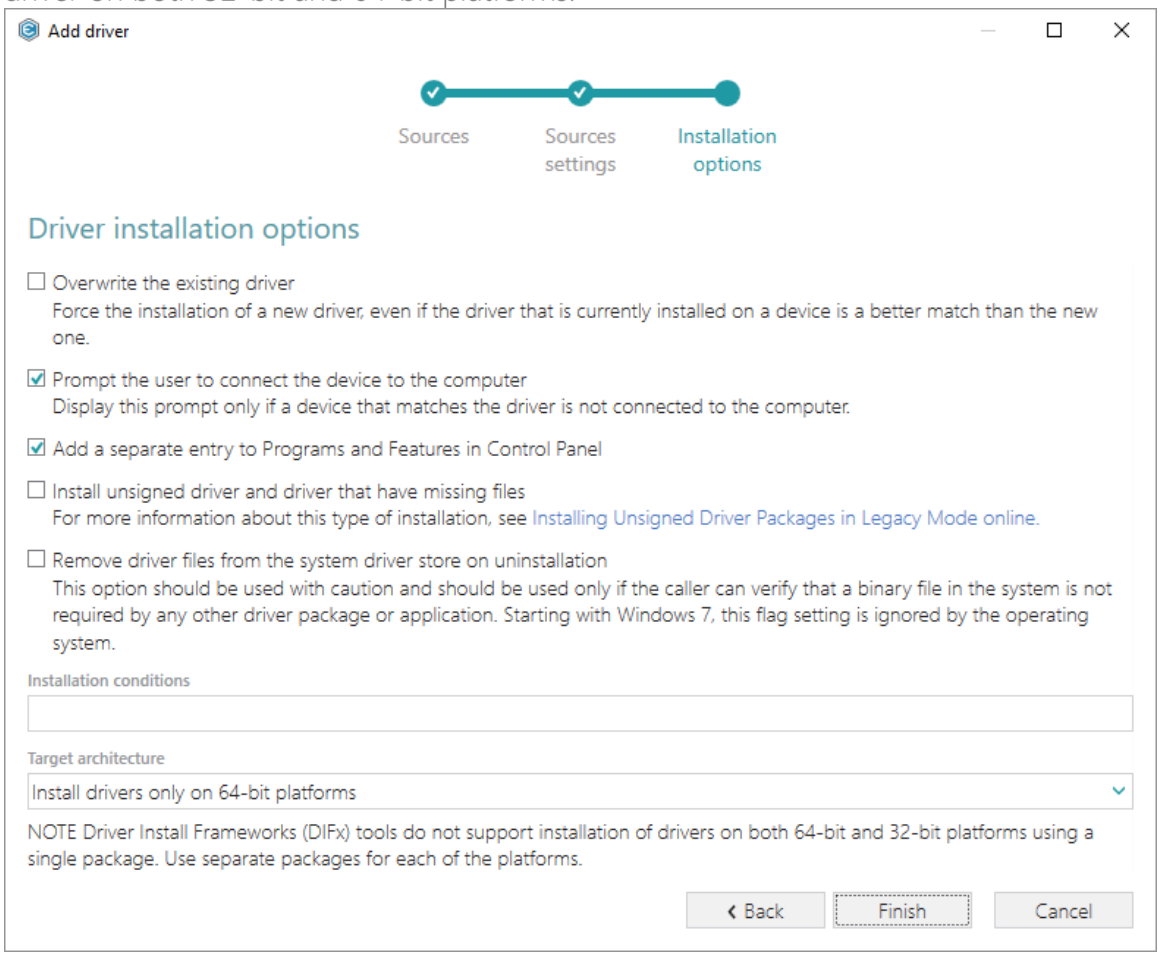


- Select a **Destination folder for the driver files** (the wizard additionally creates a subfolder in the selected folder). Also, you can manually specify a **Target feature**, to which the created driver components will be assigned. Then, click **Next**.

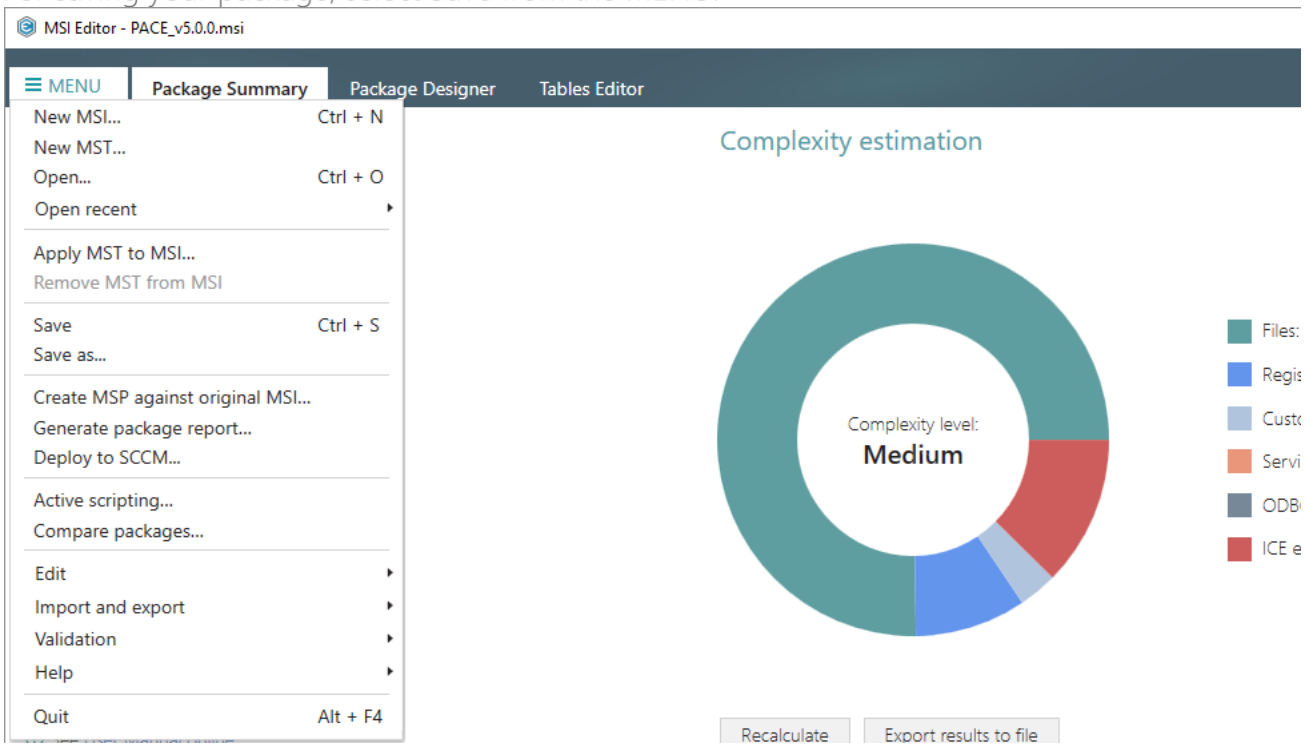


- At the last step, you can select **driver installation options**, used by Driver Install Frameworks (DIFx) tool, specify **installation conditions**, which will be used for the main driver component (see Conditional Statement Syntax), and select a **target architecture**, which defines whether a 32-bit or 64-bit DIFx tool will be used. Click **Finish** to add the driver.
  NOTE One package can not contain both 32-bit and 64-bit DIFx tool, and as a result, one package can not install driver on both platforms. Use separate packages if you need to install

driver on both 32-bit and 64-bit platforms.



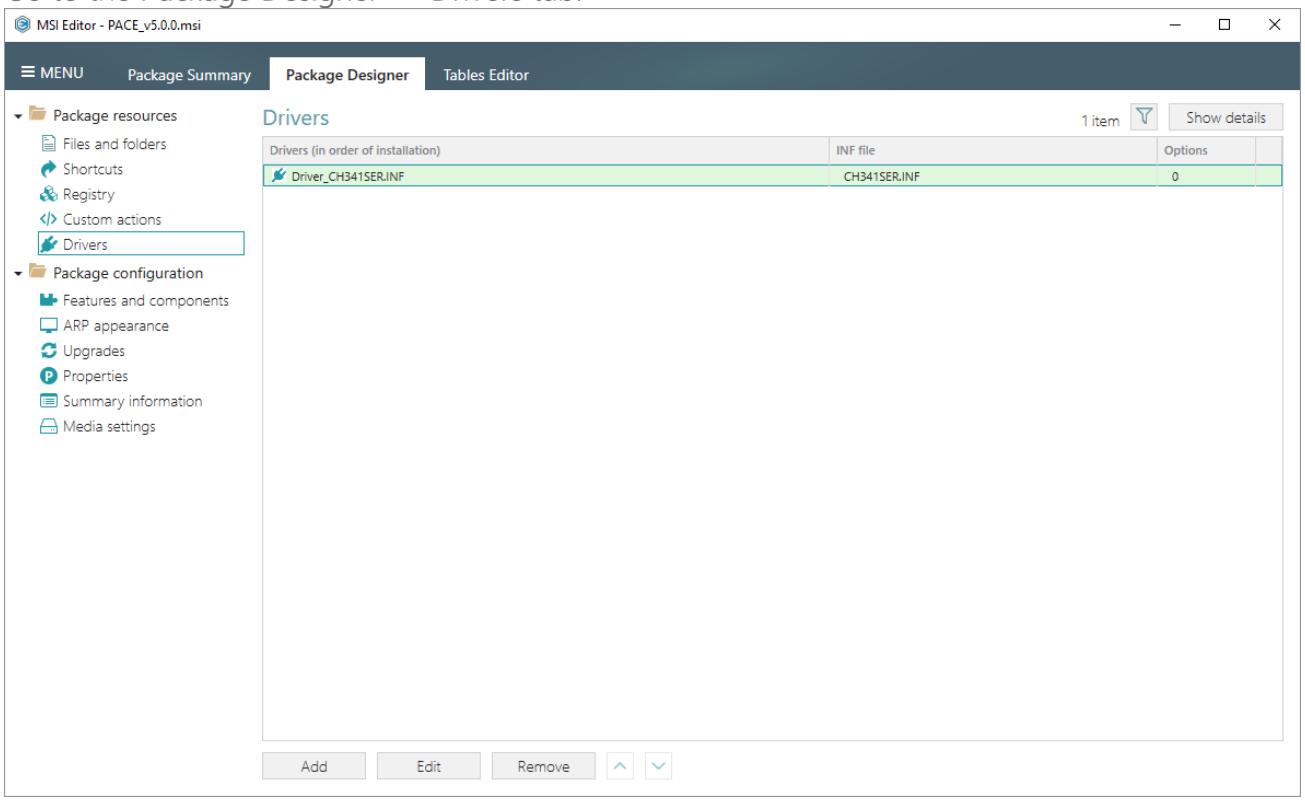
- For saving your package, select Save from the MENU.



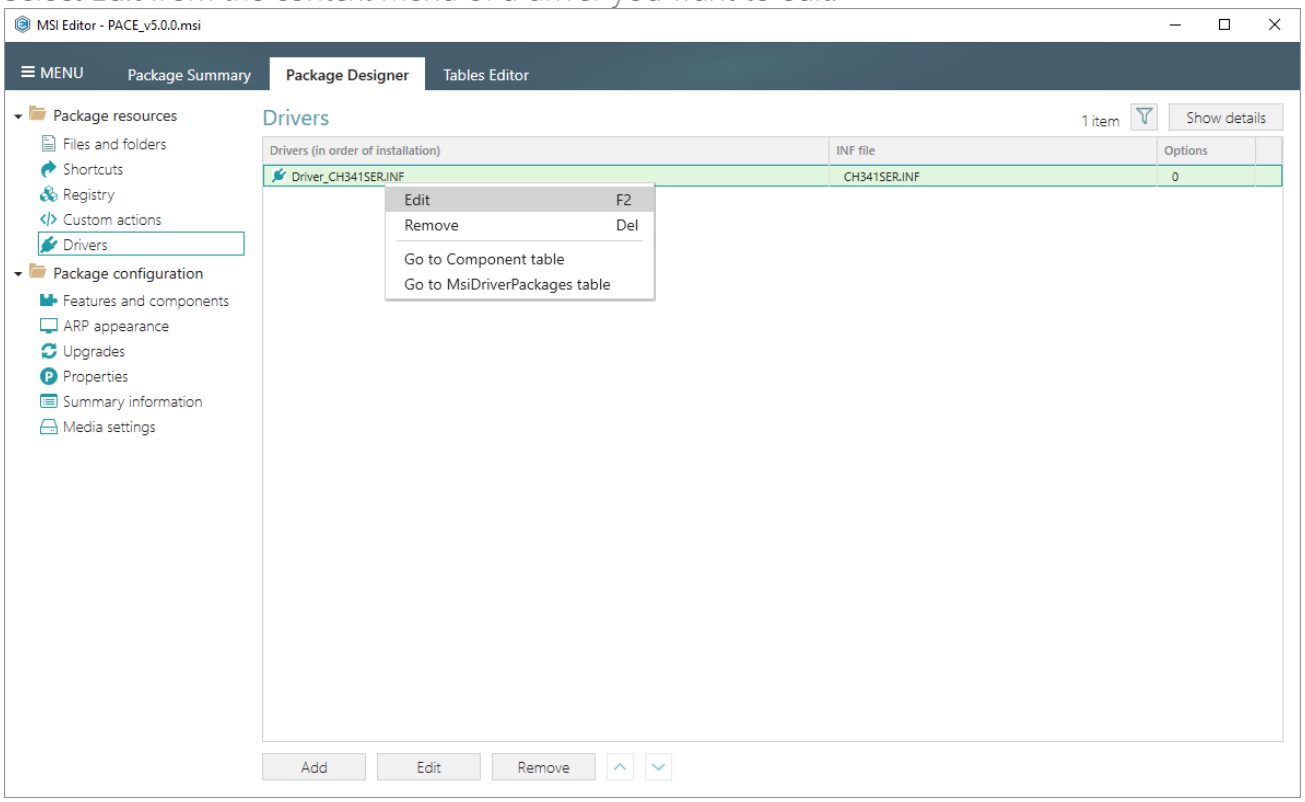
3.7.8.2 Edit Driver

Edit installation options, conditions and target architecture of the existing device driver.

- Go to the **Package Designer** -> **Drivers** tab.

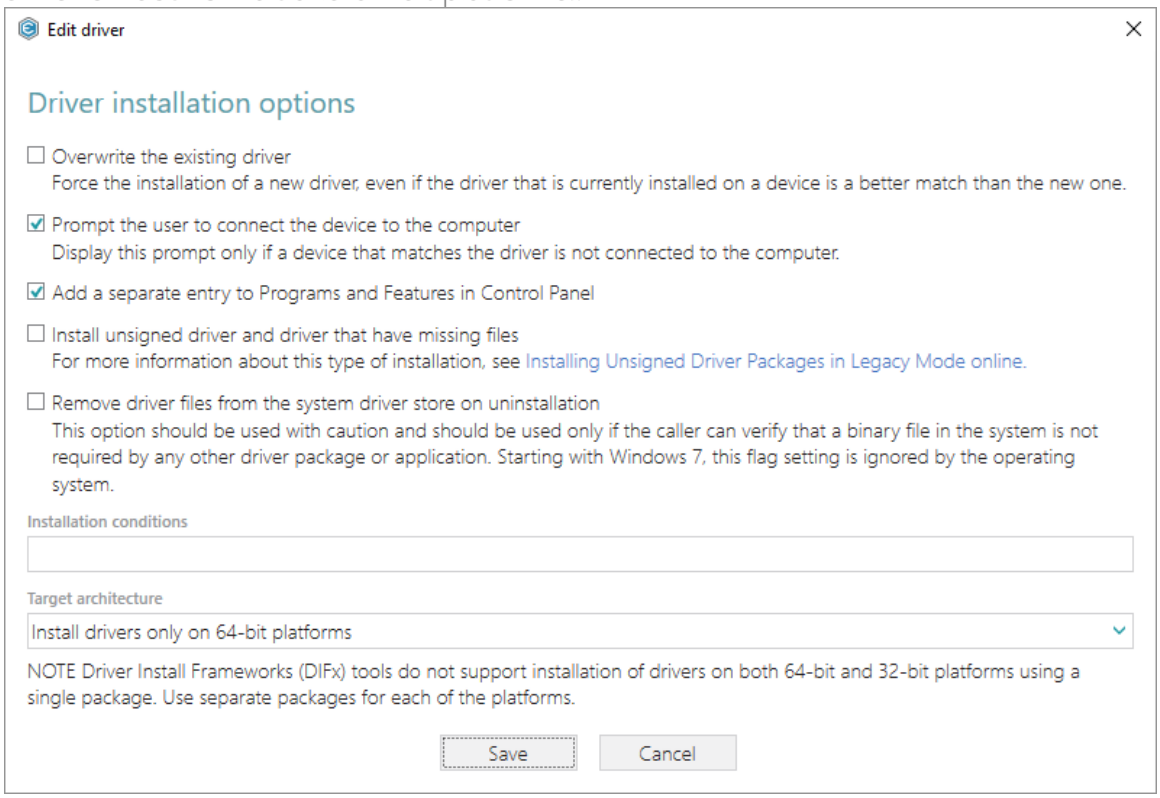


- Select **Edit** from the context menu of a driver you want to edit.

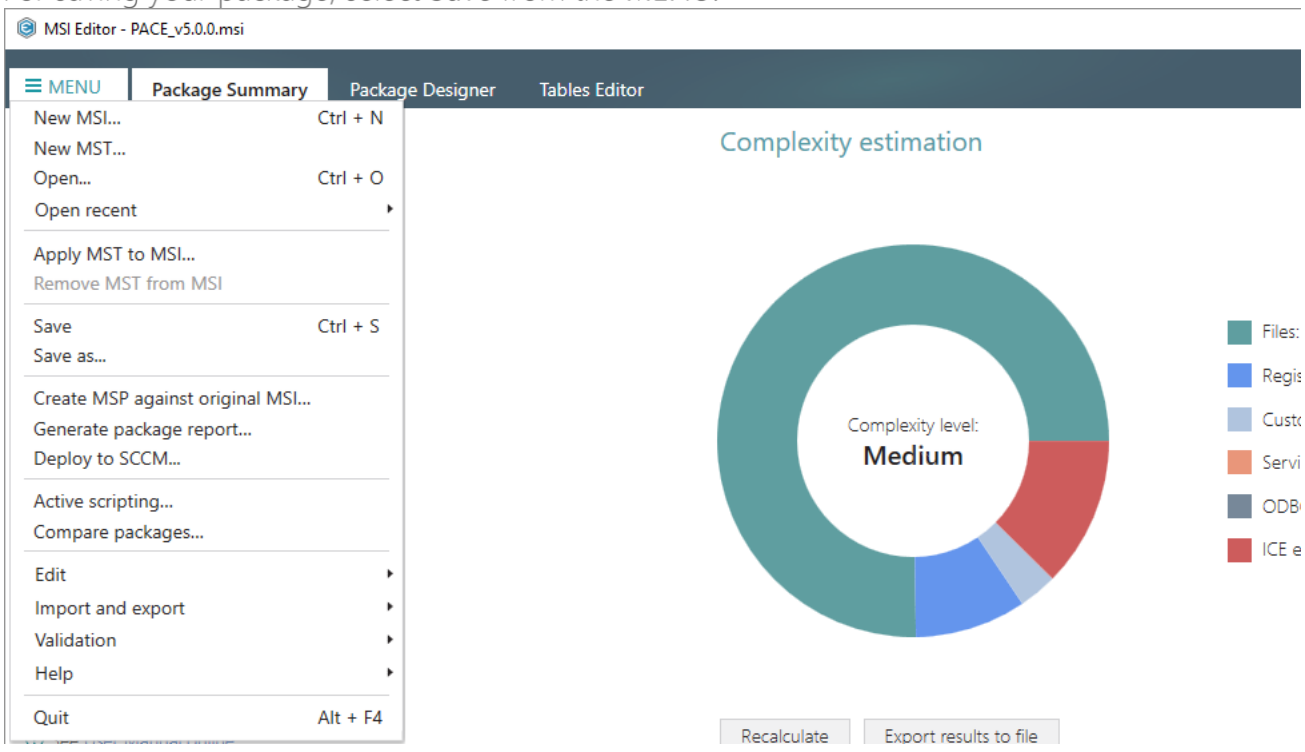


- In the opened window you can change **driver installation options**, used by Driver Install Frameworks (DIFx) tool, specify **installation conditions**, which will be used for the main driver component (see Conditional Statement Syntax), and change a **target architecture**, which defines whether a 32-bit or 64-bit DIFx tool will be used. Click **Save** to update the driver settings.

NOTE One package can not contain both 32-bit and 64-bit DIFx tool, and as a result, one package can not install driver on both platforms. Use separate packages if you need to install driver on both 32-bit and 64-bit platforms..



- For saving your package, select Save from the MENU.



3.7.9 Features and Components

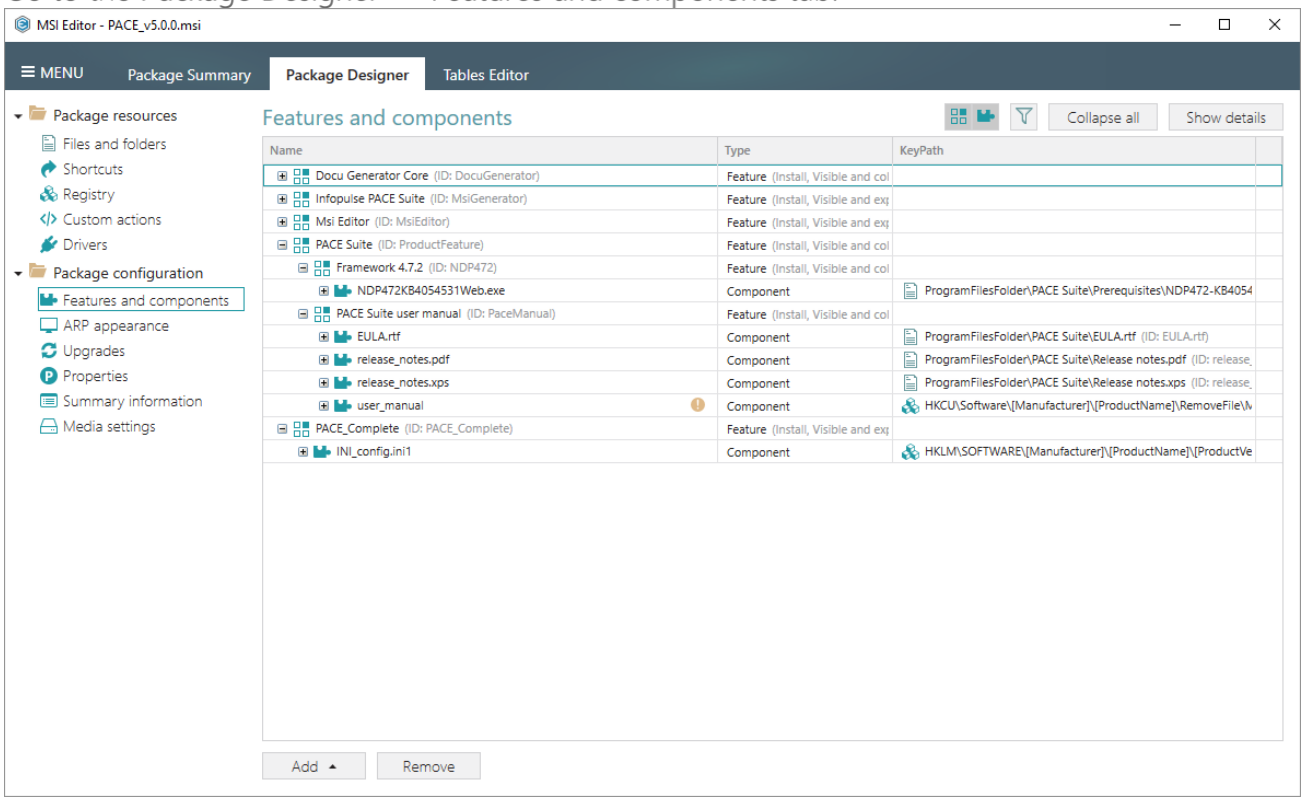
Choose a scenario that better suits your needs:

- Add Feature, described in section 3.7.9.1
  Add a new root feature or subfeature to the package.
- Add Component, described in section 3.7.9.2
  Add a new component to one or many features and set the component properties.
- Move Feature, Component, described in section 3.7.9.3
  Move a feature or a component to another feature.
- Edit Feature Properties, described in section 3.7.9.4
  Edit feature properties such as ID, title, folder, installation state, type, display options and advertisement.
- Edit Component Properties, described in section 3.7.9.5
  Edit component properties such as name, GUID, conditions, key path, attributes and relationship with features.
- Detect Empty Components, described in section 3.7.9.6
  Find empty components with no linked resources in order to remove them.
- Review Component Resources, described in section 3.7.9.7
  Review what resources (files, registry, shortcuts, etc.) are linked with the component.

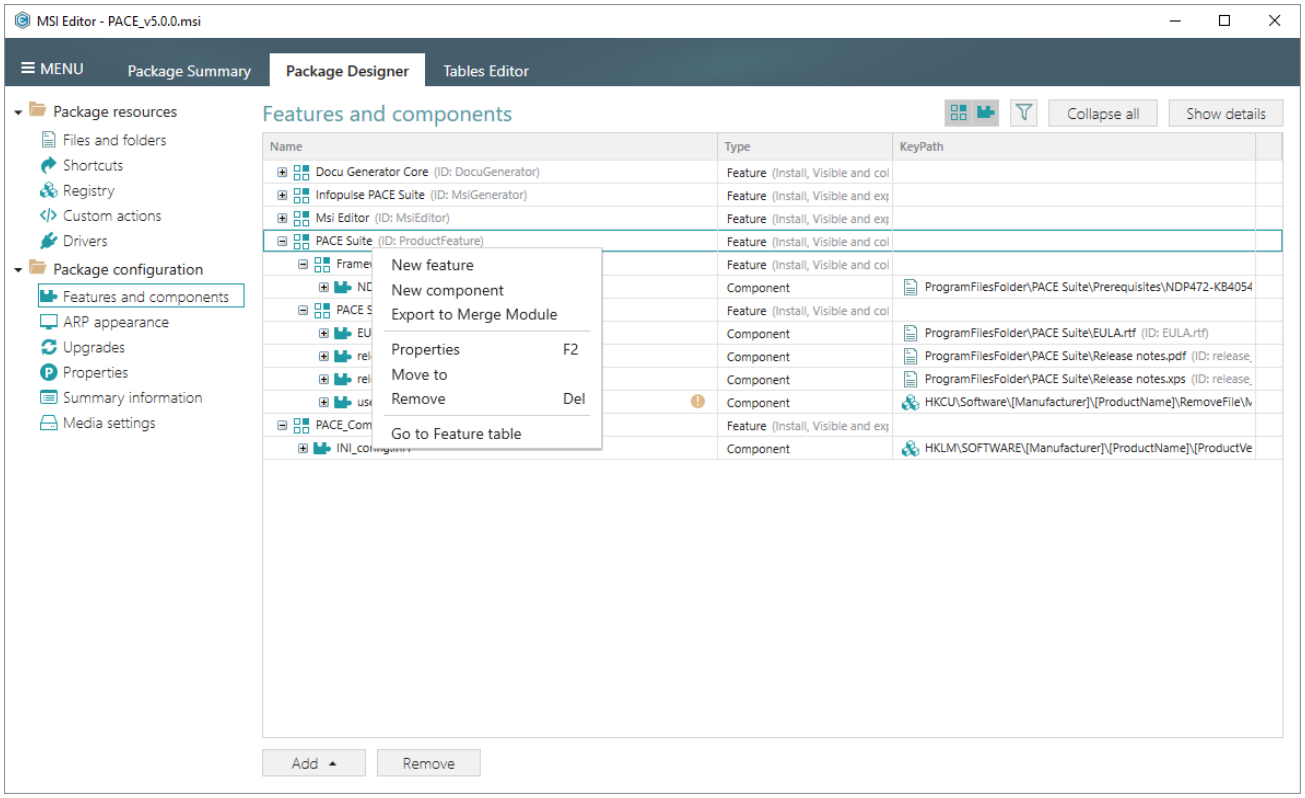
3.7.9.1 Add Feature

Add a new root feature or subfeature to the package.

- Go to the Package Designer -> Features and components tab.

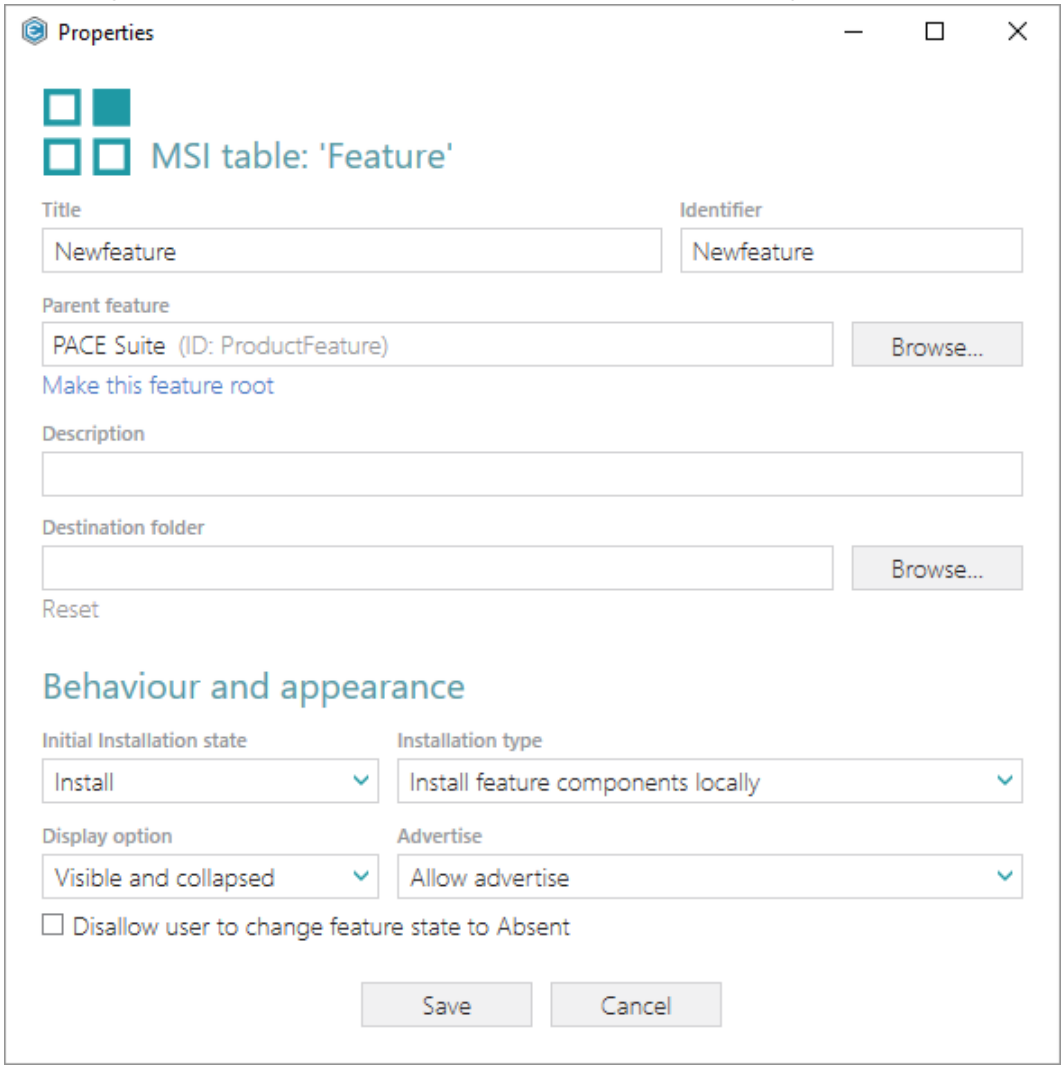


- Select **New feature** from the context menu of a feture, where you want to create a new feature. For creating a root feature, use any feature and later you can clean-up its parent feature.

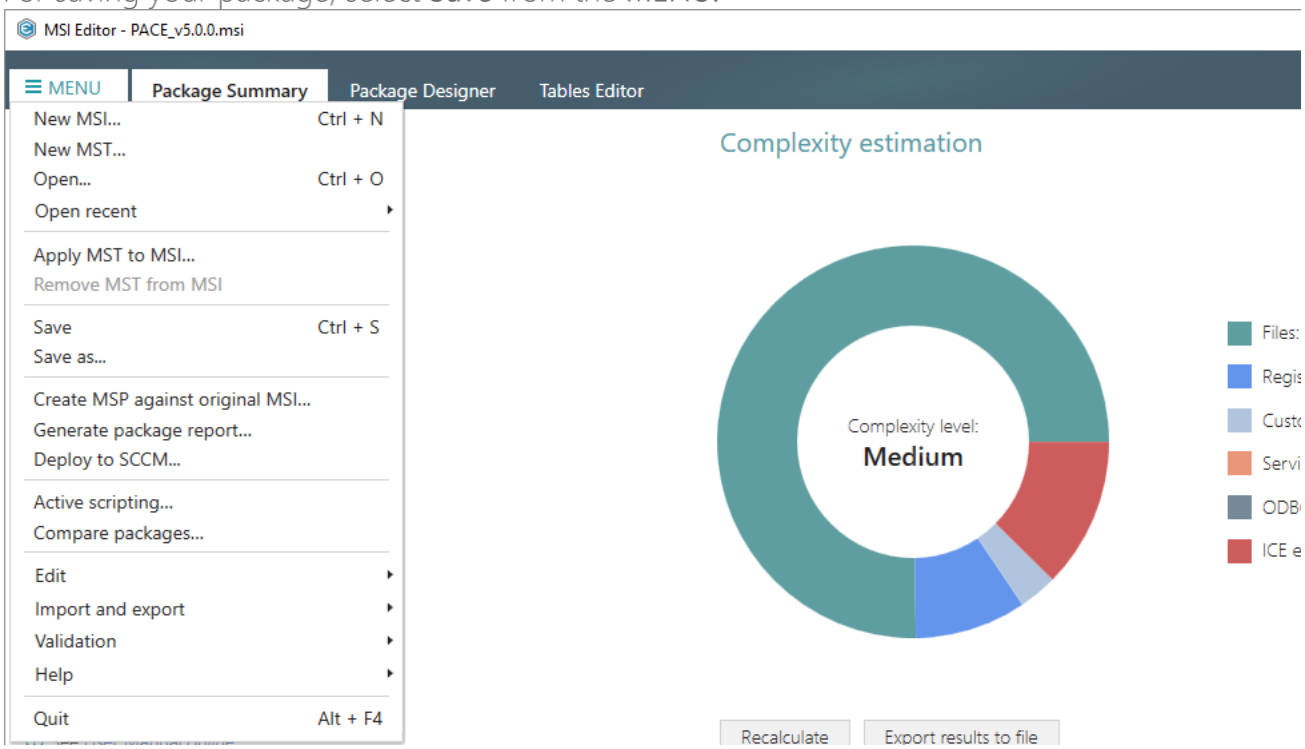


- Using the **Properties** window, update a feature title, identifier and select the necessary behavior and appearance settings. For creating a root feature, click on the **Make this feature**

root link, located below the Parent feature field. Afterwards, click Save.



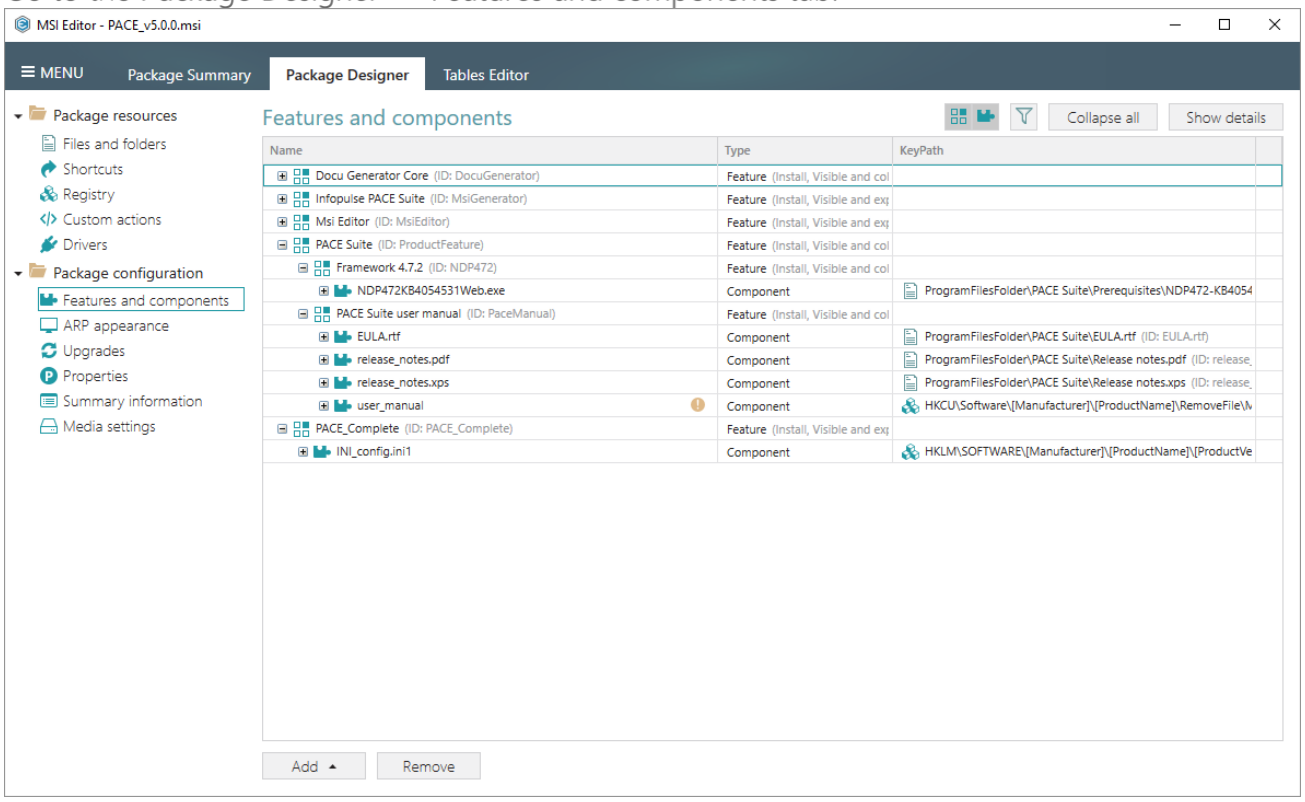
- For saving your package, select Save from the MENU.



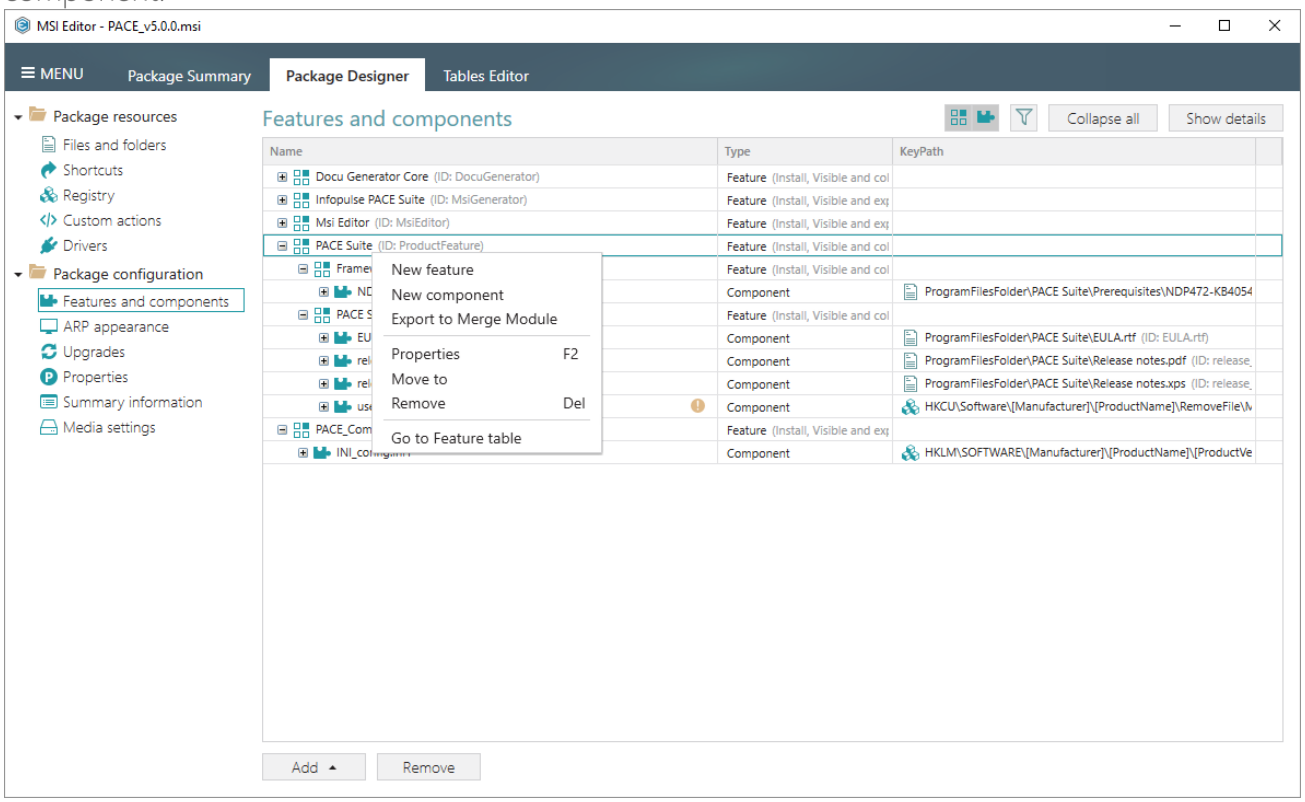
3.7.9.2 Add Component

Add a new component to one or many features and set the component properties.

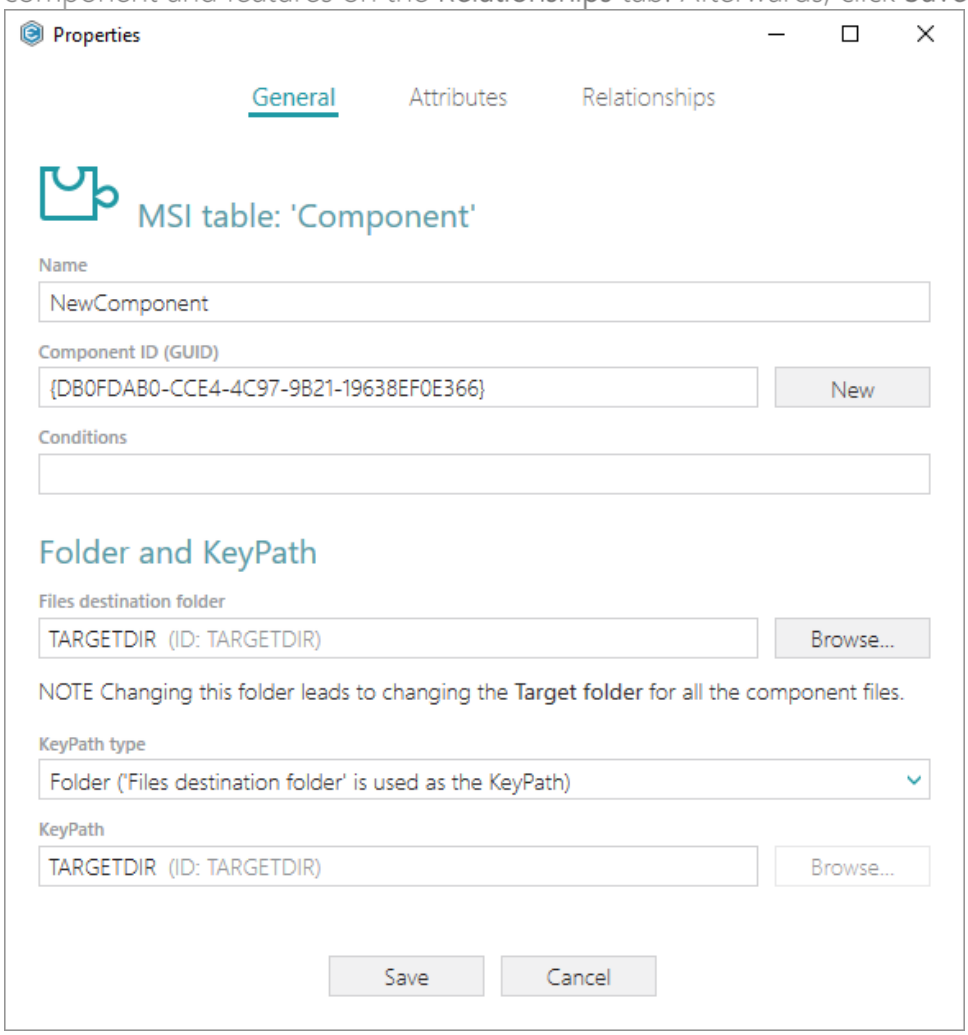
- Go to the Package Designer -> Features and components tab.



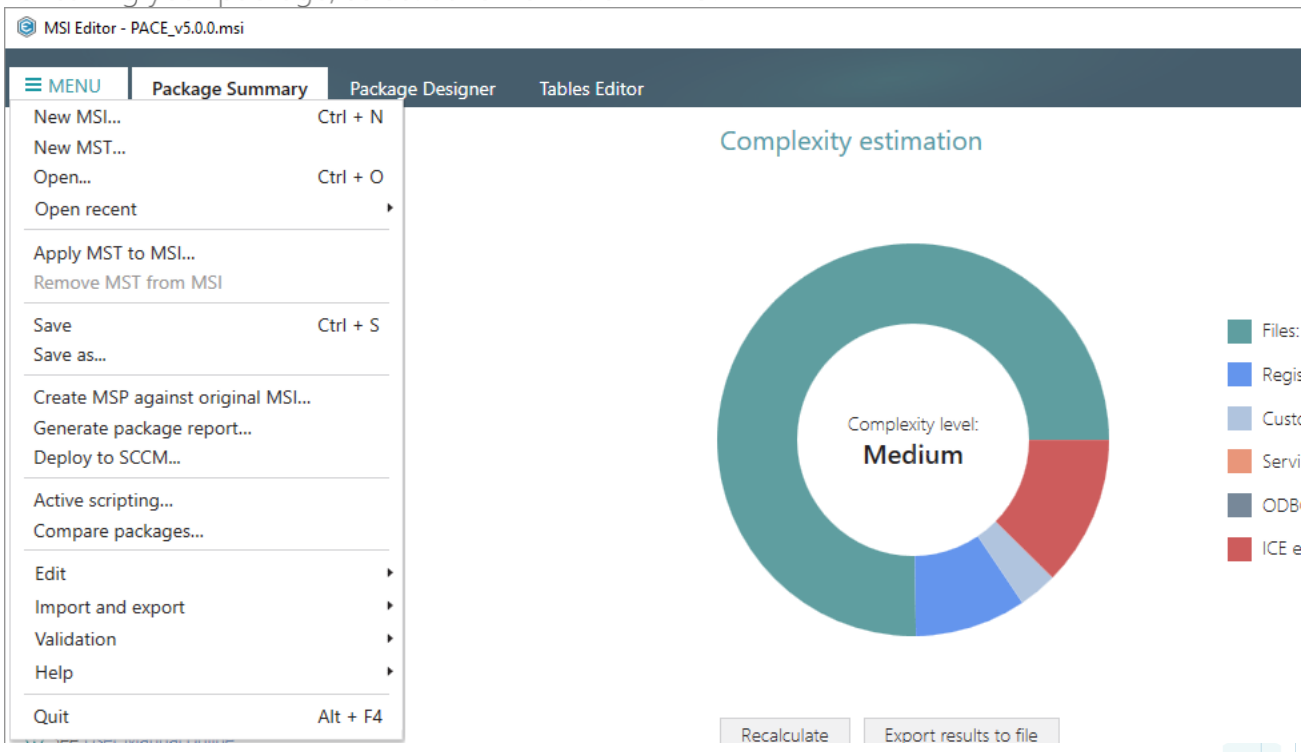
- Select New component from the context menu a feature, where you want to create a new component.



- Using the Properties window, update a component name, GUID and select the necessary installation conditions, target folder. Note that to select a key path from files or registry, this component should contain these resources. Switch to the Attributes tab to manage component attributes and define new or change existing relationships between this component and features on the Relationships tab. Afterwards, click Save.



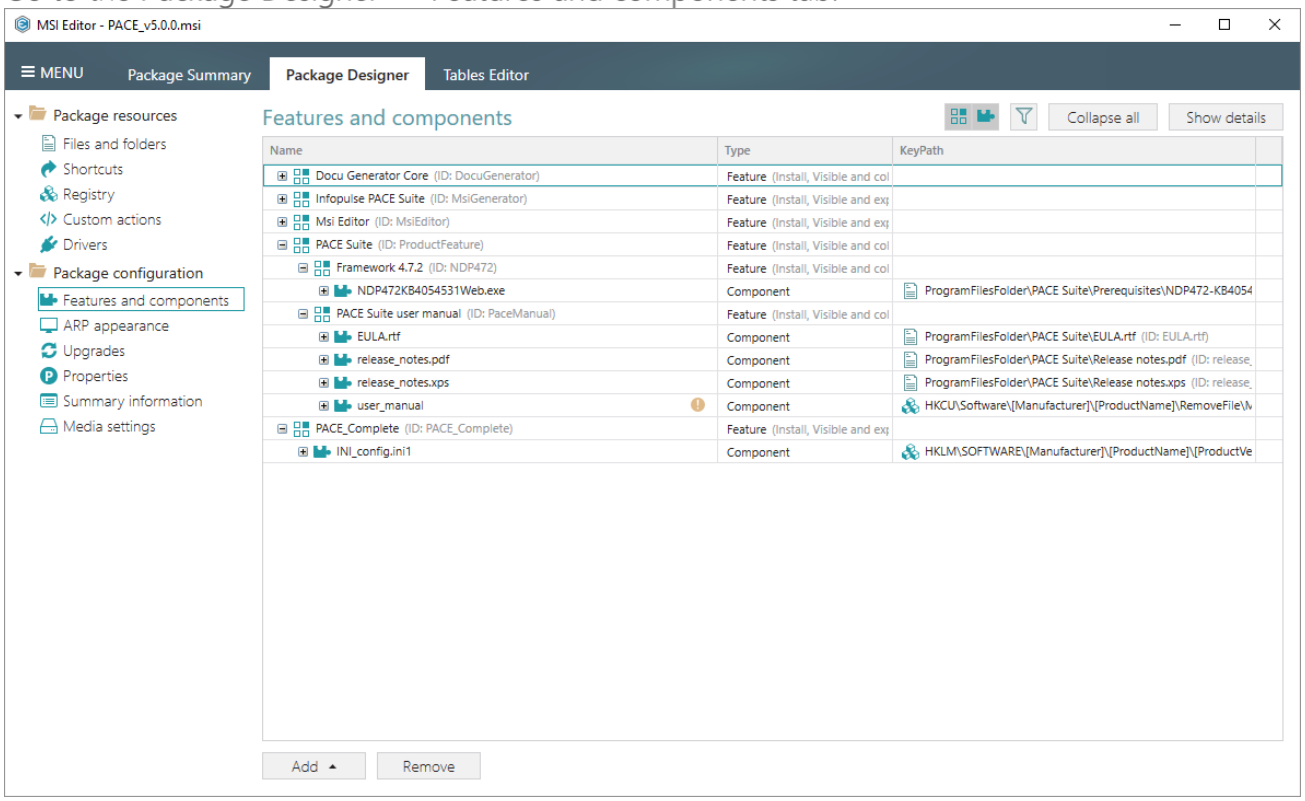
- For saving your package, select Save from the MENU.



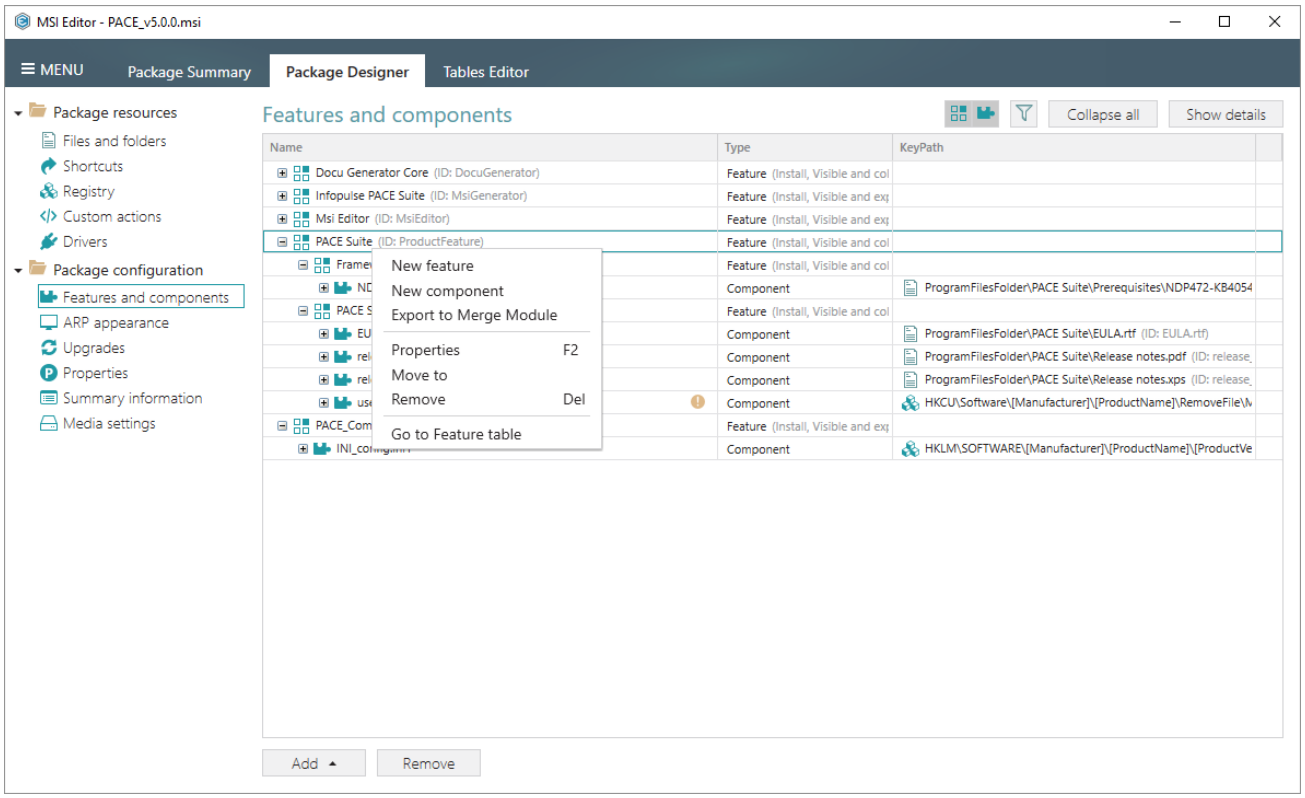
3.7.9.3 Move Feature, Component

Move a feature or a component to another feature.

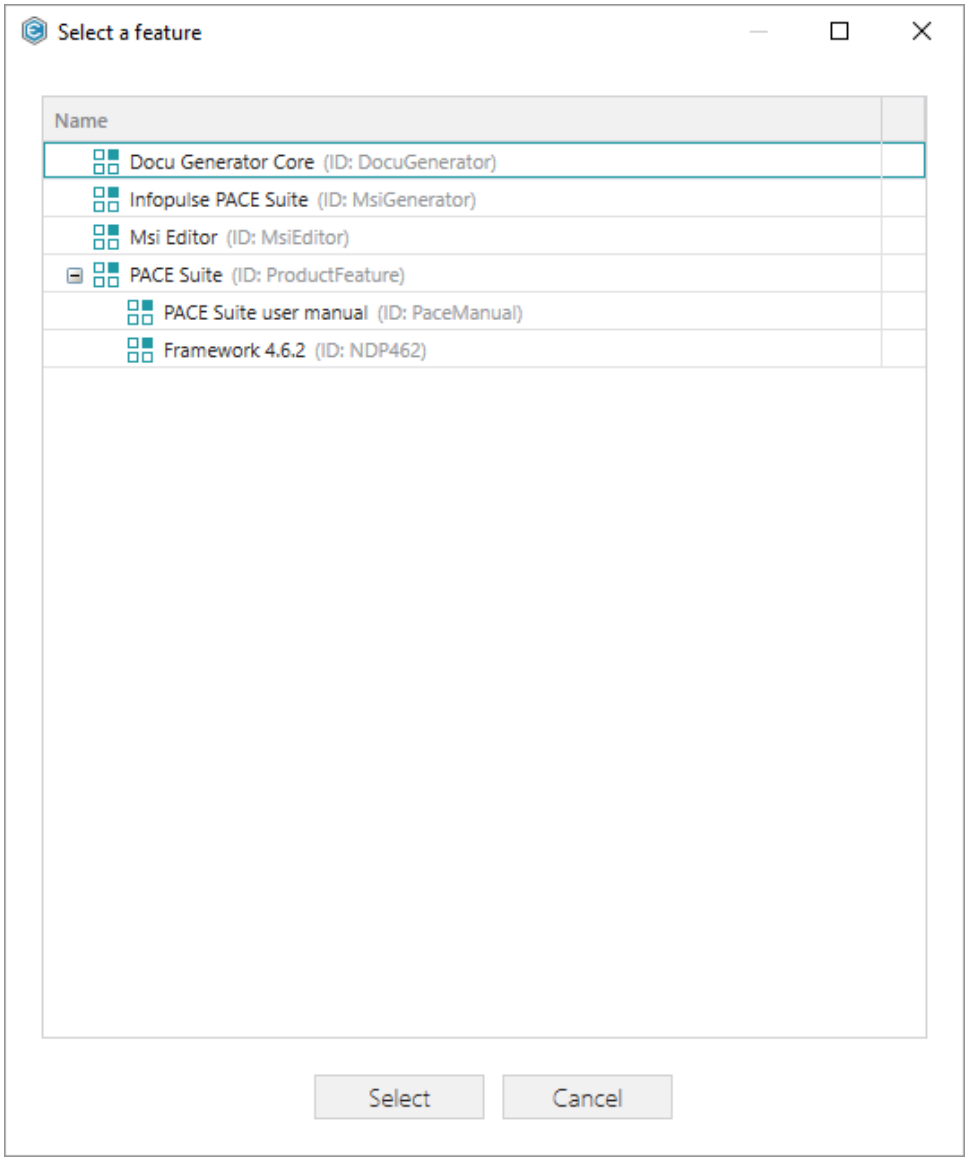
- Go to the Package Designer -> Features and components tab.



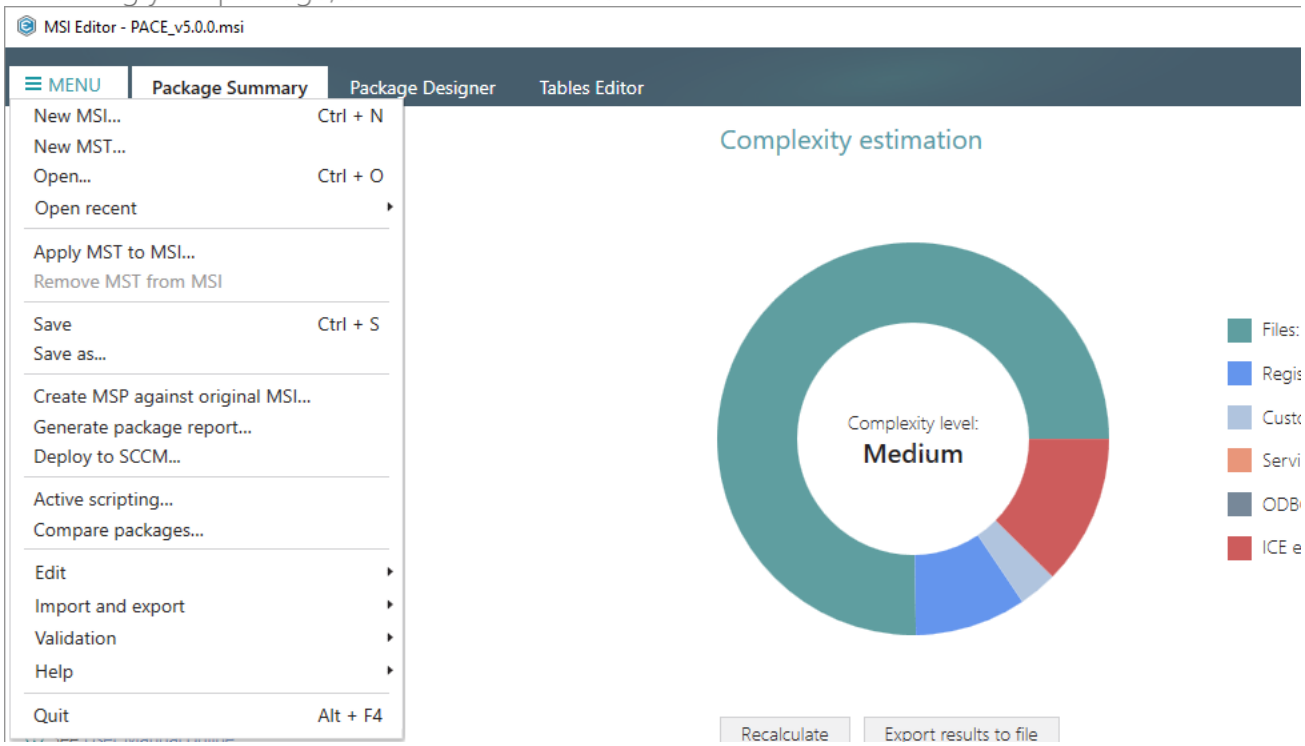
- Select Move to from the context menu of a feature or component, which you want to move to another feature.



- Choose a destionation feature from the tree and click Select.



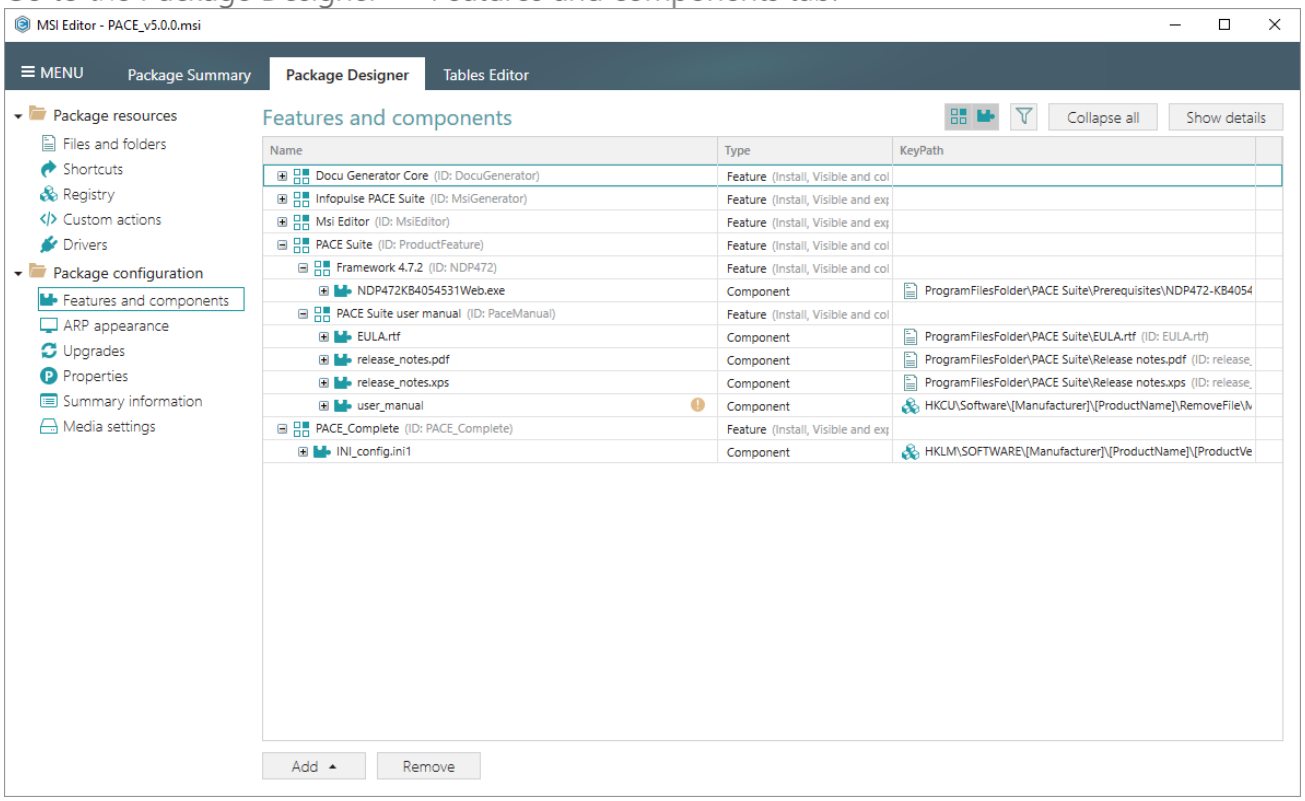
- For saving your package, select Save from the MENU.



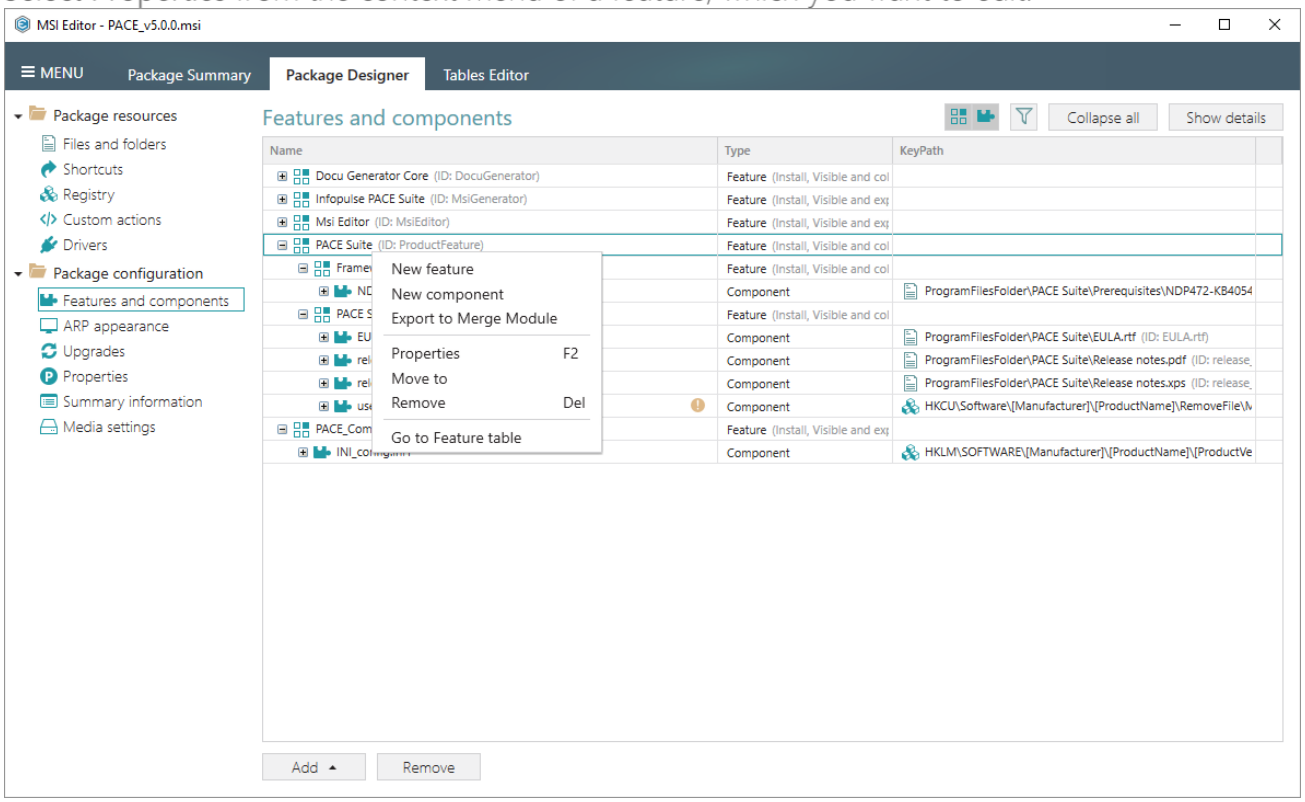
3.7.9.4 Edit Feature Properties

Edit feature properties such as ID, title, folder, installation state, type, display options and advertisement.

- Go to the Package Designer -> Features and components tab.

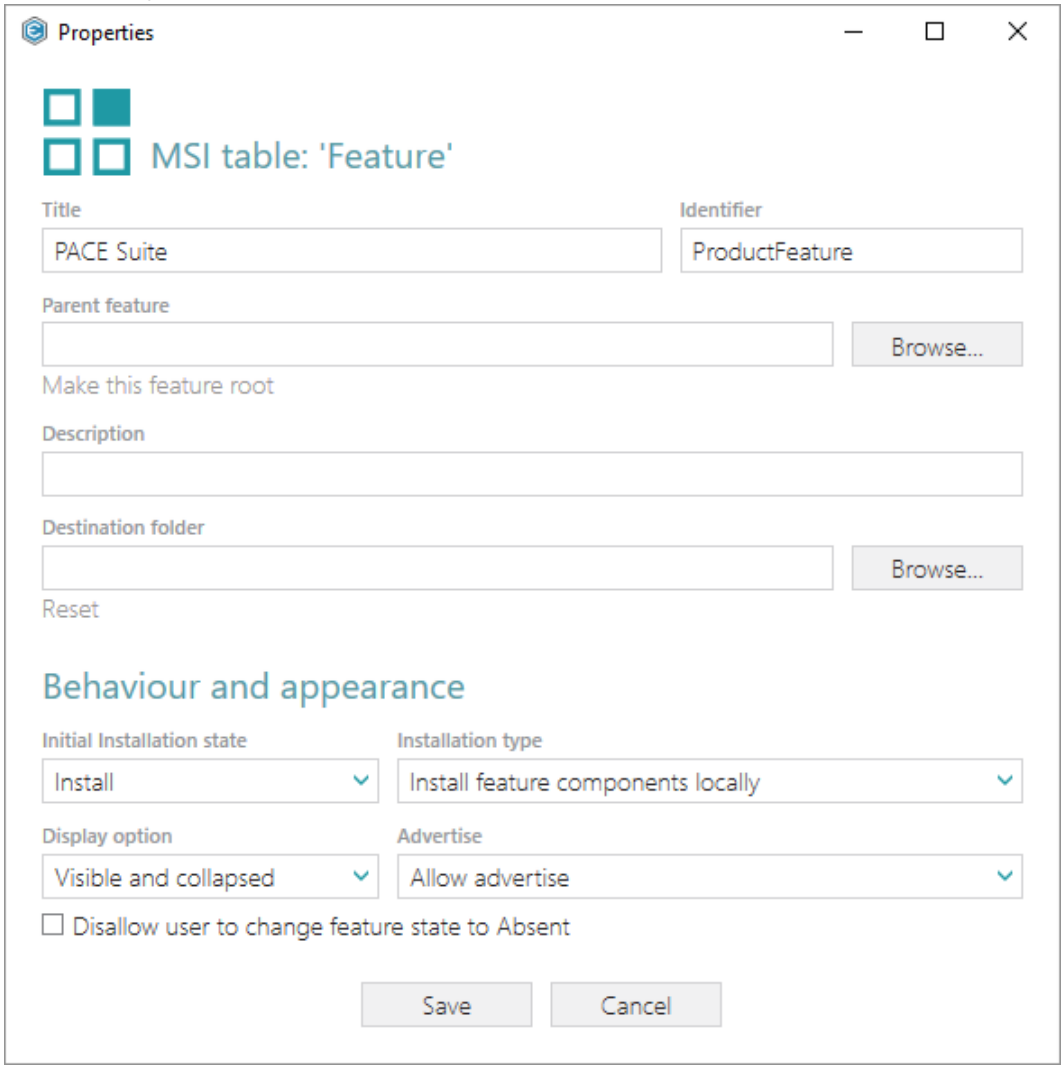


- Select Properties from the context menu of a feature, which you want to edit.

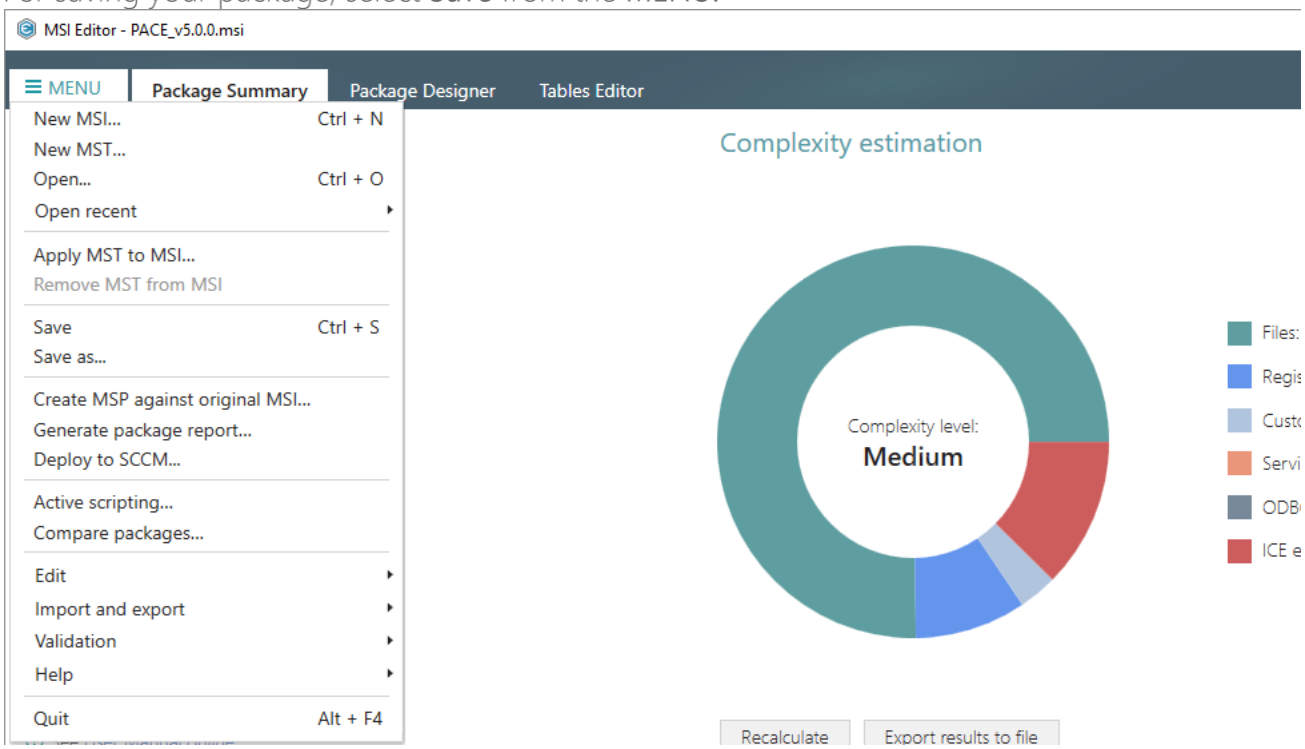


- Using the Properties window, update the necessary feature properties, such as a feature title, identifier, parent feature, installation state and type, display option and advertisement.

Afterwards, click Save.



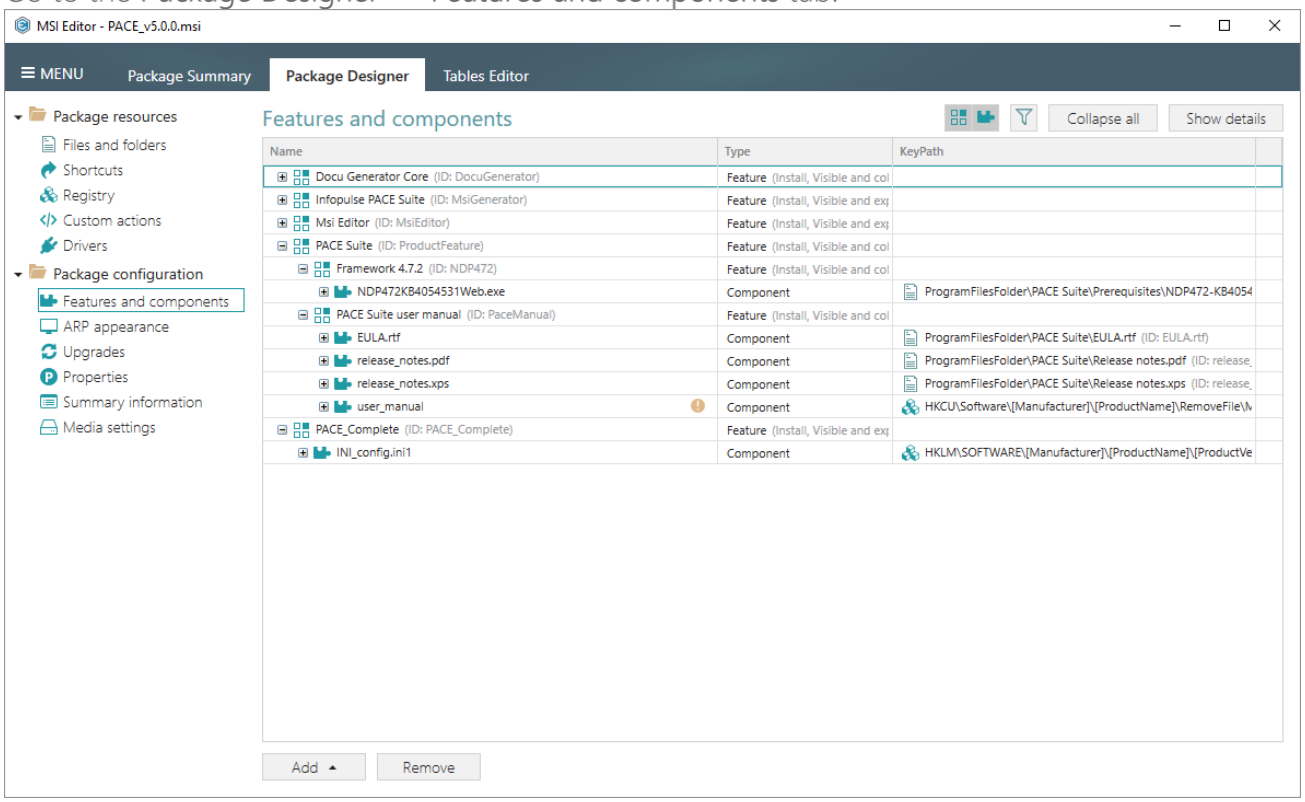
- For saving your package, select Save from the MENU.



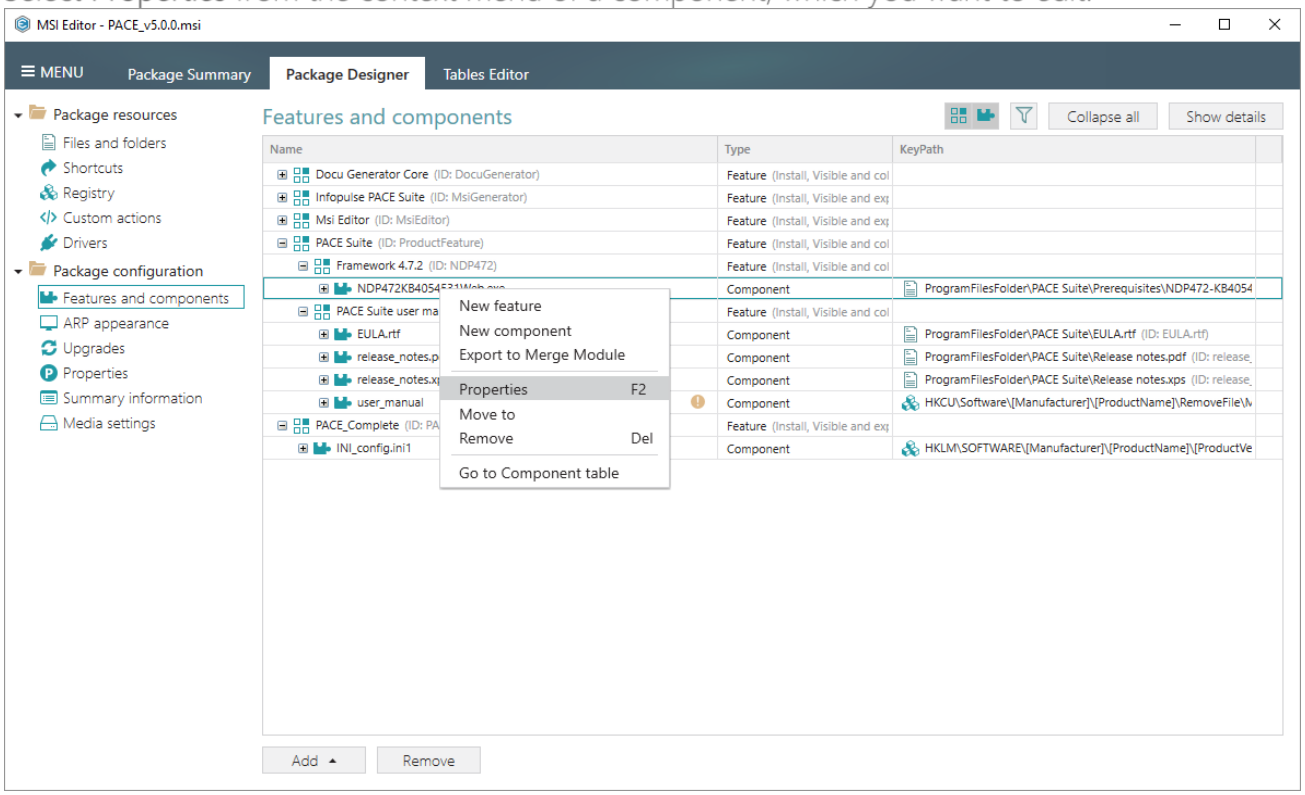
3.7.9.5 Edit Component Properties

Edit component properties such as name, GUID, conditions, key path, attributes and relationship with features.

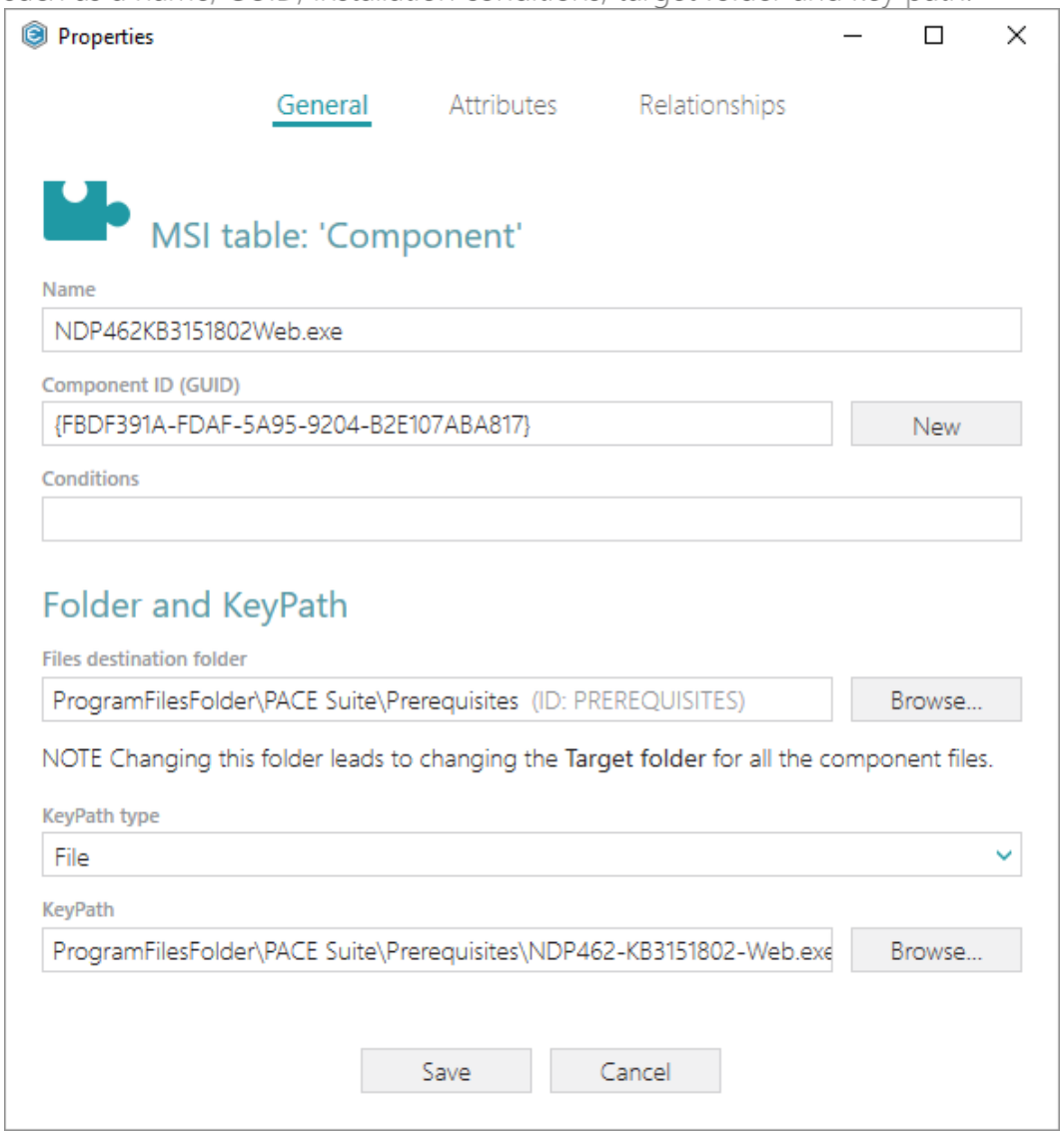
- Go to the Package Designer -> Features and components tab.



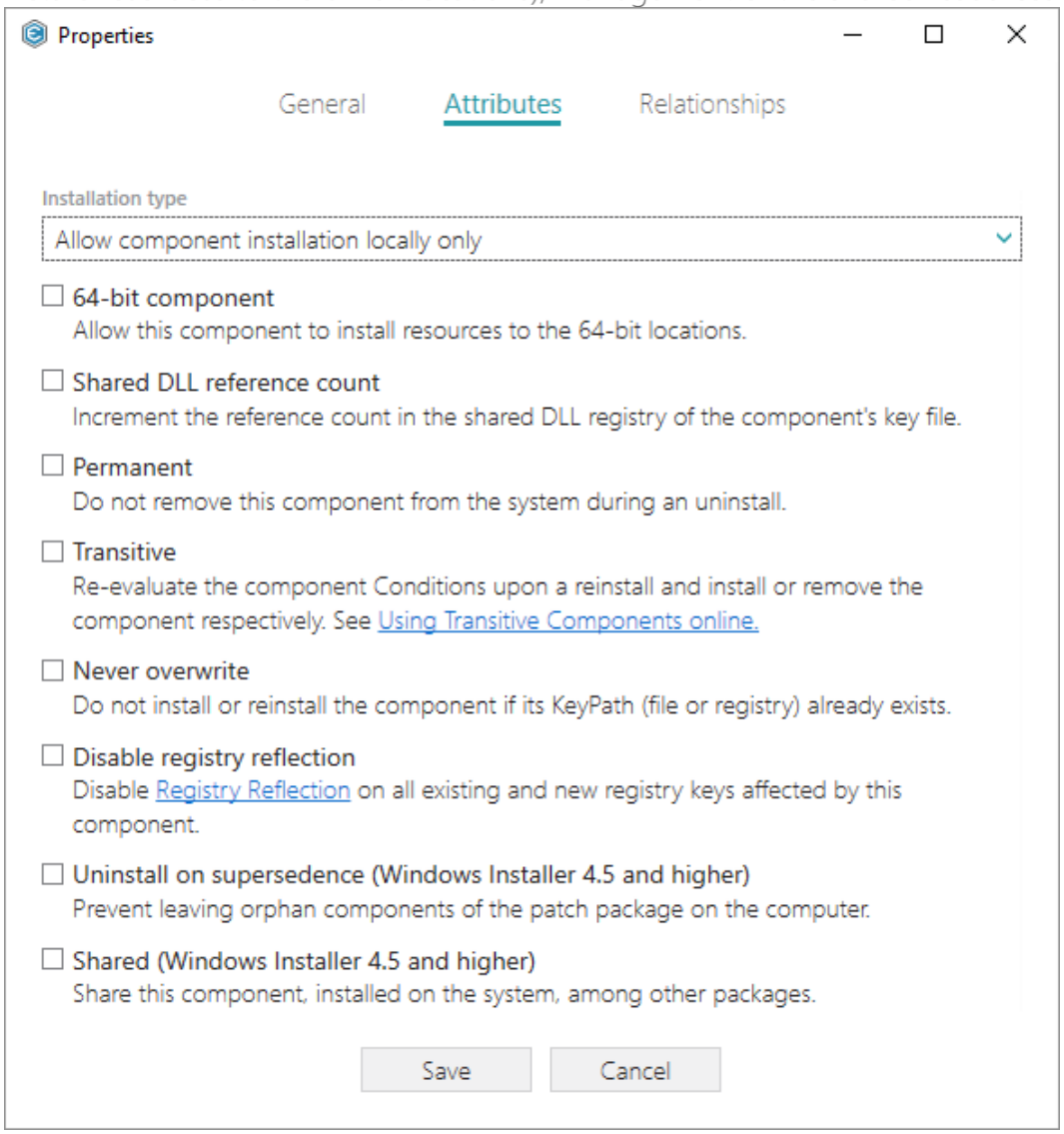
- Select Properties from the context menu of a component, which you want to edit.



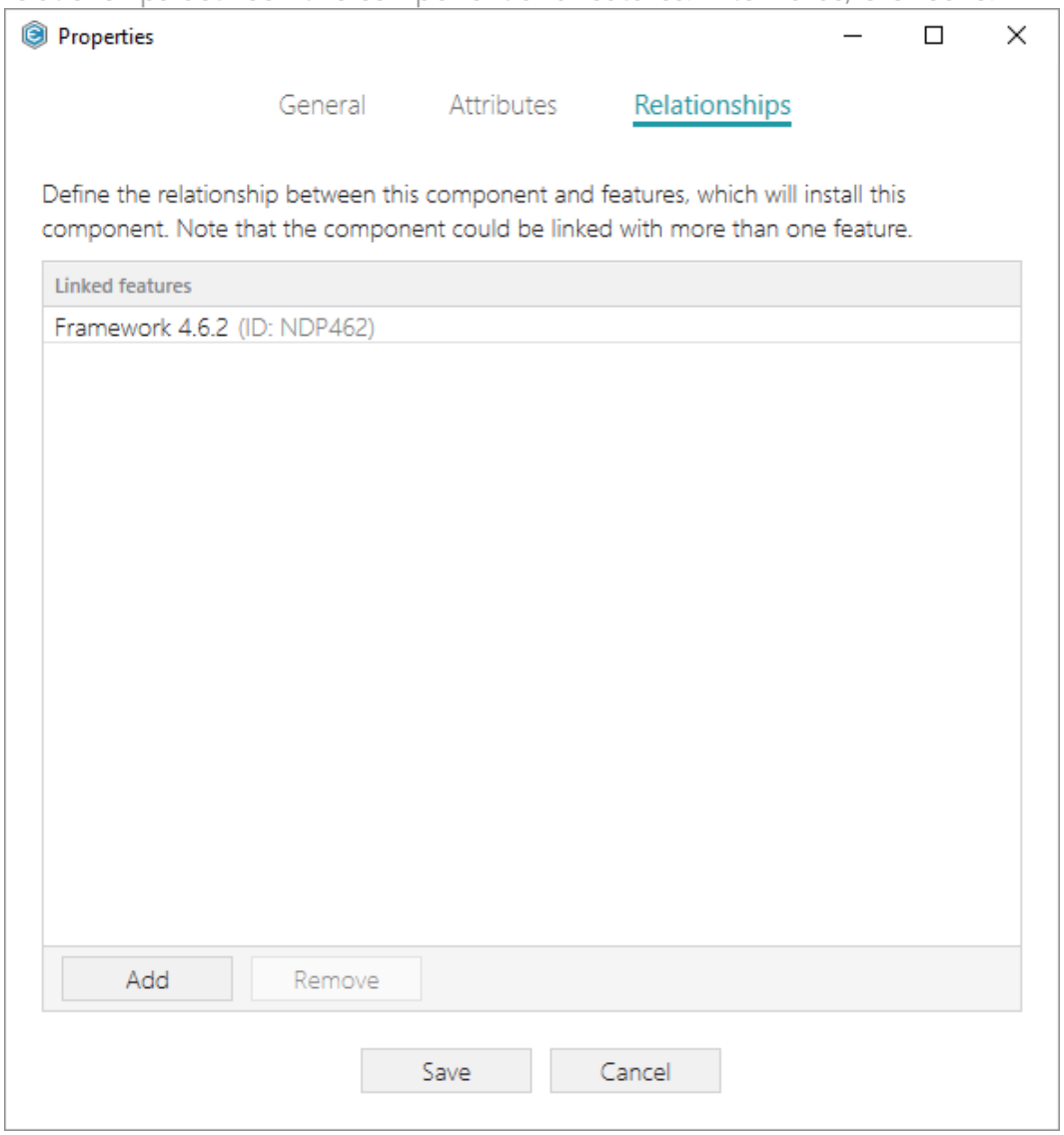
- On the **General** tab of the **Properties** window, update the necessary component properties, such as a name, GUID, installation conditions, target folder and key path.



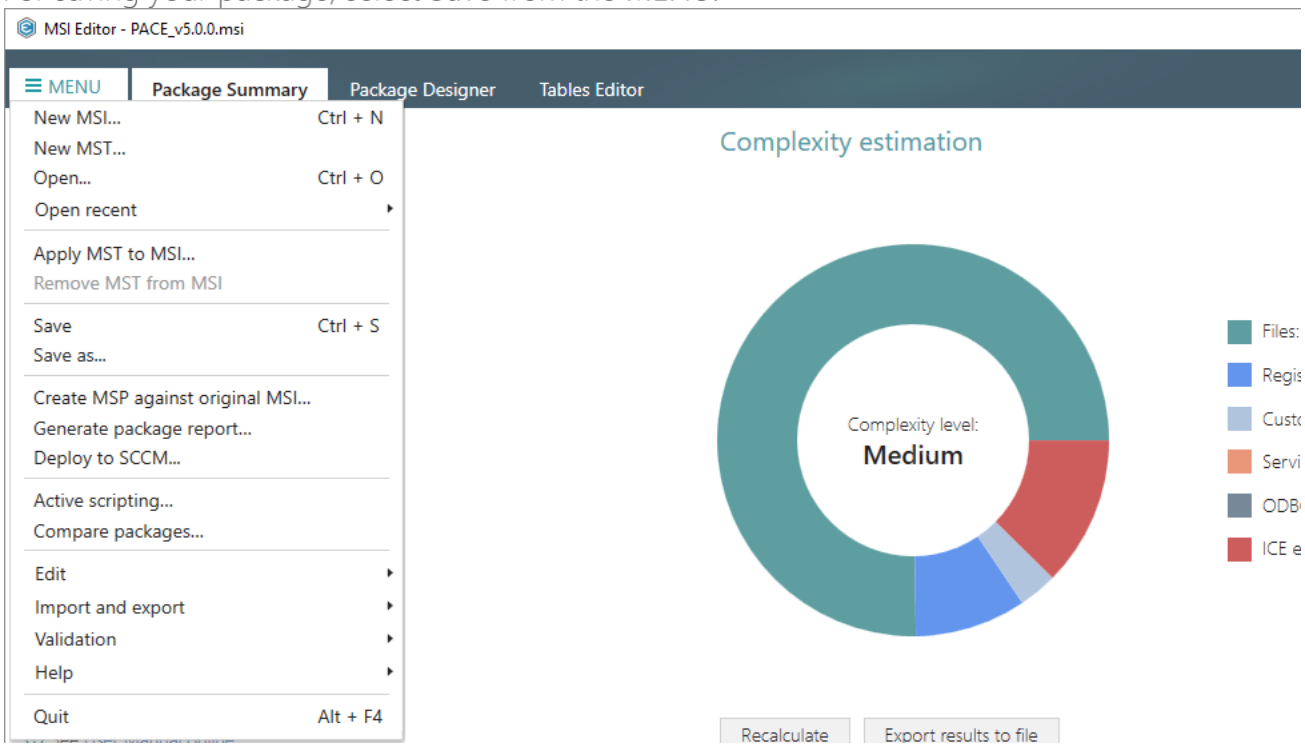
- On the **Attributes** tab of the **Properties** window you can manage component attributes such as component installation type, bitness (set the **64-bit component** attribute if your component installs resoruces to the 64-bit locations), management of the shared resources and so on.



- The **Relationships** tab of the **Properties** window allows changing existing and adding new relationships between this component and features. Afterwards, click **Save**.



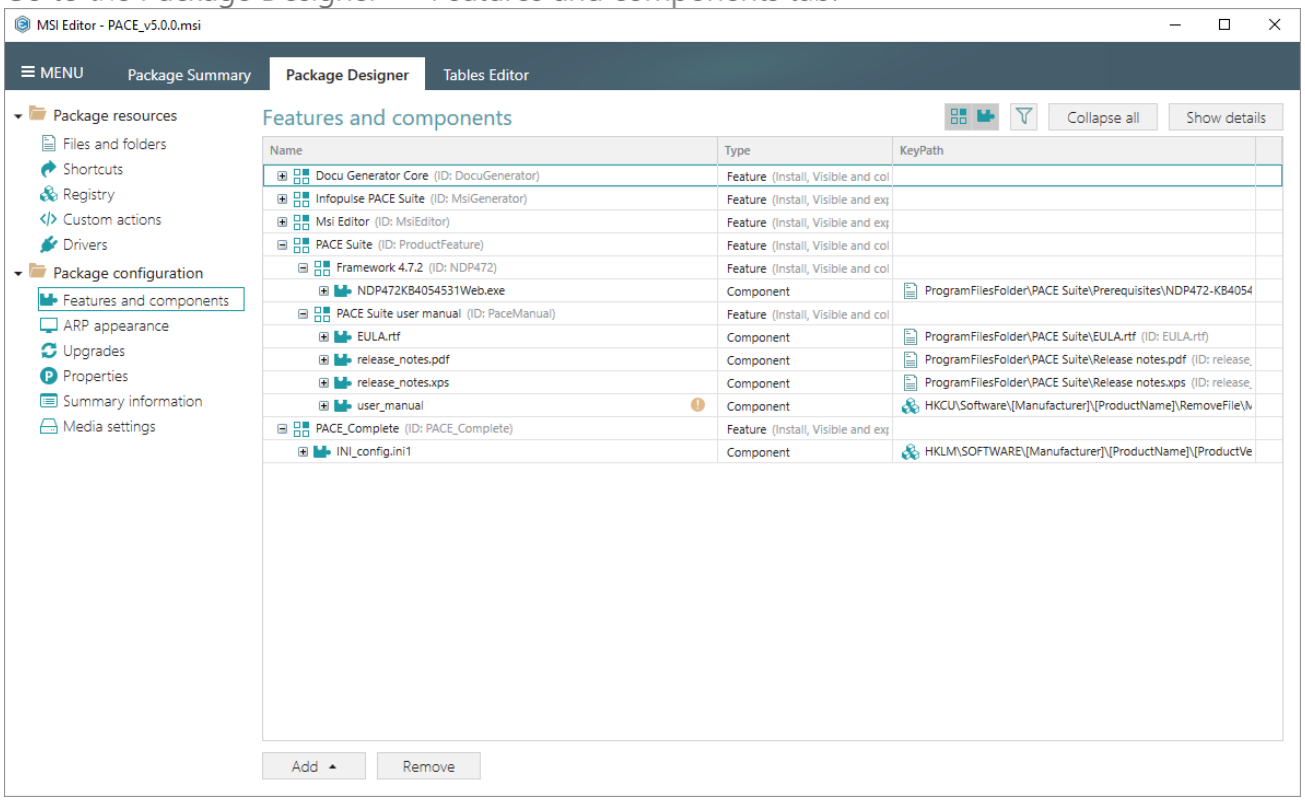
- For saving your package, select **Save** from the **MENU**.



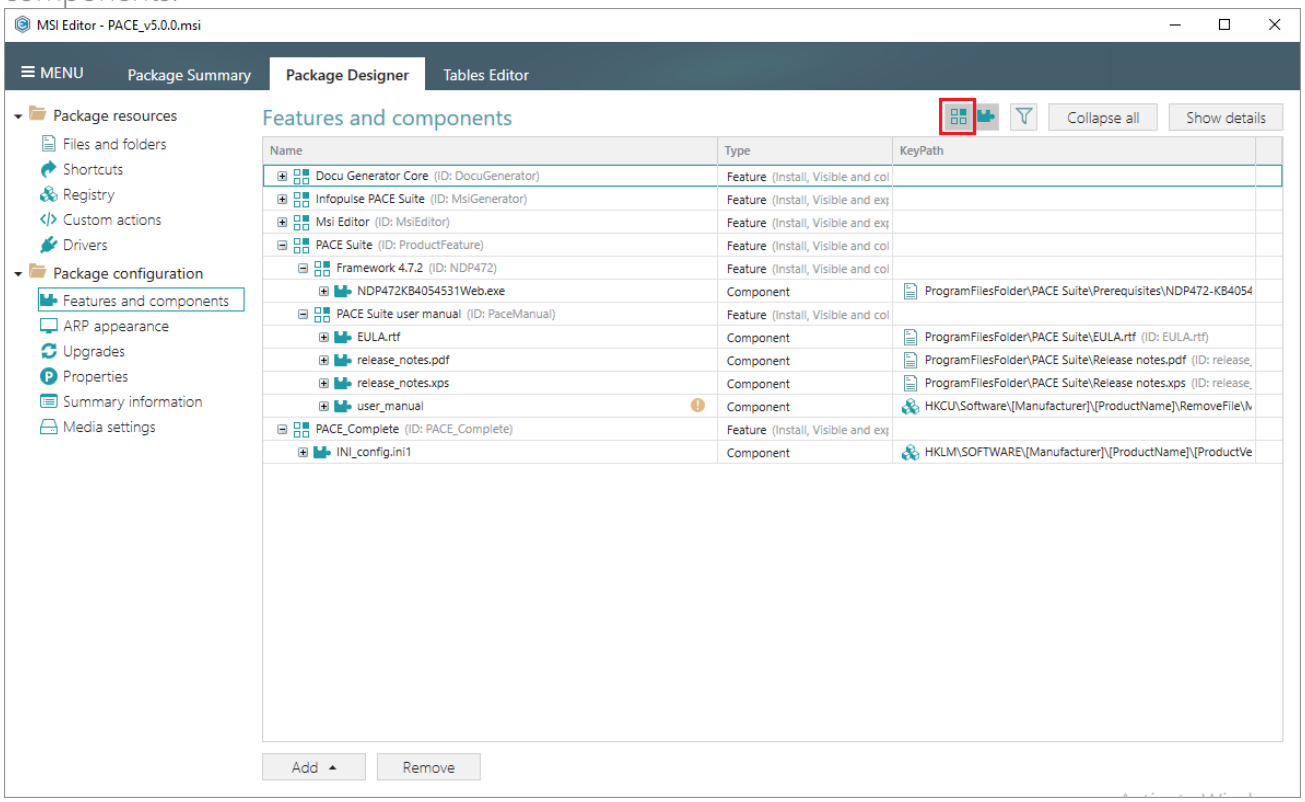
3.7.9.6 Detect Empty Components

Find empty components with no linked resources in order to remove them.

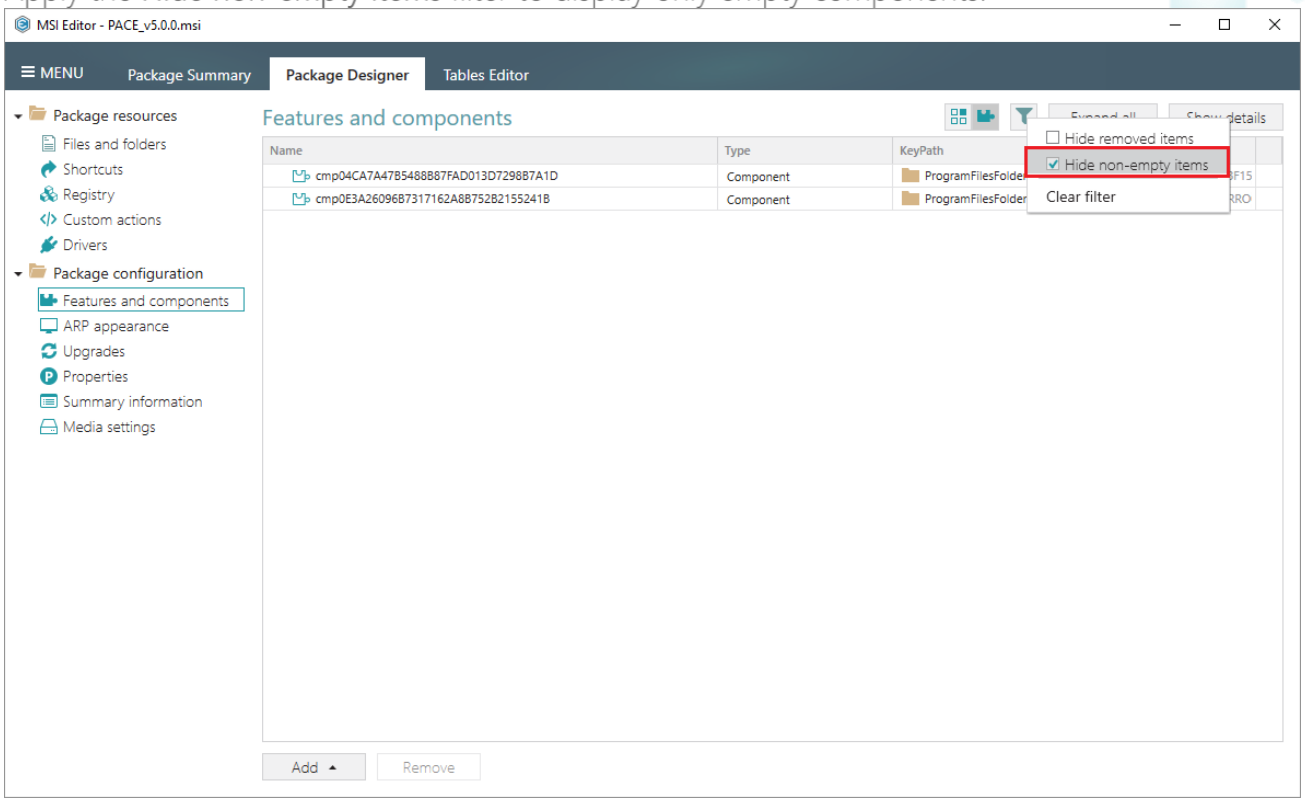
- Go to the Package Designer -> Features and components tab.



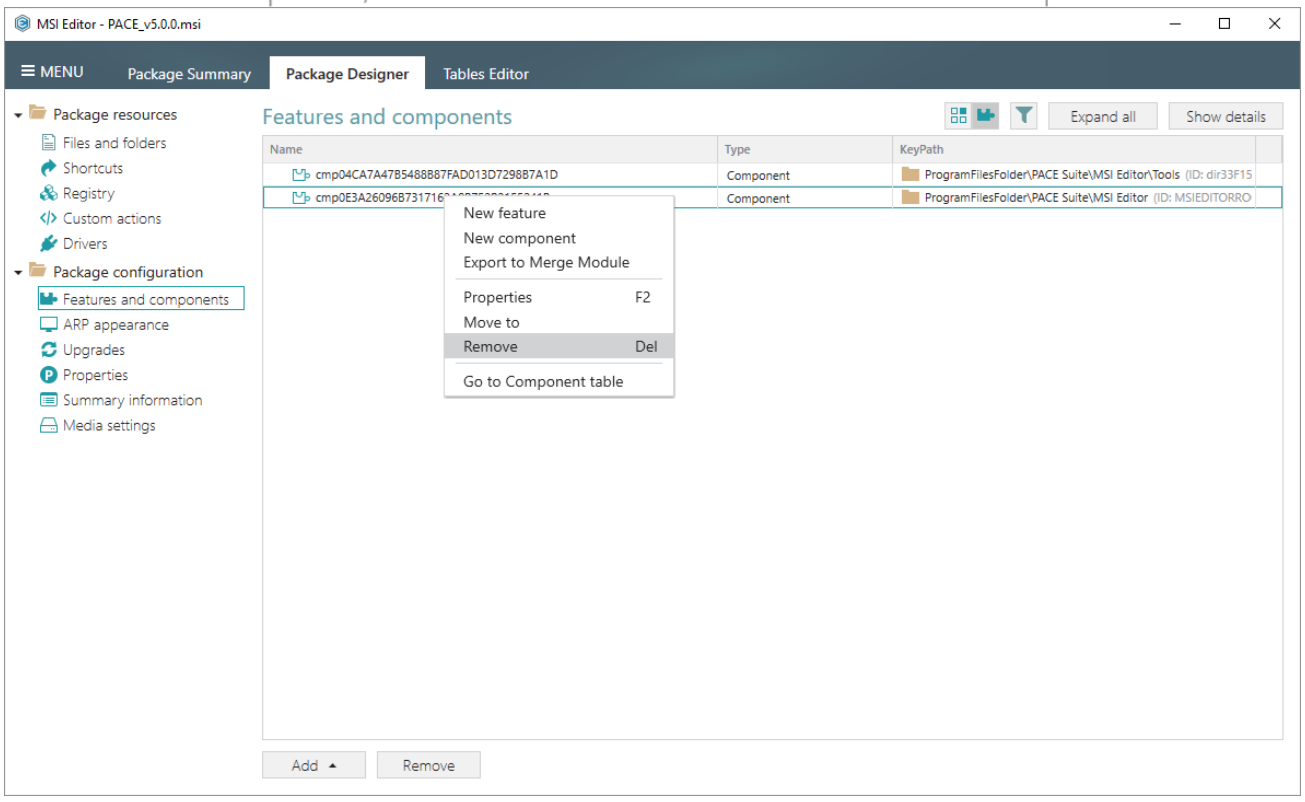
- Click on the 'feature' icon , located near the Filters, to hide all features and display only components.



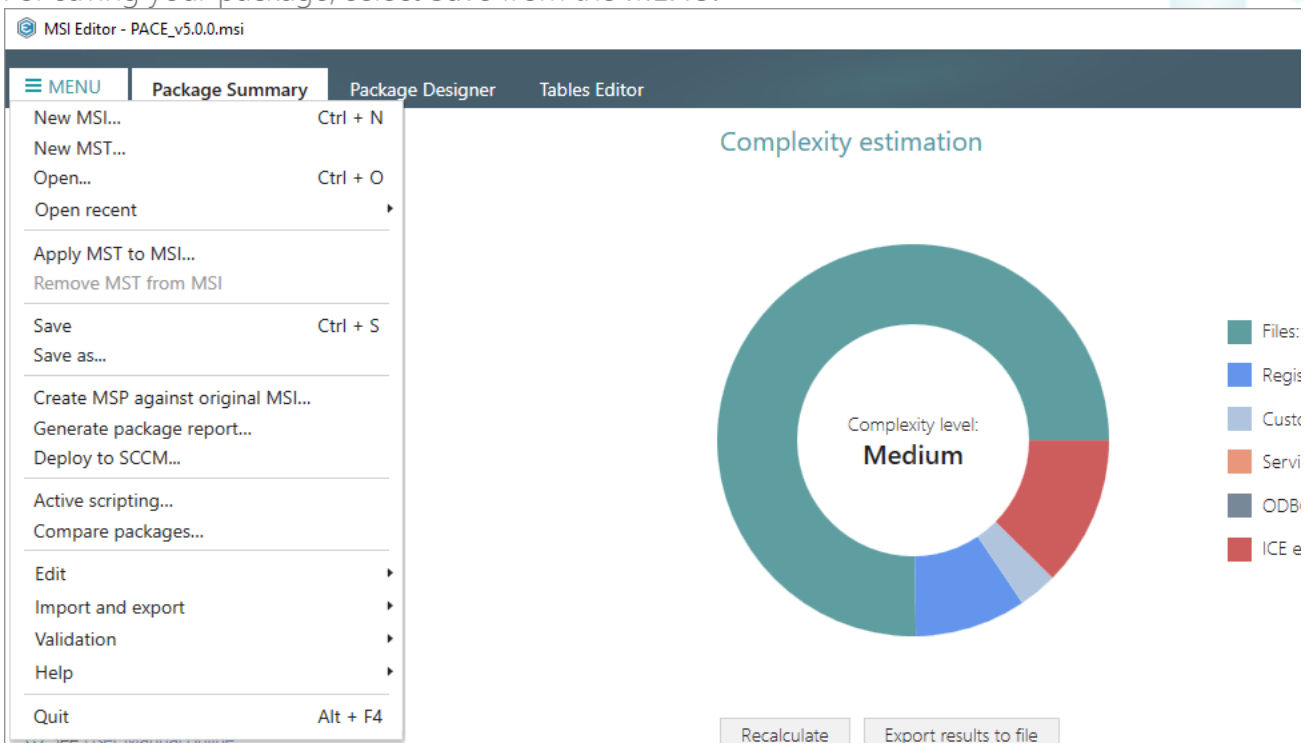
- Apply the **Hide non-empty items** filter to display only empty components.



- To remove the component, select **Remove** from the context menu of the component.



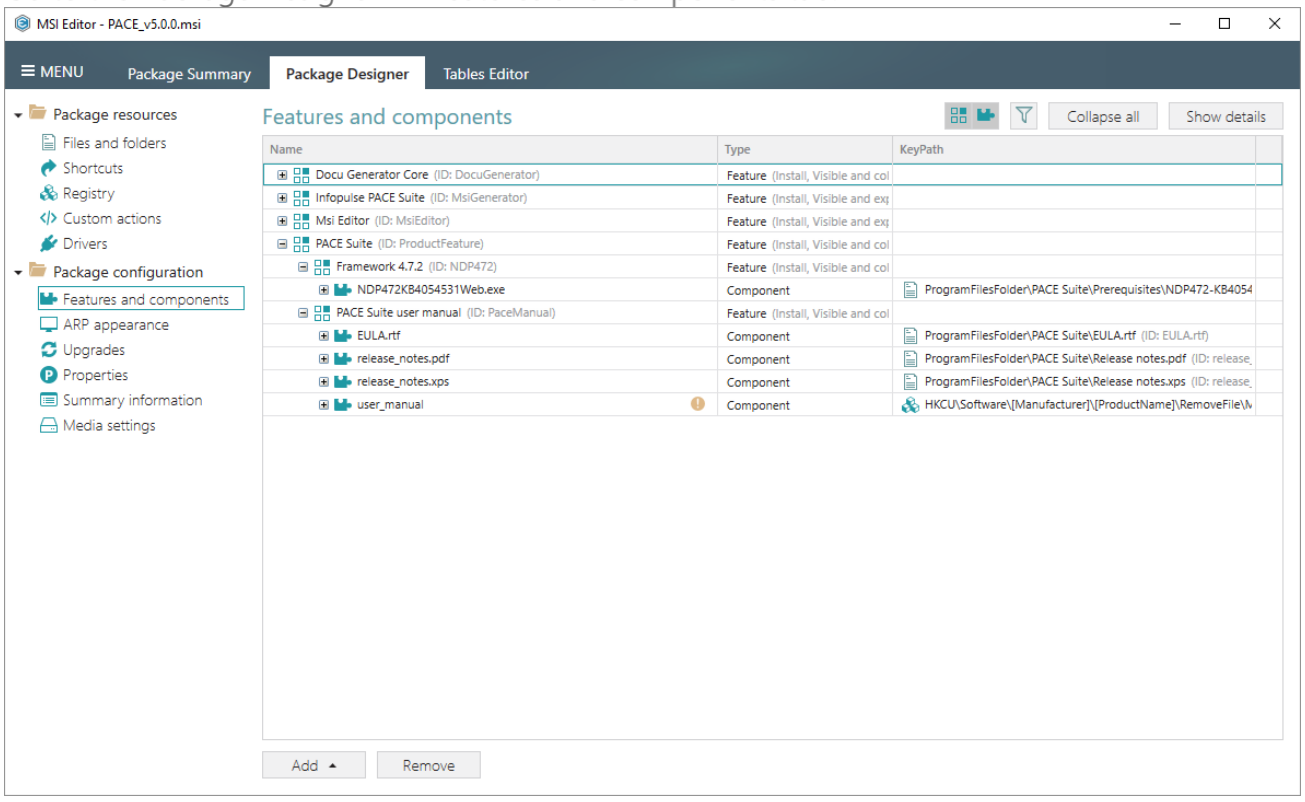
- For saving your package, select Save from the MENU.



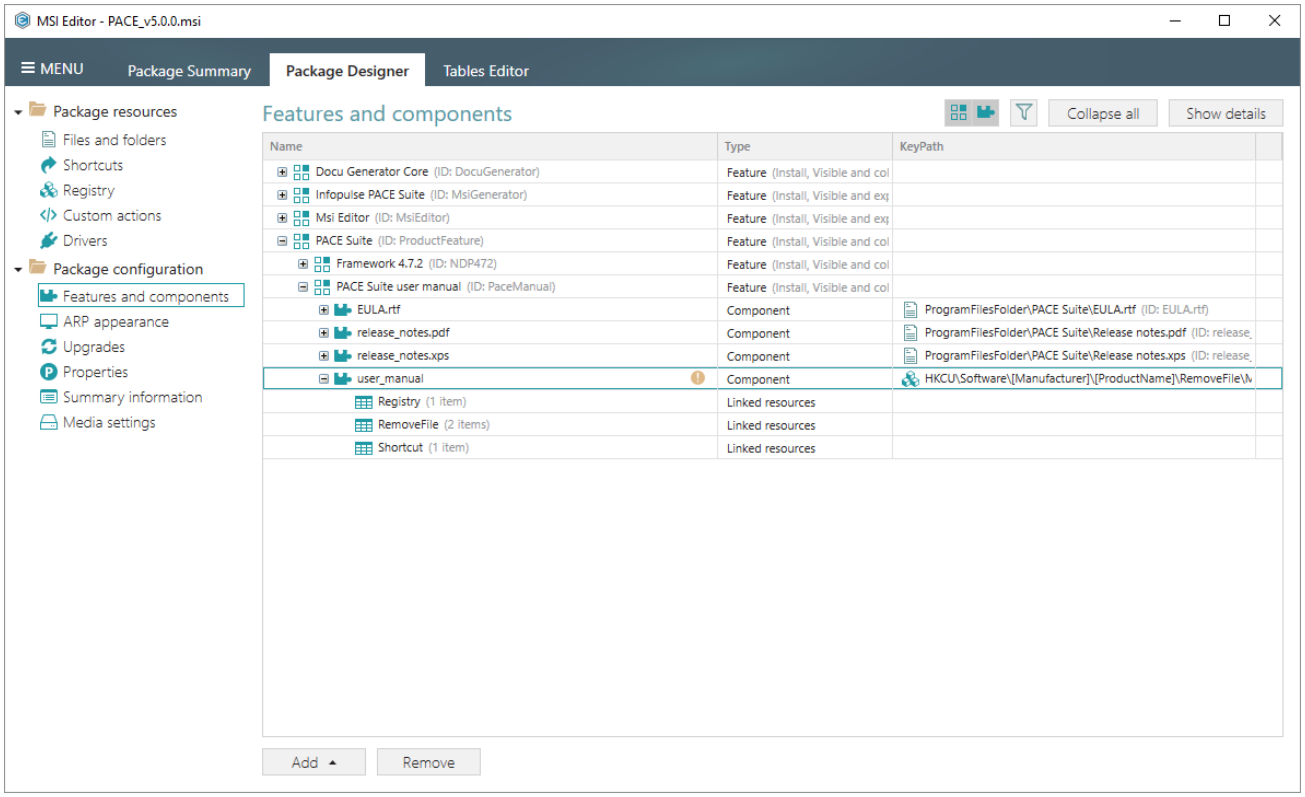
3.7.9.7 Review Component Resources

Review what resources (files, registry, shortcuts, etc.) are linked with the component.

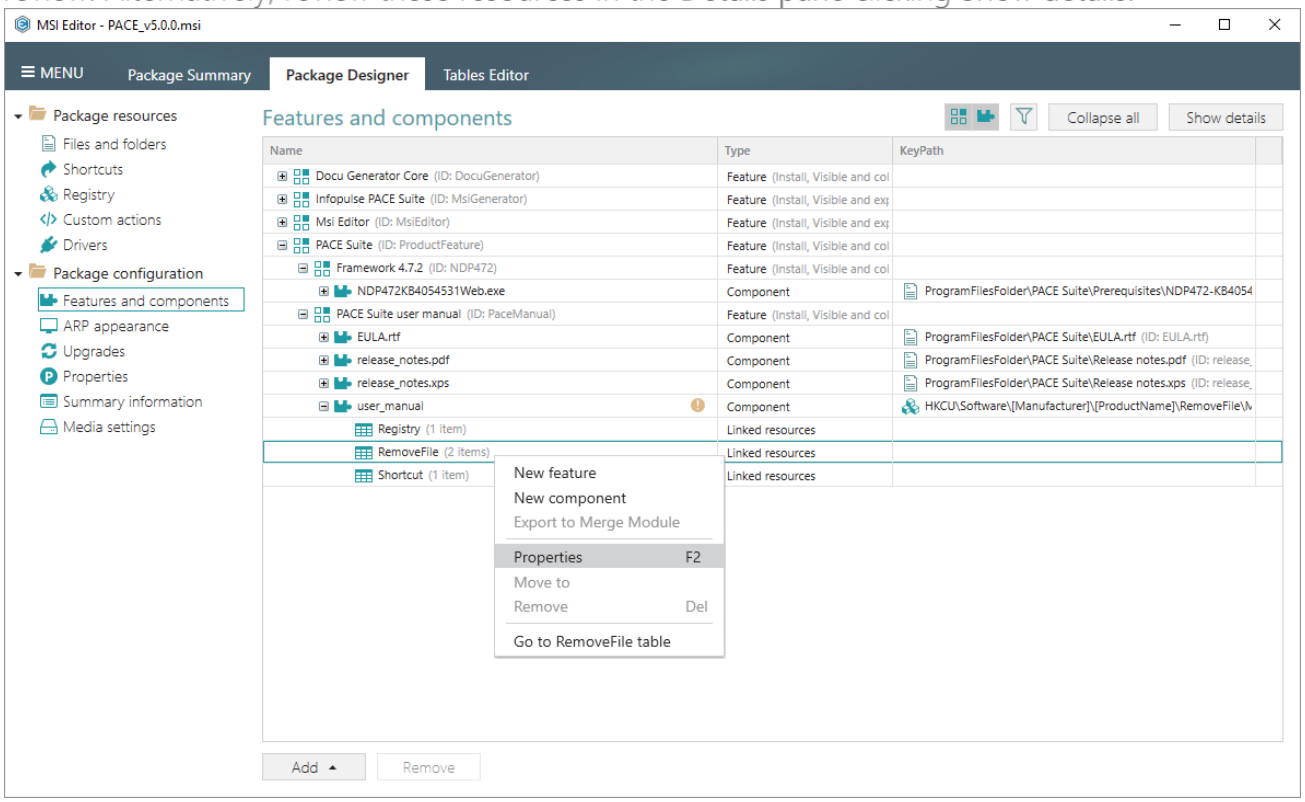
- Go to the Package Designer -> Features and components tab.



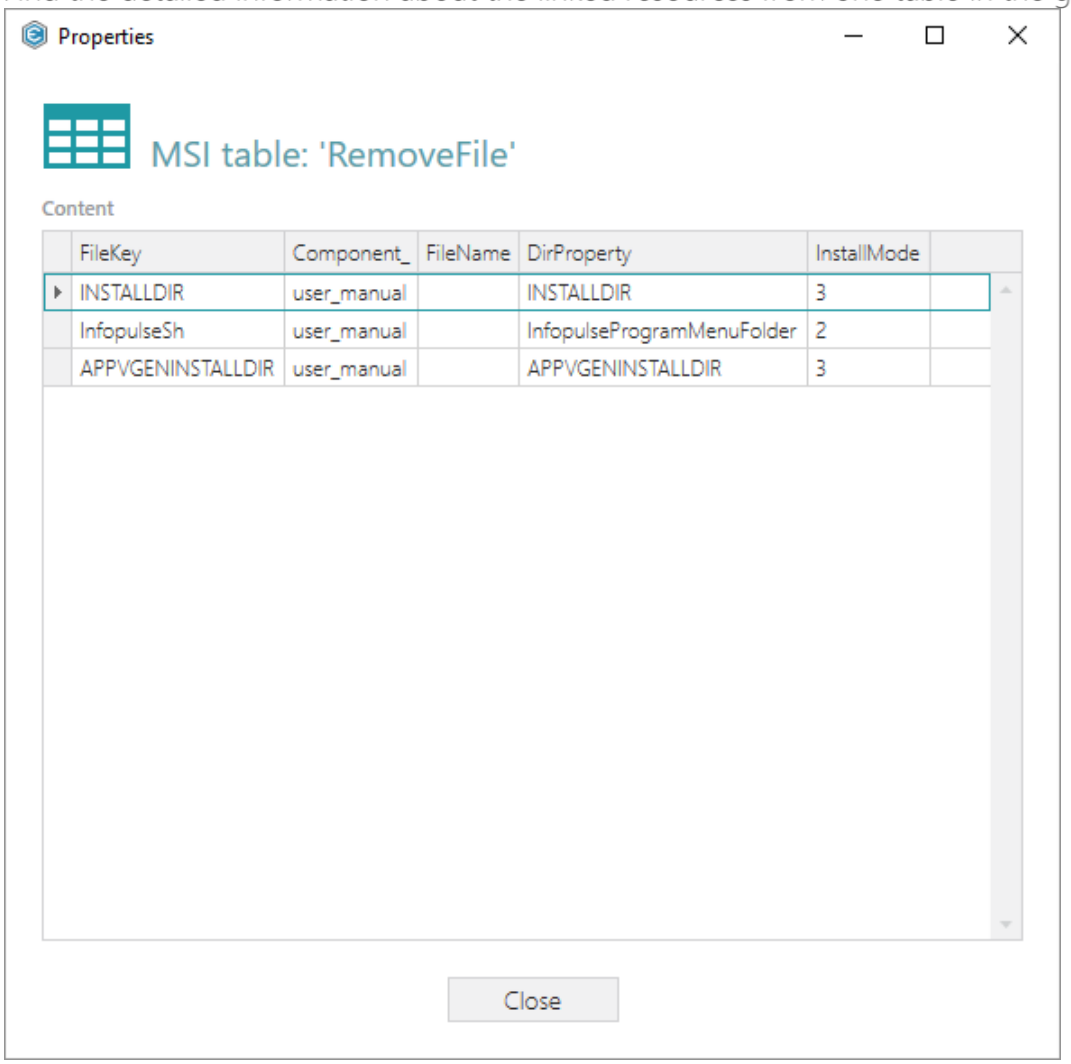
- Find the necessary component and click ⊞ to expand the component node and to review resources, linked with this component. Note that all resources are grouped by tables, from which these resources were read.



- Select **Properties** from the context menu of a group of component resources you want to review. Alternatively, review these resources in the **Details pane** clicking **Show details**.



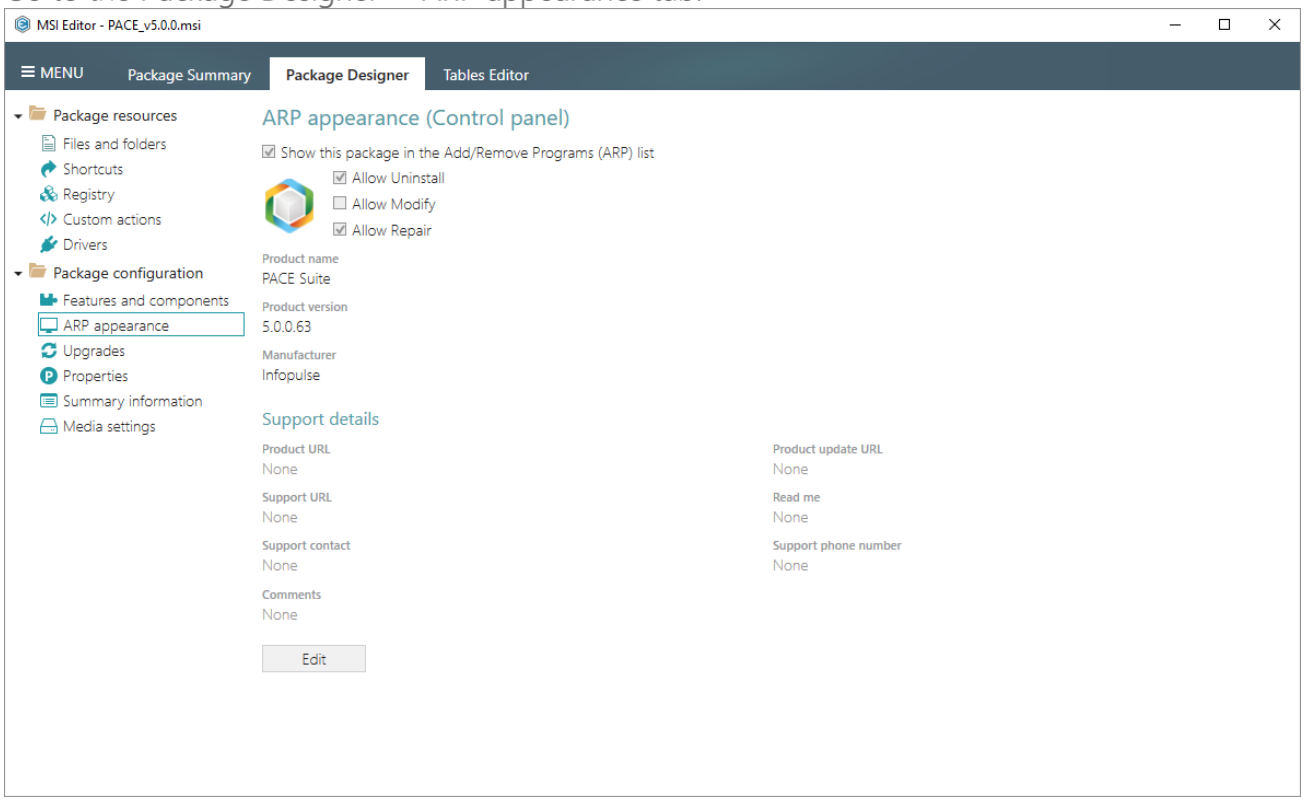
- Find the detailed information about the linked resources from one table in the grid view.



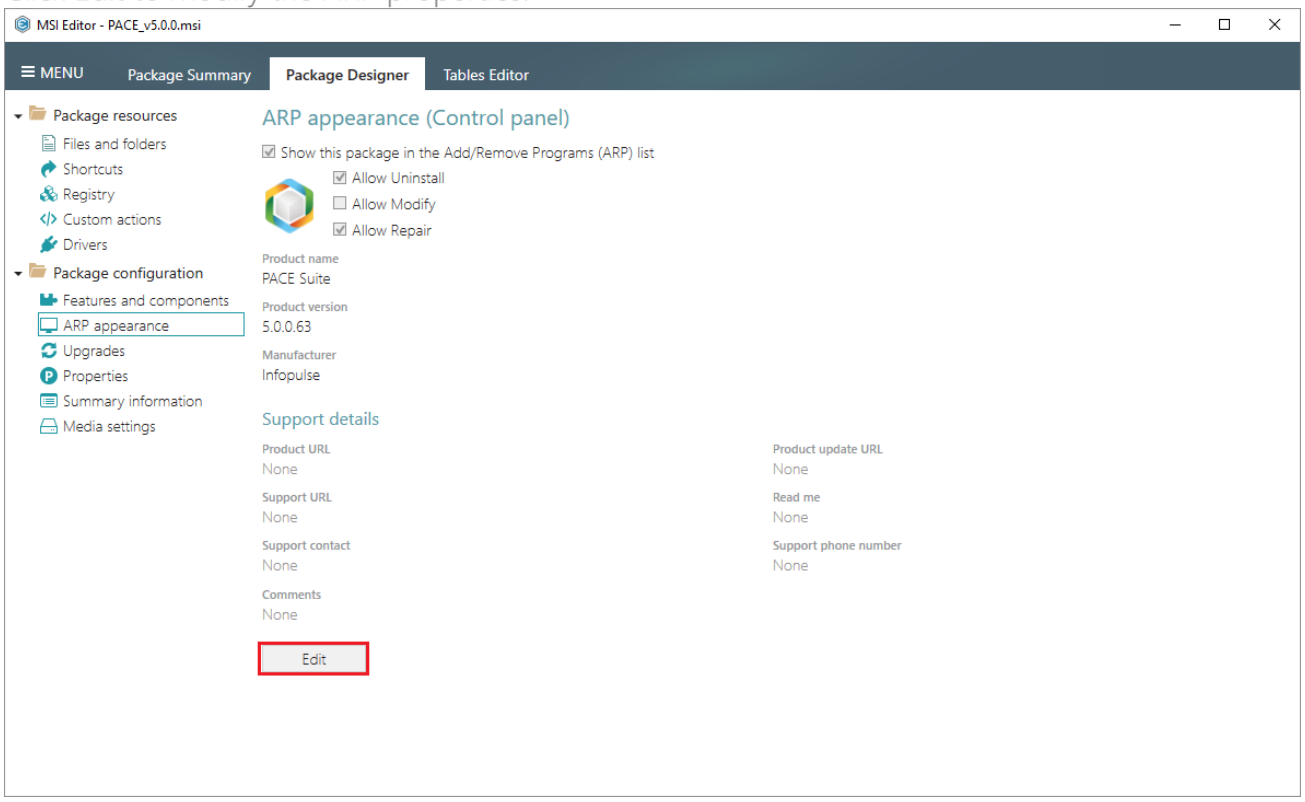
3.7.10 ARP Appearance

Control how the package will be displayed in the App list (Add & Remove Programs).

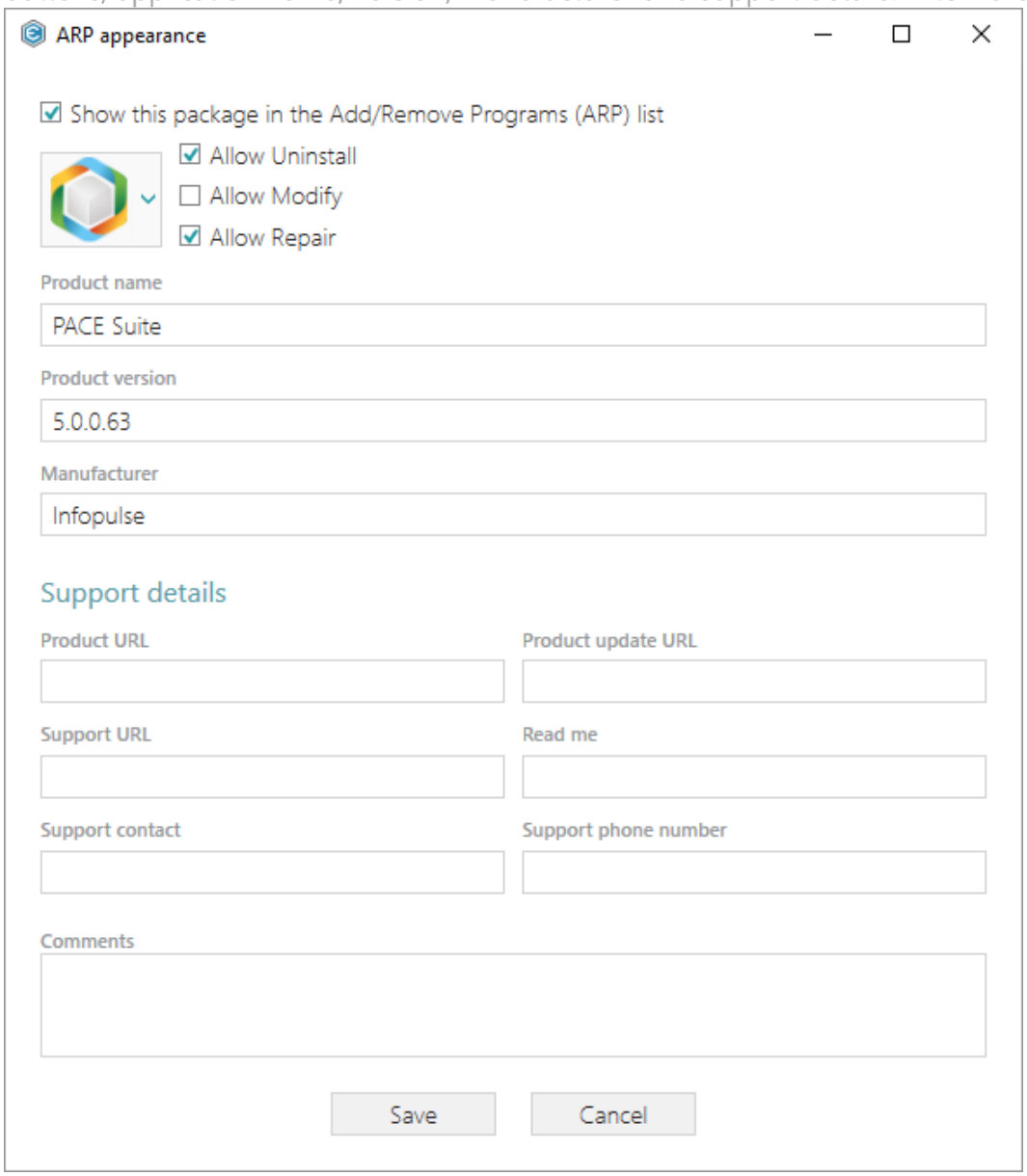
- Go to the Package Designer ->ARP appearance tab.



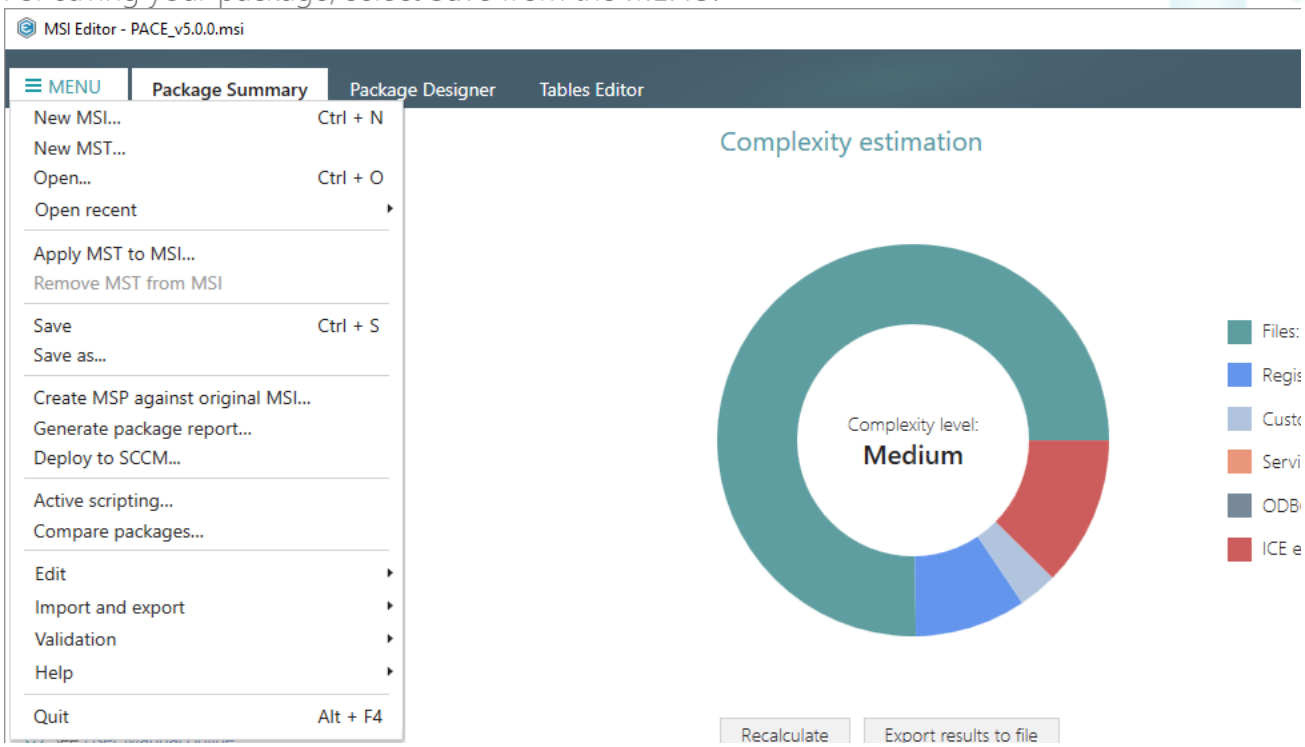
- Click Edit to modify the ARP properties.



- Update the necessary values such as package icon, availability of the Uninstall, Modify, Repair buttons, application name, version, manufacturer and support details. Afterwards, click Save.



- For saving your package, select Save from the MENU.



3.7.11 Upgrades

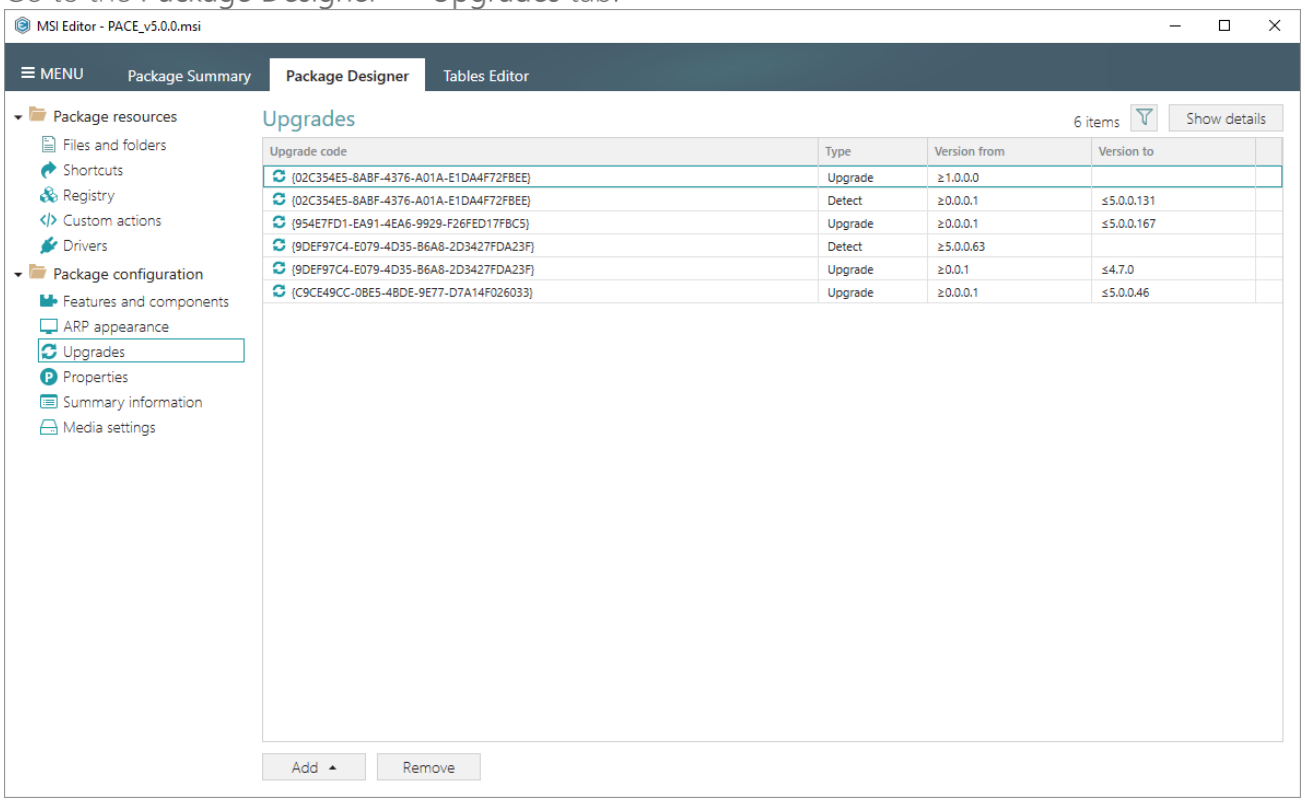
Choose a scenario that better suits your needs:

- Import Upgrades, described in section 3.7.11.1
  Implement the upgrade of previous MSI packages just selecting them from the file system.
- Add Upgrade, described in section 3.7.11.2
  Add the new upgrade manually.
- Edit Upgrade, described in section 3.7.11.3
  Edit the upgrade settings.

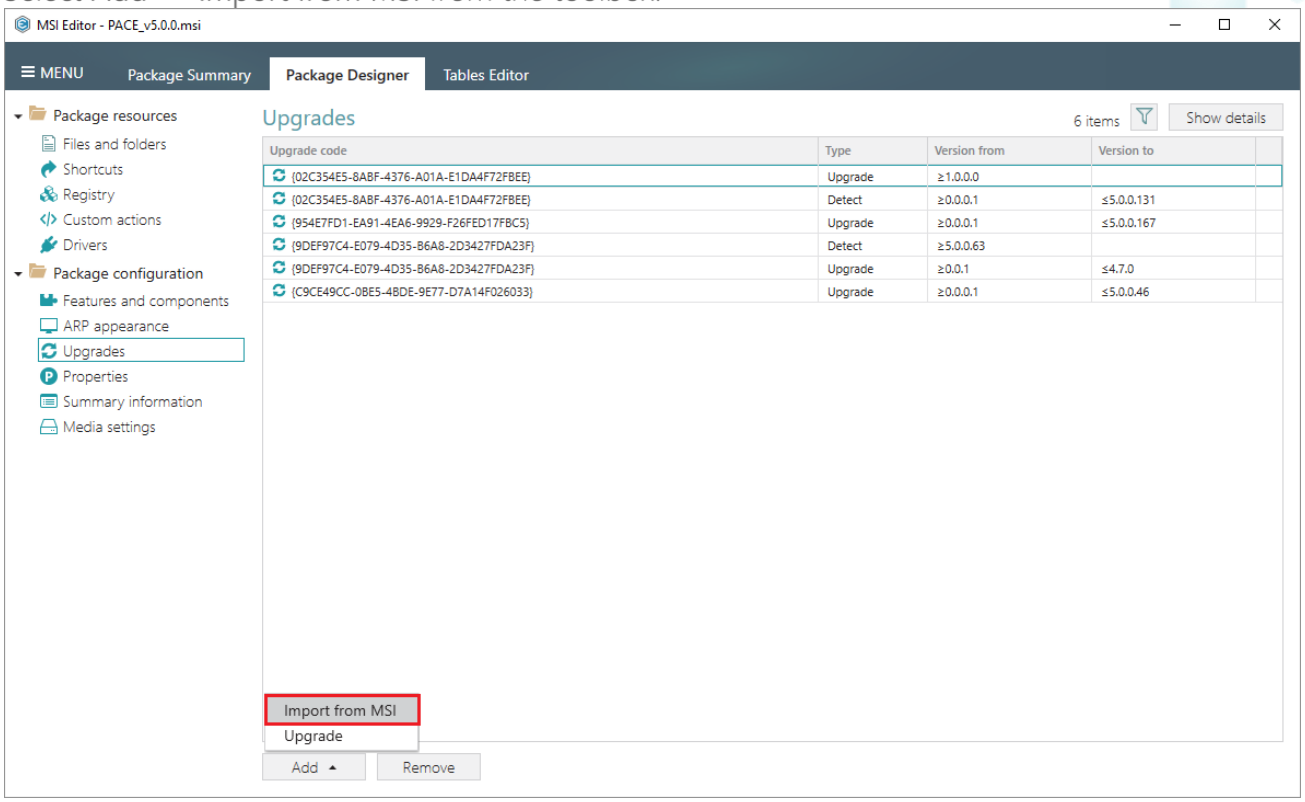
3.7.11.1 Import Upgrades

Implement the upgrade of previous MSI packages just selecting them from the file system.

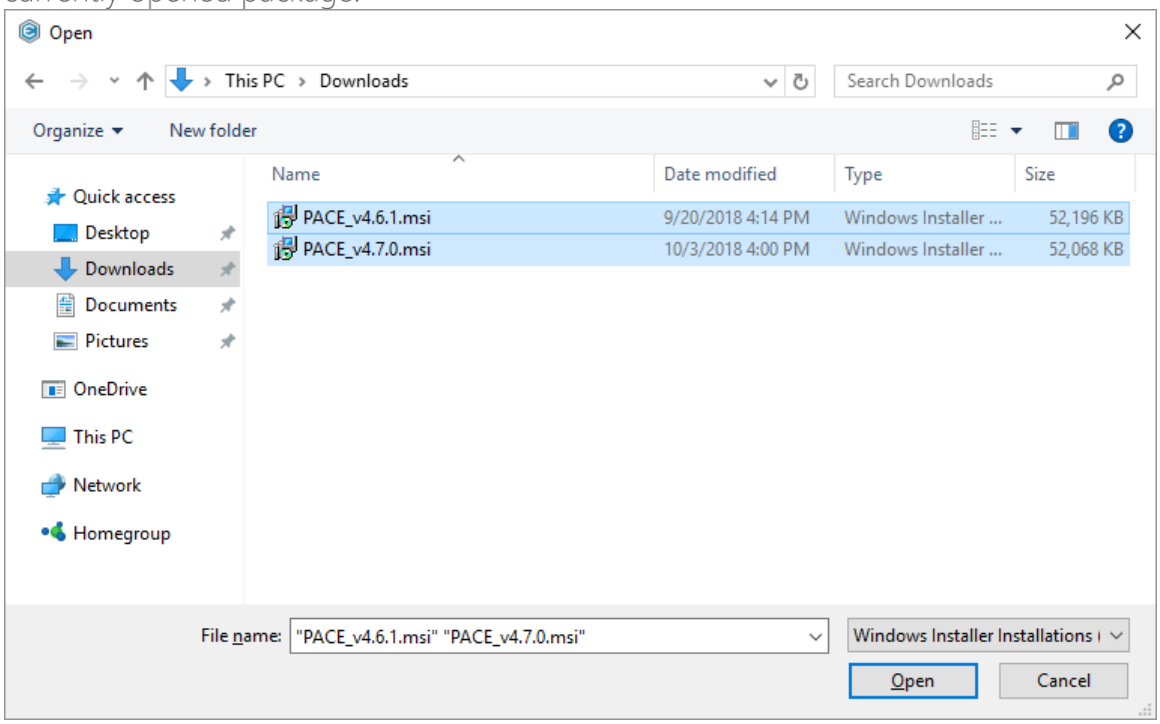
- Go to the Package Designer -> Upgrades tab.



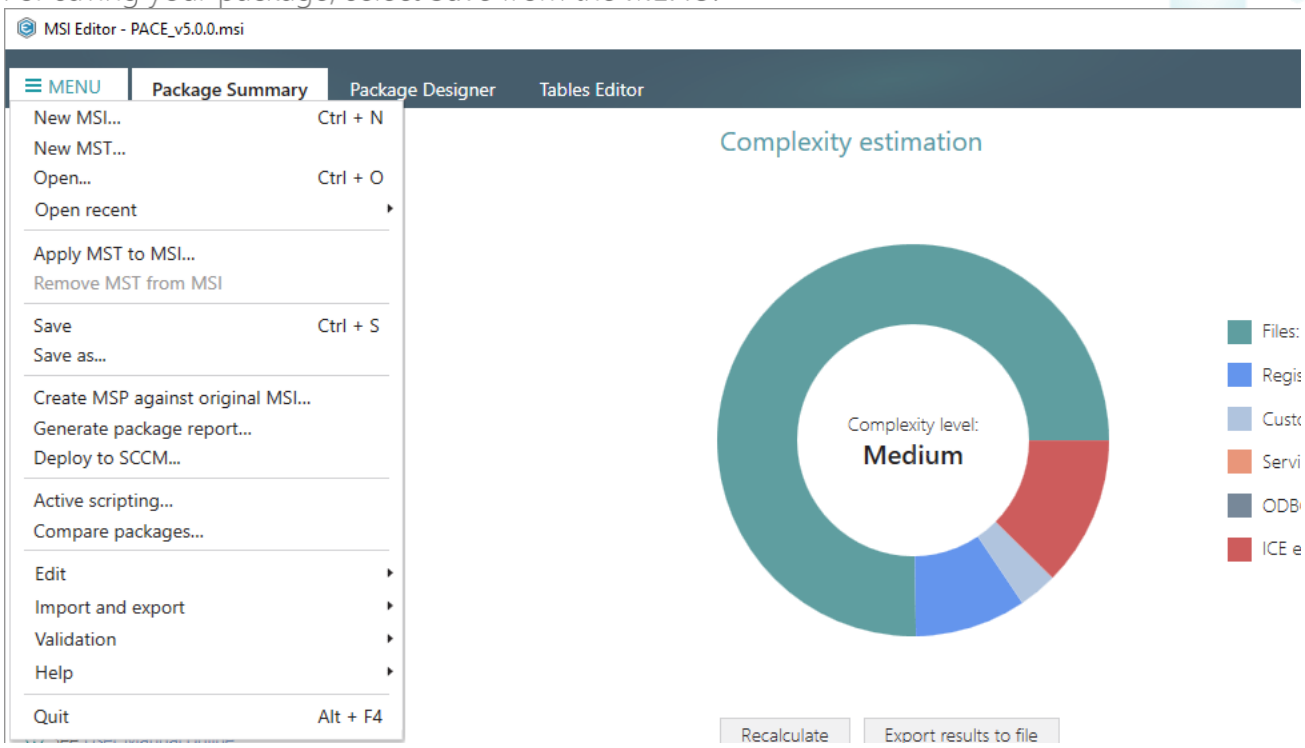
- Select Add -> Import from MSI from the toolbox.



- Select MSI packages, which you want to upgrade and click Open. The upgraded packages will be automatically removed from the target system, if exist, during the installation of the currently opened package.



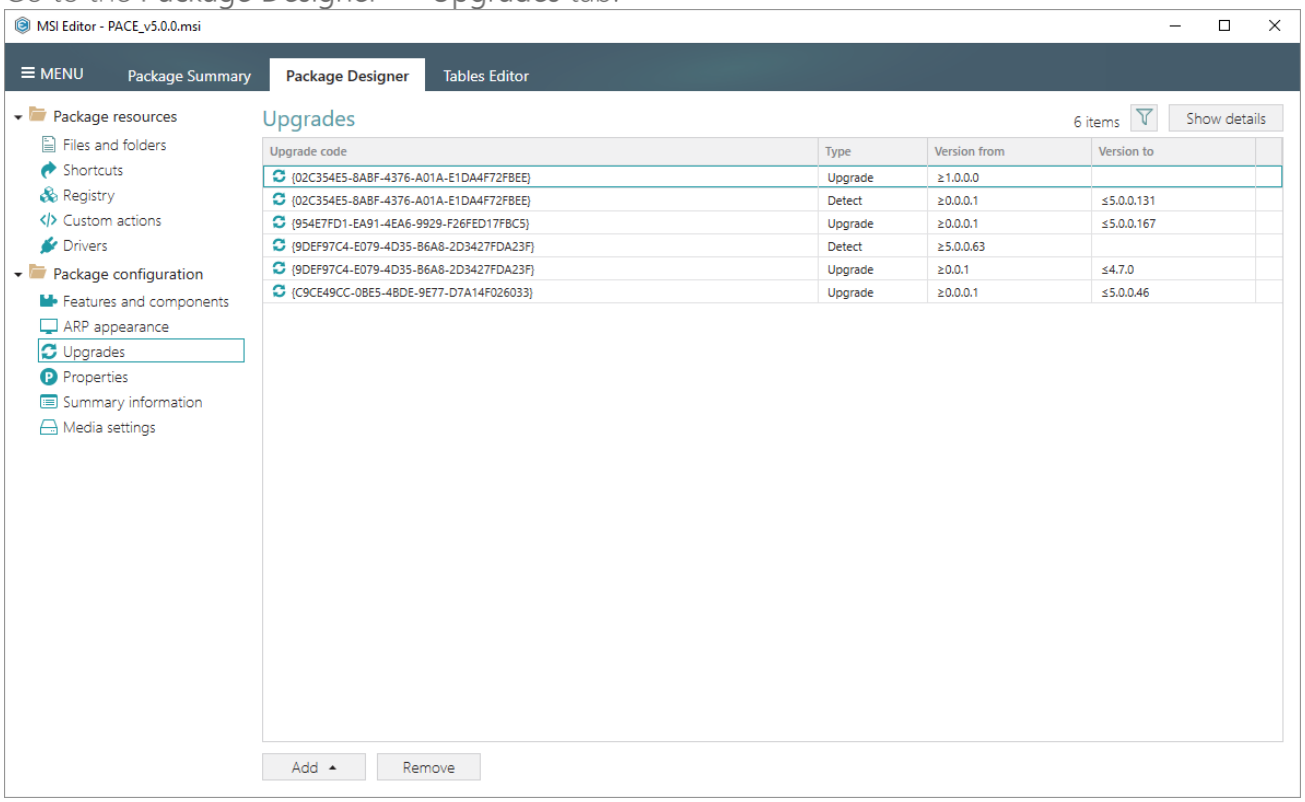
- For saving your package, select Save from the MENU.



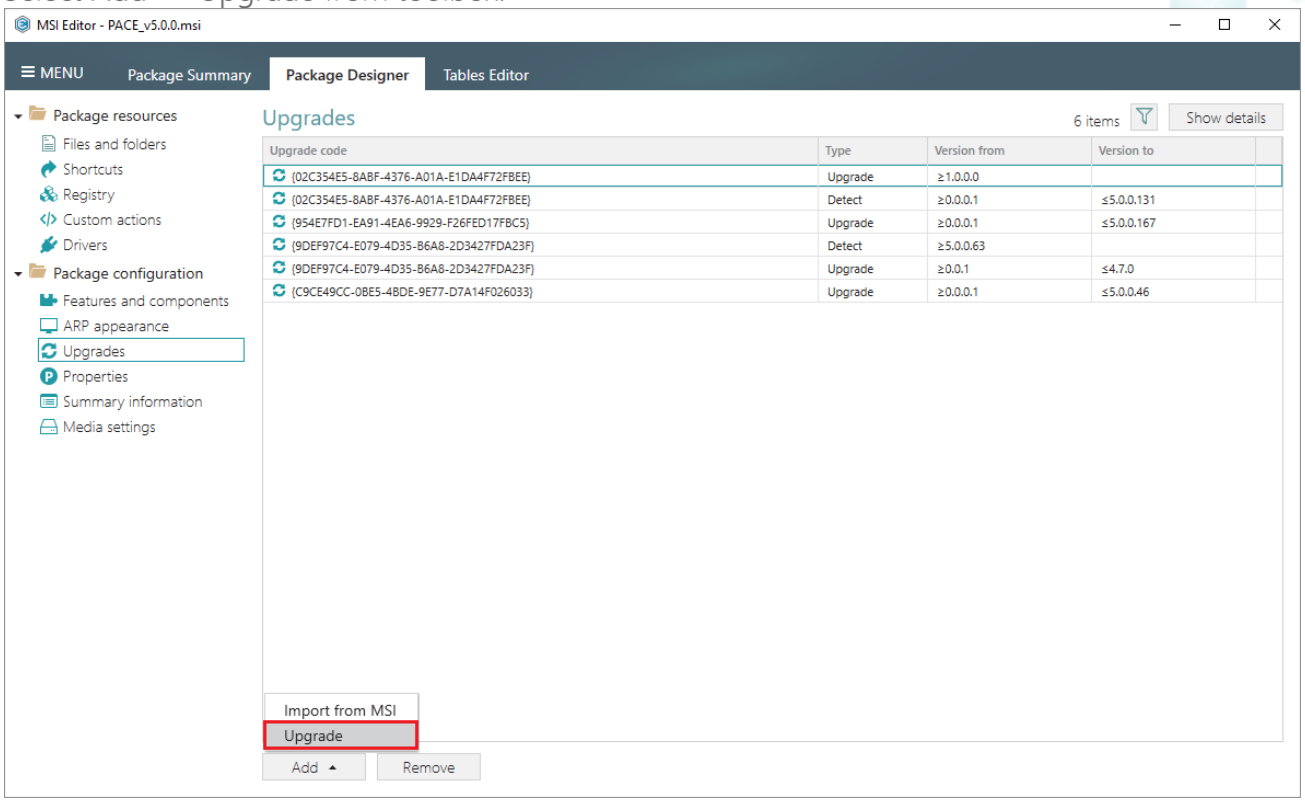
3.7.11.2 Add Upgrade

Add the new upgrade manually.

- Go to the Package Designer -> Upgrades tab.

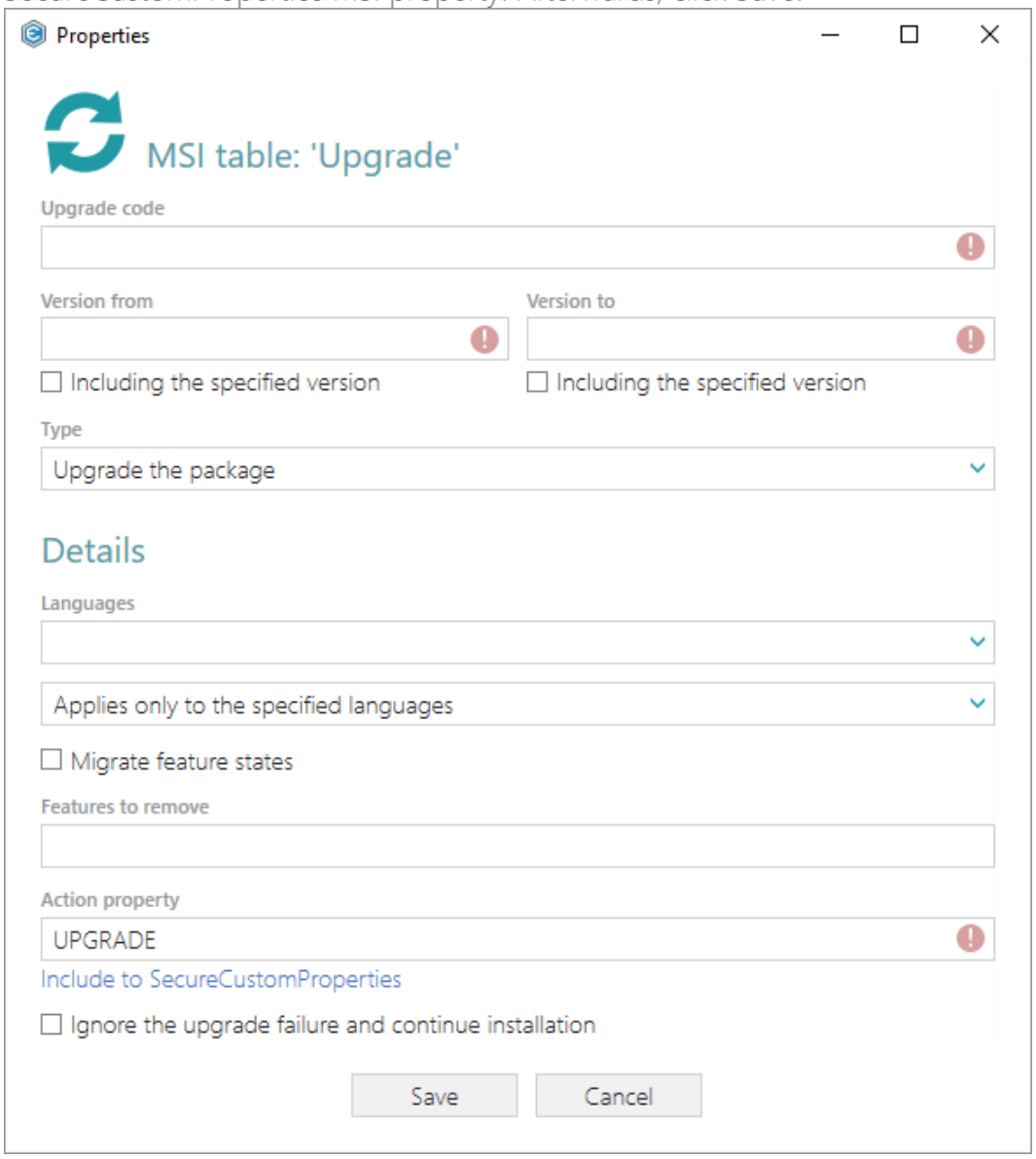


- Select Add -> Upgrade from toolbox.

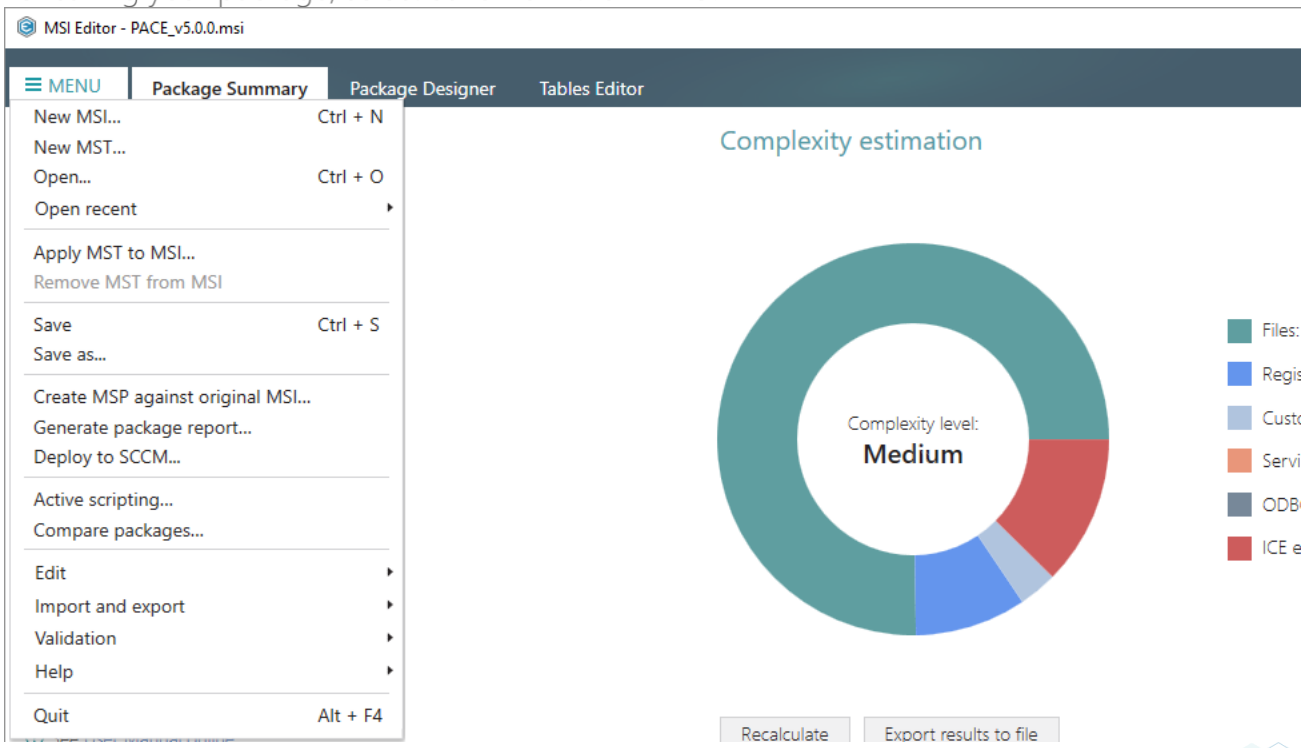


- Specify an upgrade code of the previous package(s), range of product versions, which should be upgraded, type of upgrade (detect only or upgrade), and click the Include to SecureCustomProperties link to include the specified action property to the

SecureCustomProperties MSI property. Afterwards, click Save.



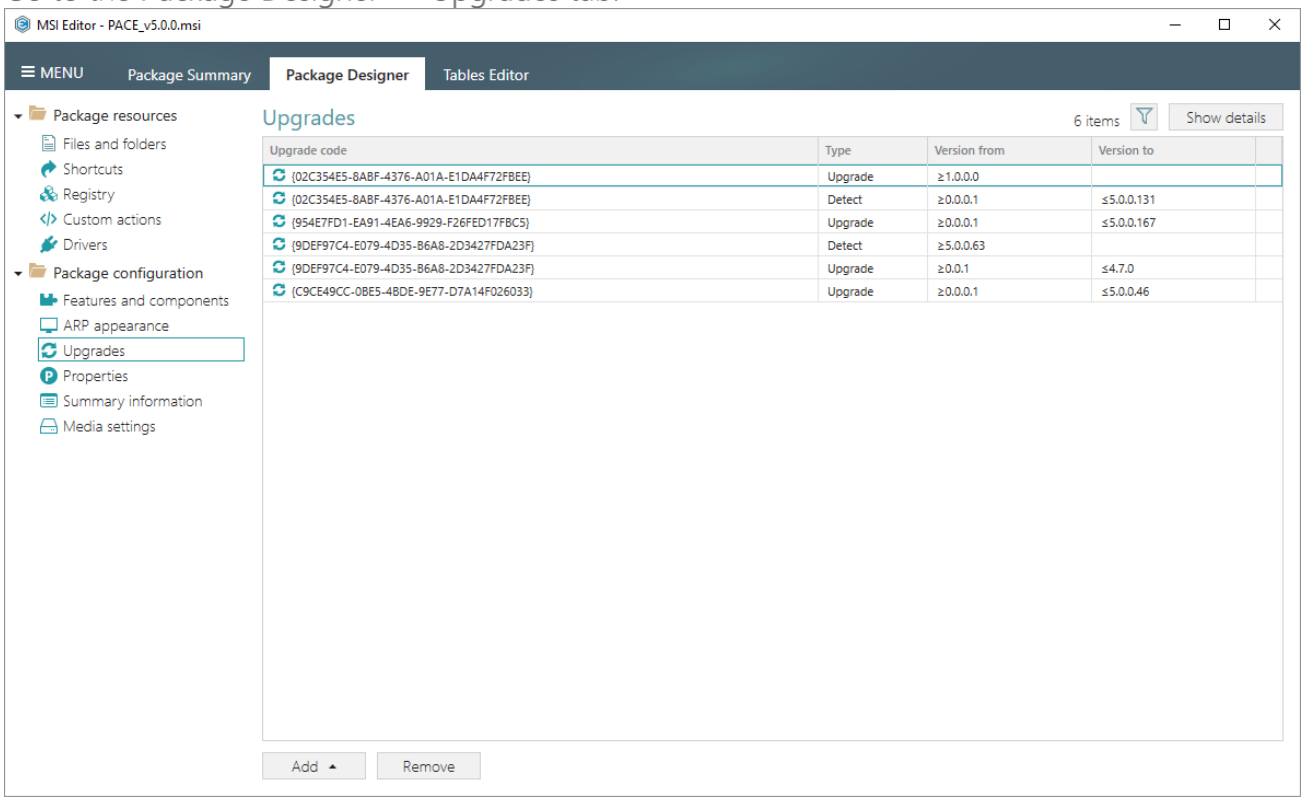
- For saving your package, select Save from the MENU.



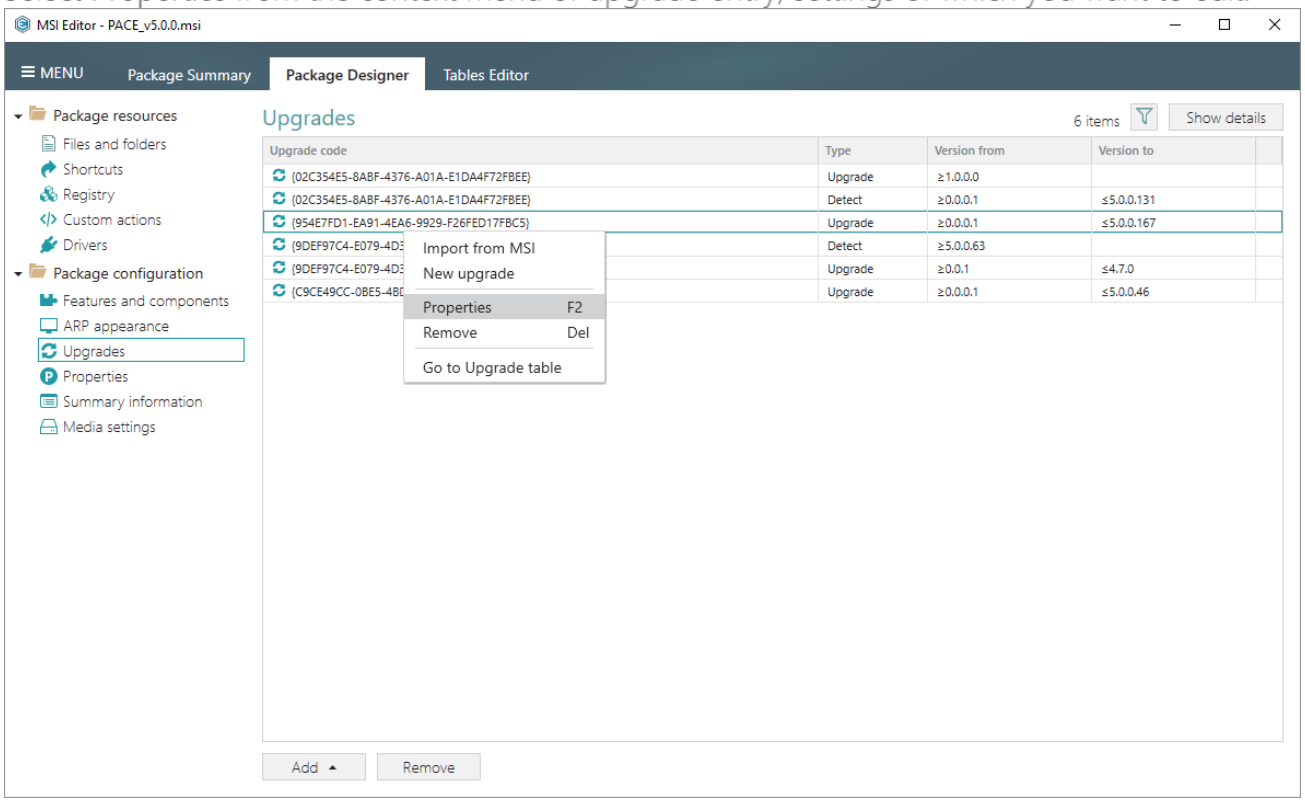
3.7.11.3 Edit Upgrade

Edit the upgrade settings.

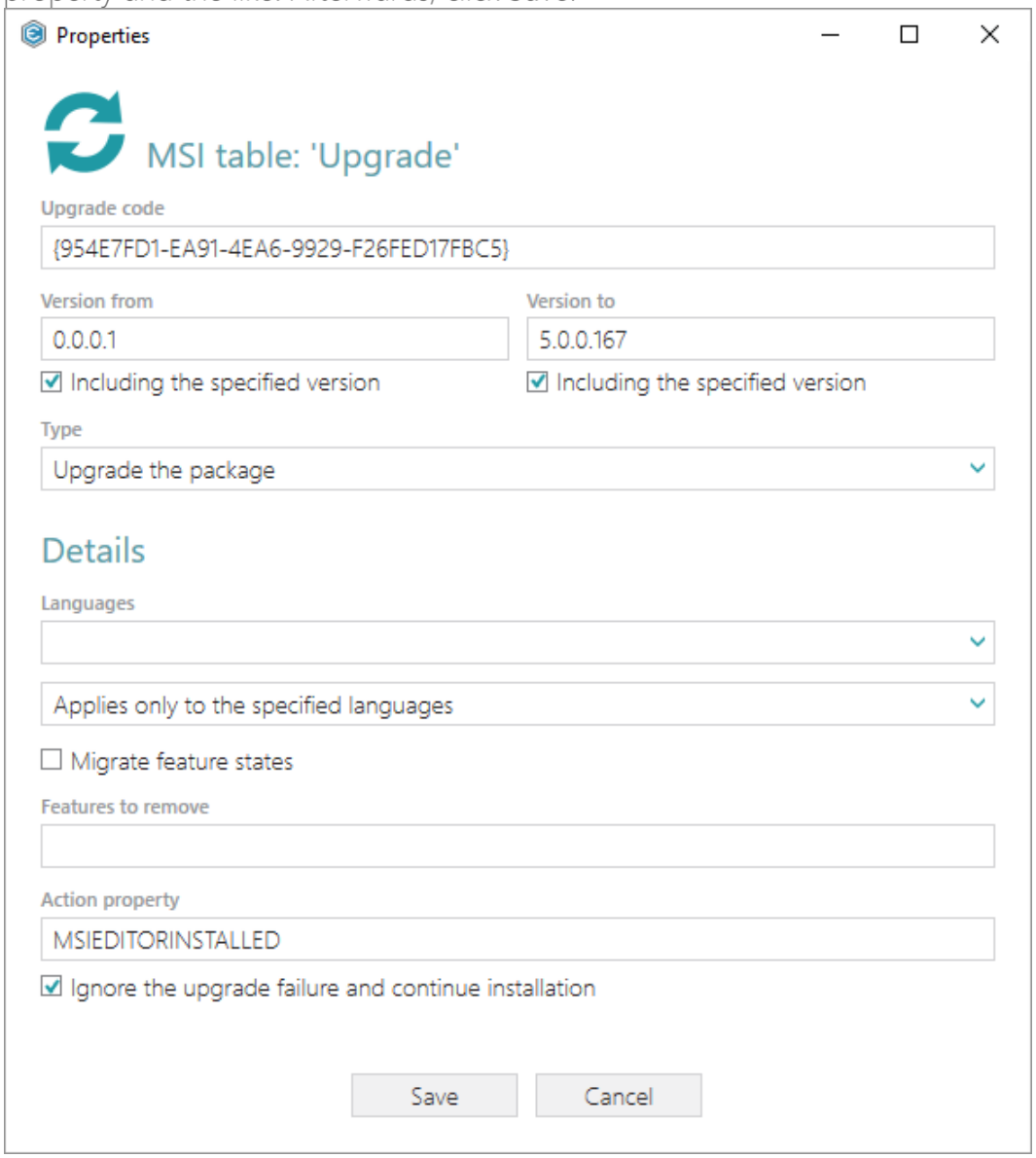
- Go to the Package Designer -> Upgrades tab.



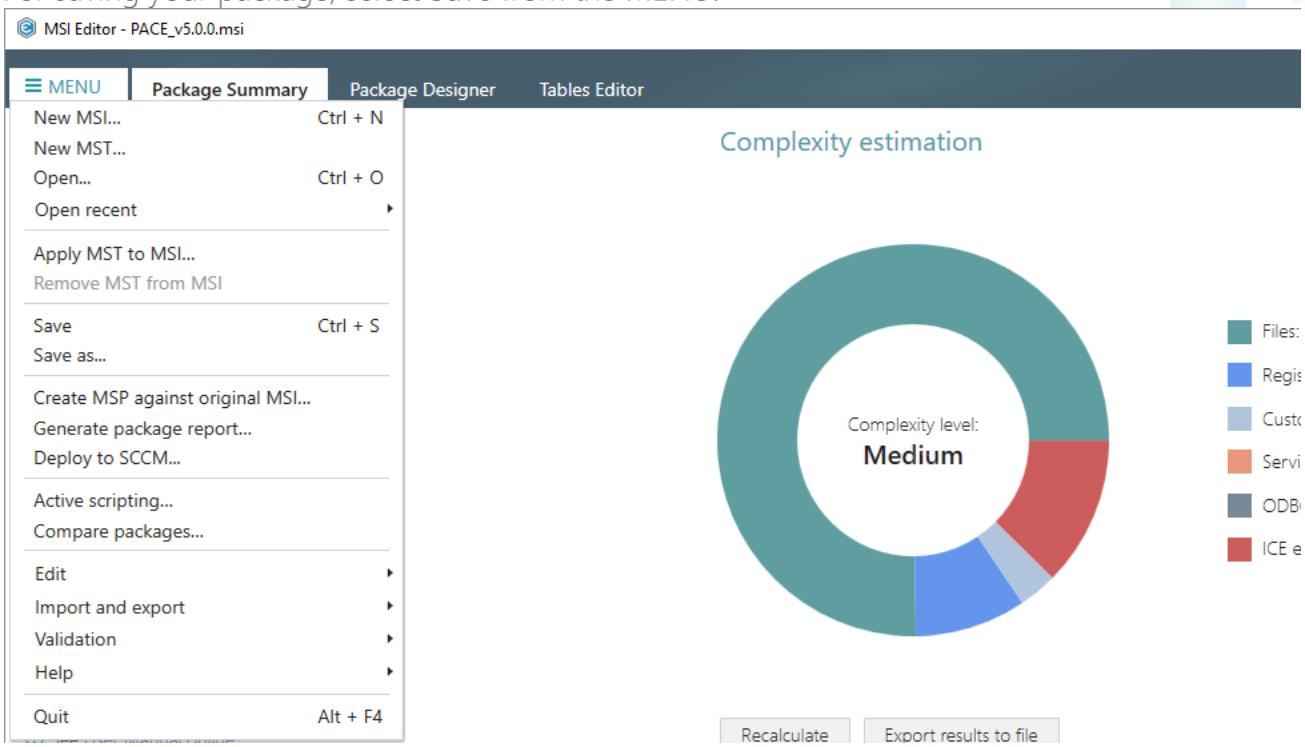
- Select Properties from the context menu of upgrade entry, settings of which you want to edit.



- Update the necessary values such as upgrade code, range of versions, upgrade type, action property and the like. Afterwards, click Save.



- For saving your package, select Save from the MENU.



3.7.12 Properties

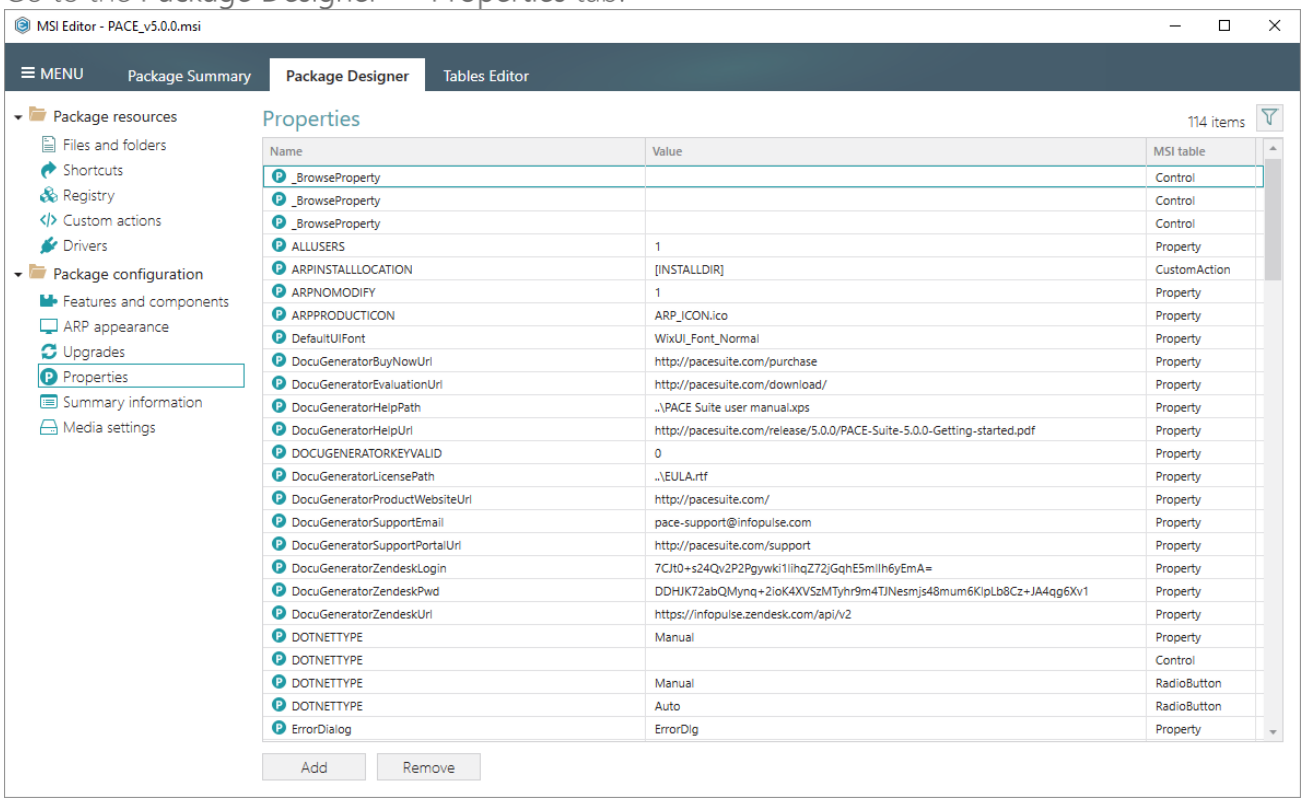
Choose a scenario that better suits your needs:

- Add Property, described in section 3.7.12.1
  Add a new property to the package.
- Edit Property, described in section 3.7.12.2
  Edit a name and a value of the existing property.

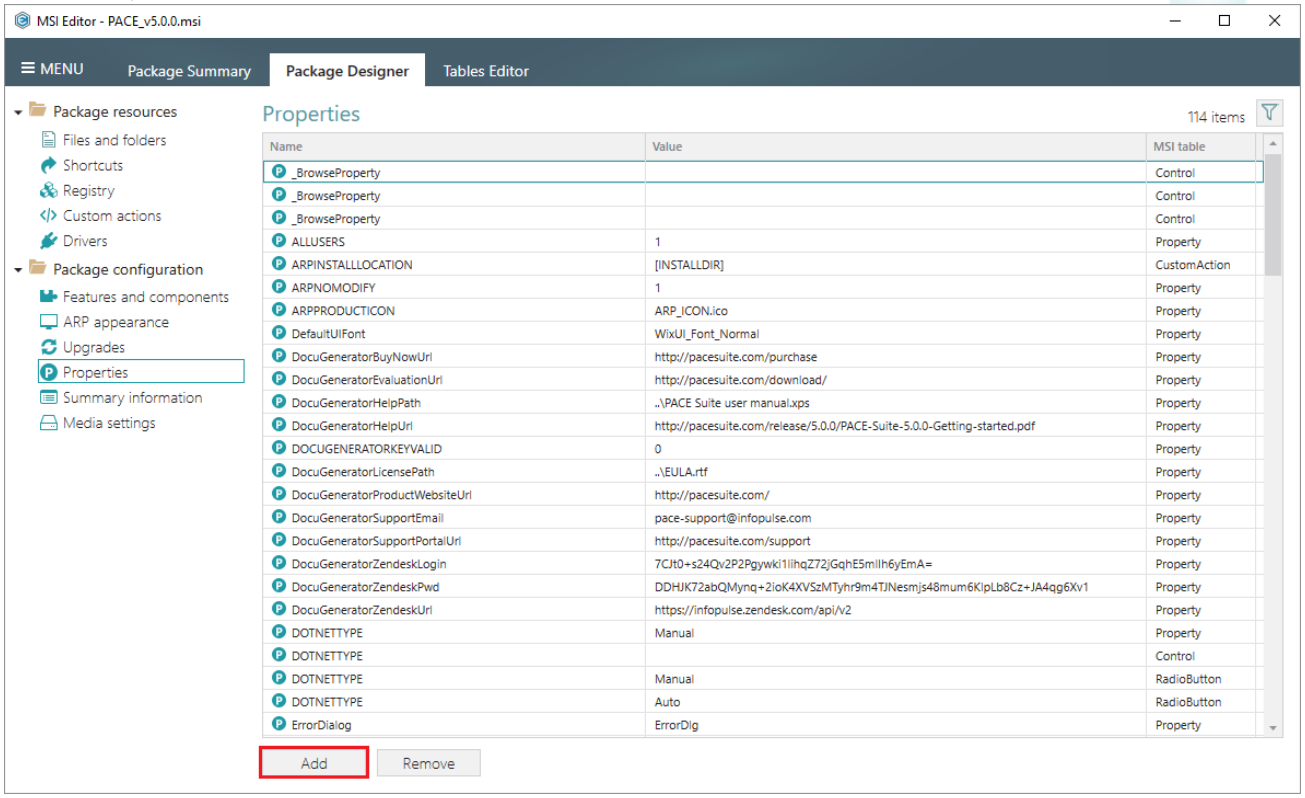
3.7.12.1 Add Property

Add a new property to the package.

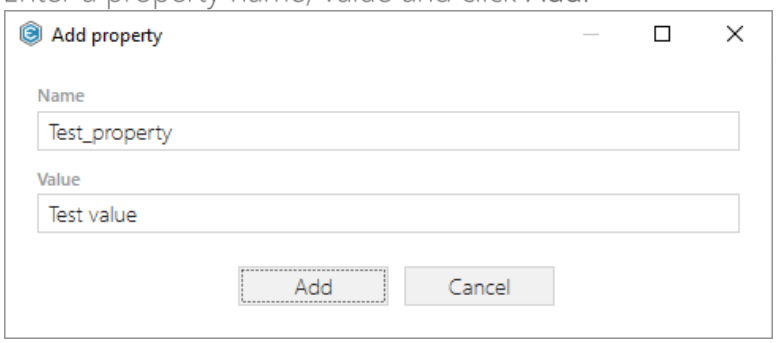
- Go to the Package Designer -> Properties tab.



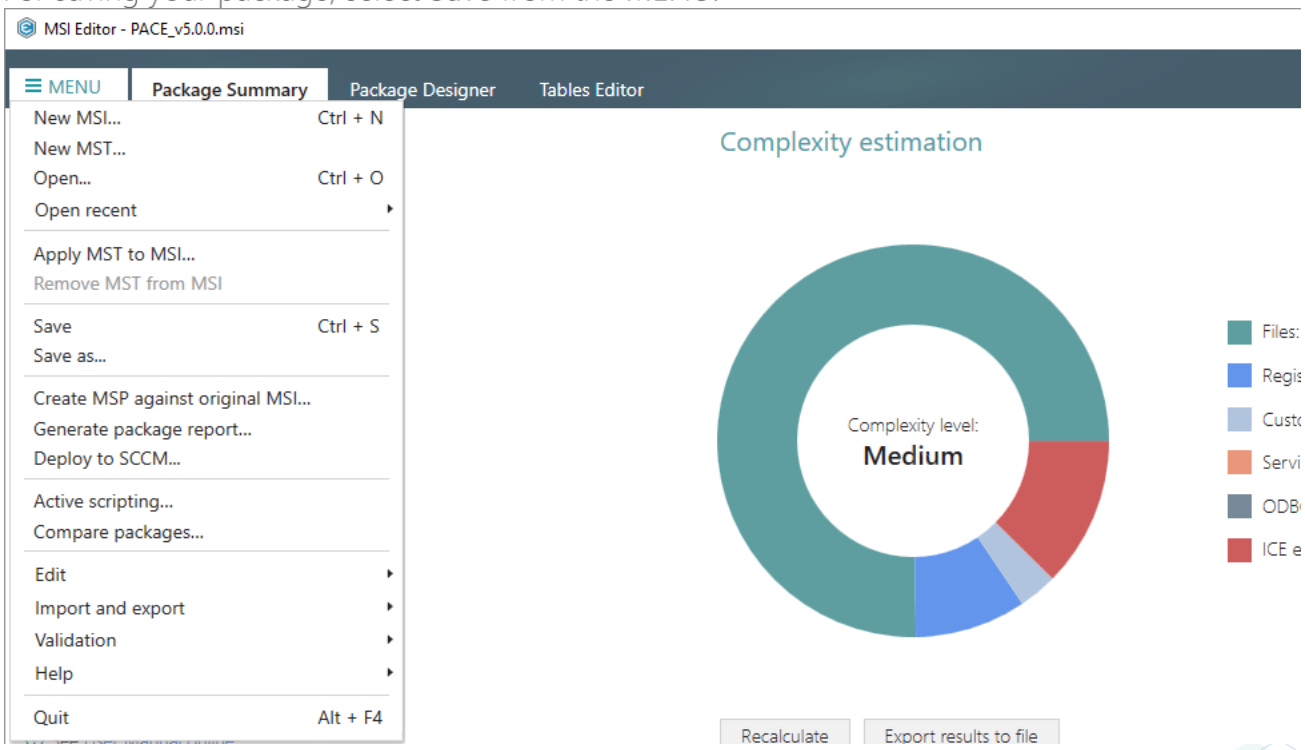
- Click Add, located in the toolbox.



- Enter a property name, value and click Add.



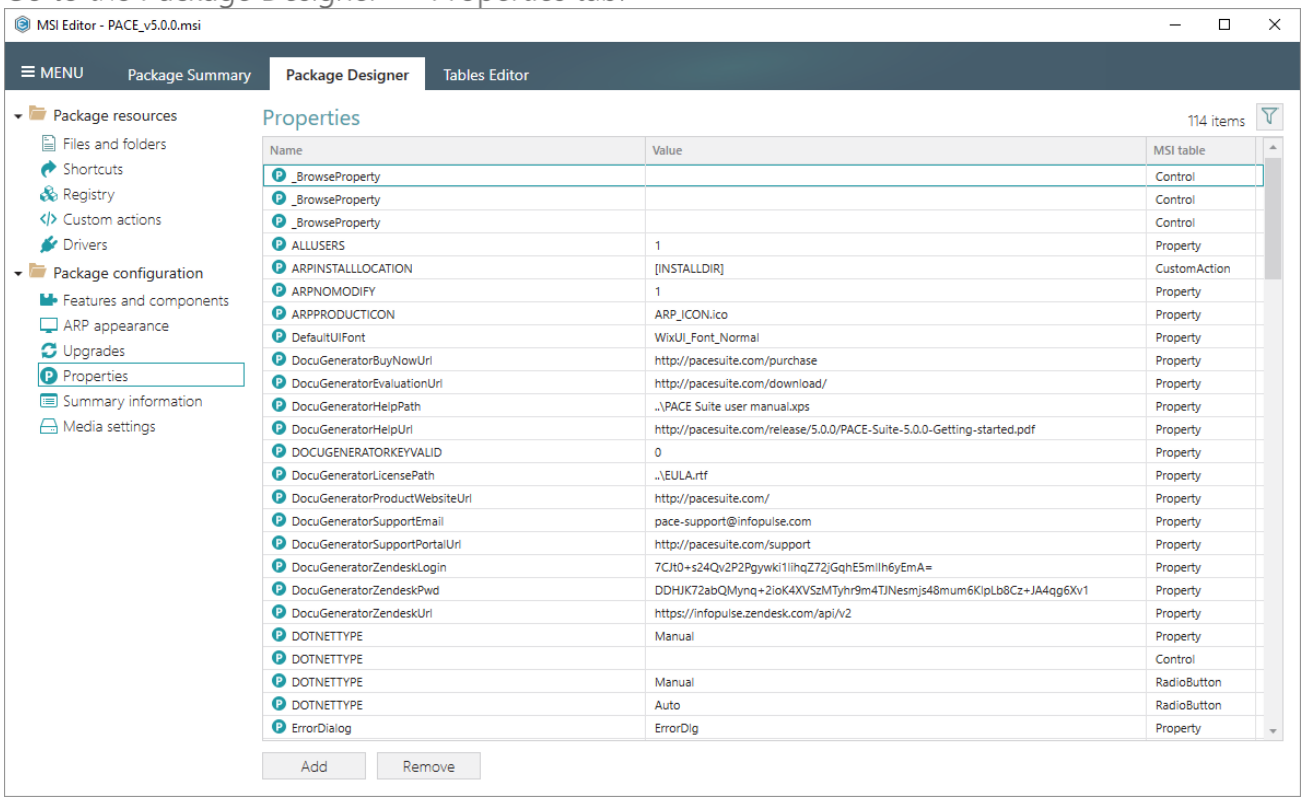
- For saving your package, select Save from the MENU.



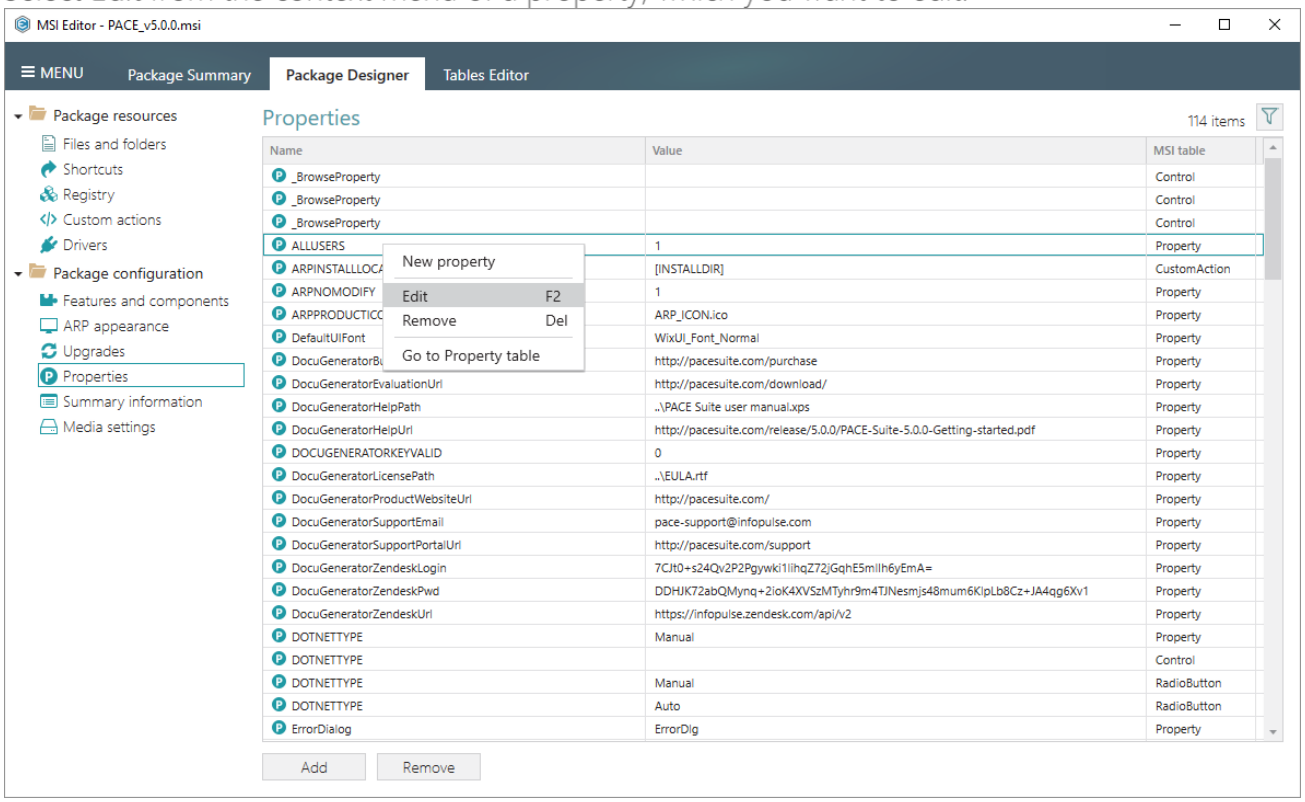
3.7.12.2 Edit Property

Edit a name and value of the existing property.

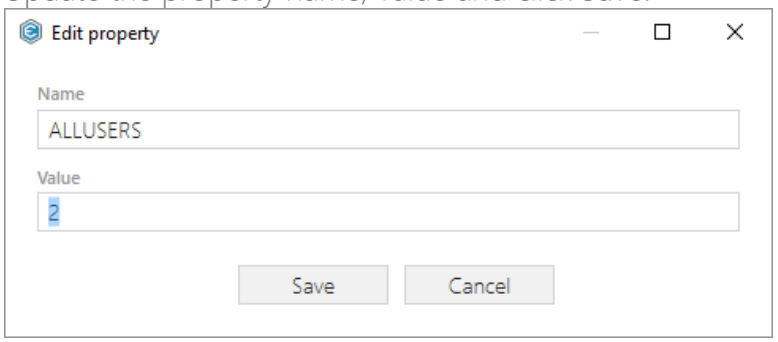
- Go to the Package Designer -> Properties tab.



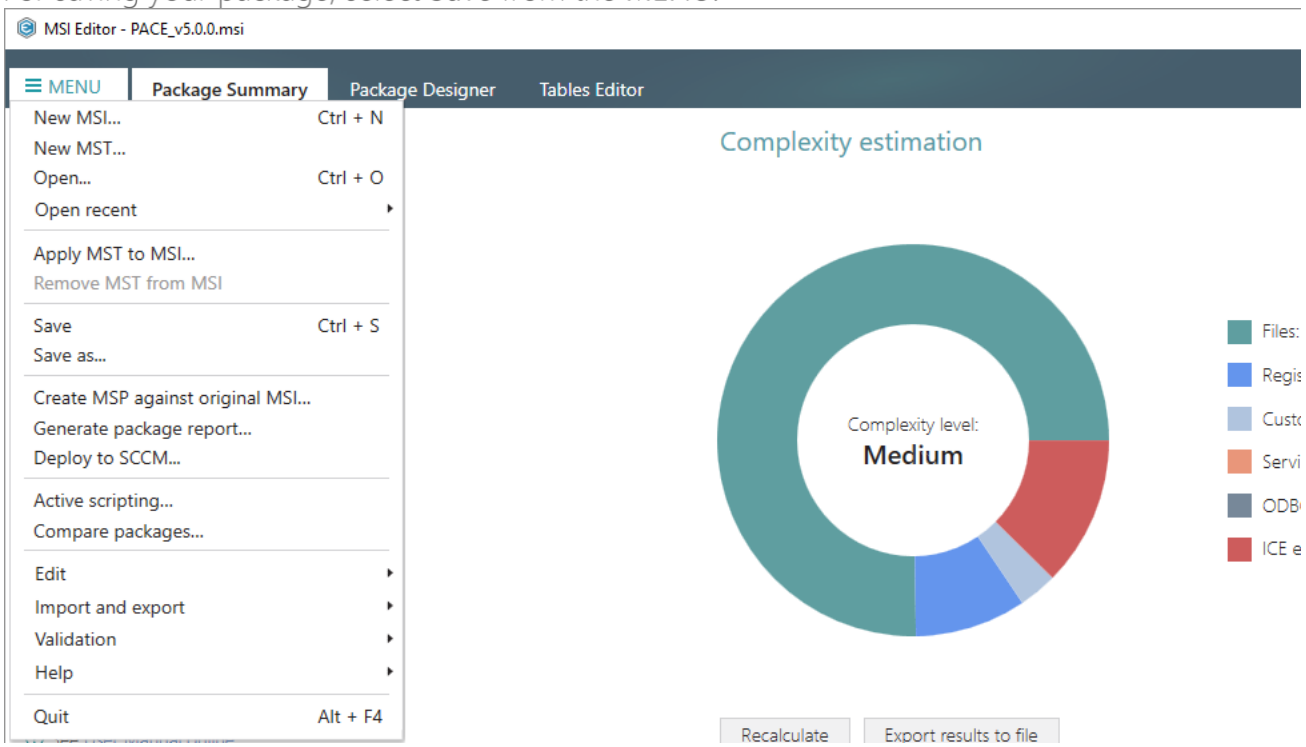
- Select Edit from the context menu of a property, which you want to edit.



- Update the property name, value and click Save.



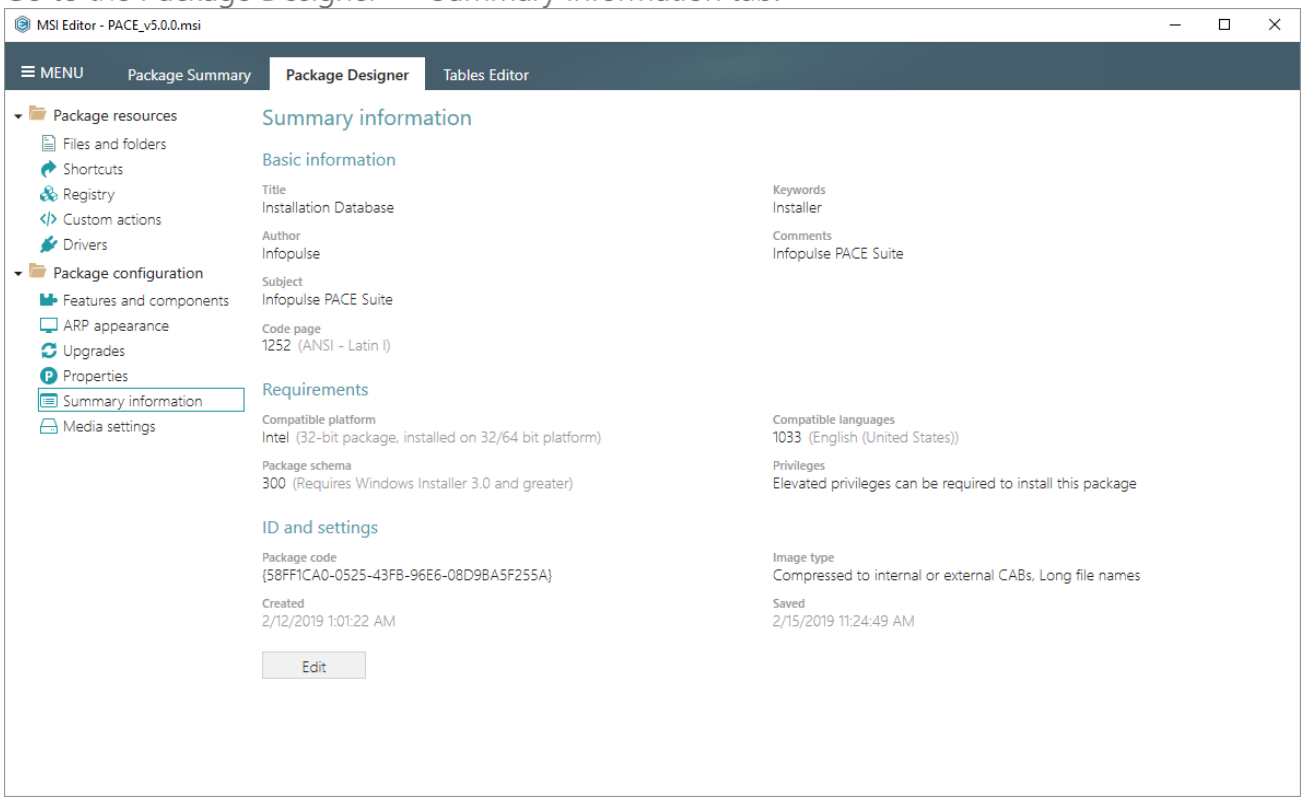
- For saving your package, select Save from the MENU.



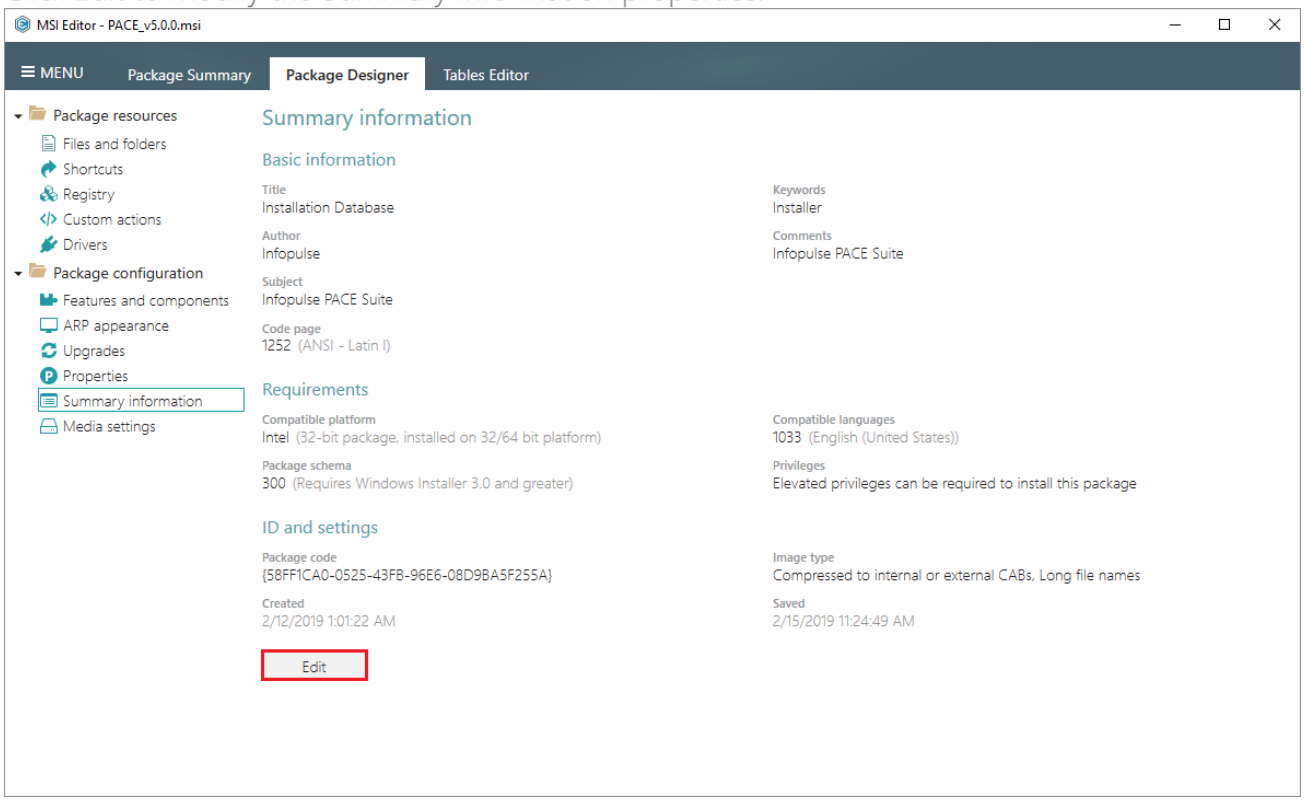
3.7.13 Summary Information

Edit the Summary Information properties of the opened MSI package or applied MST file.

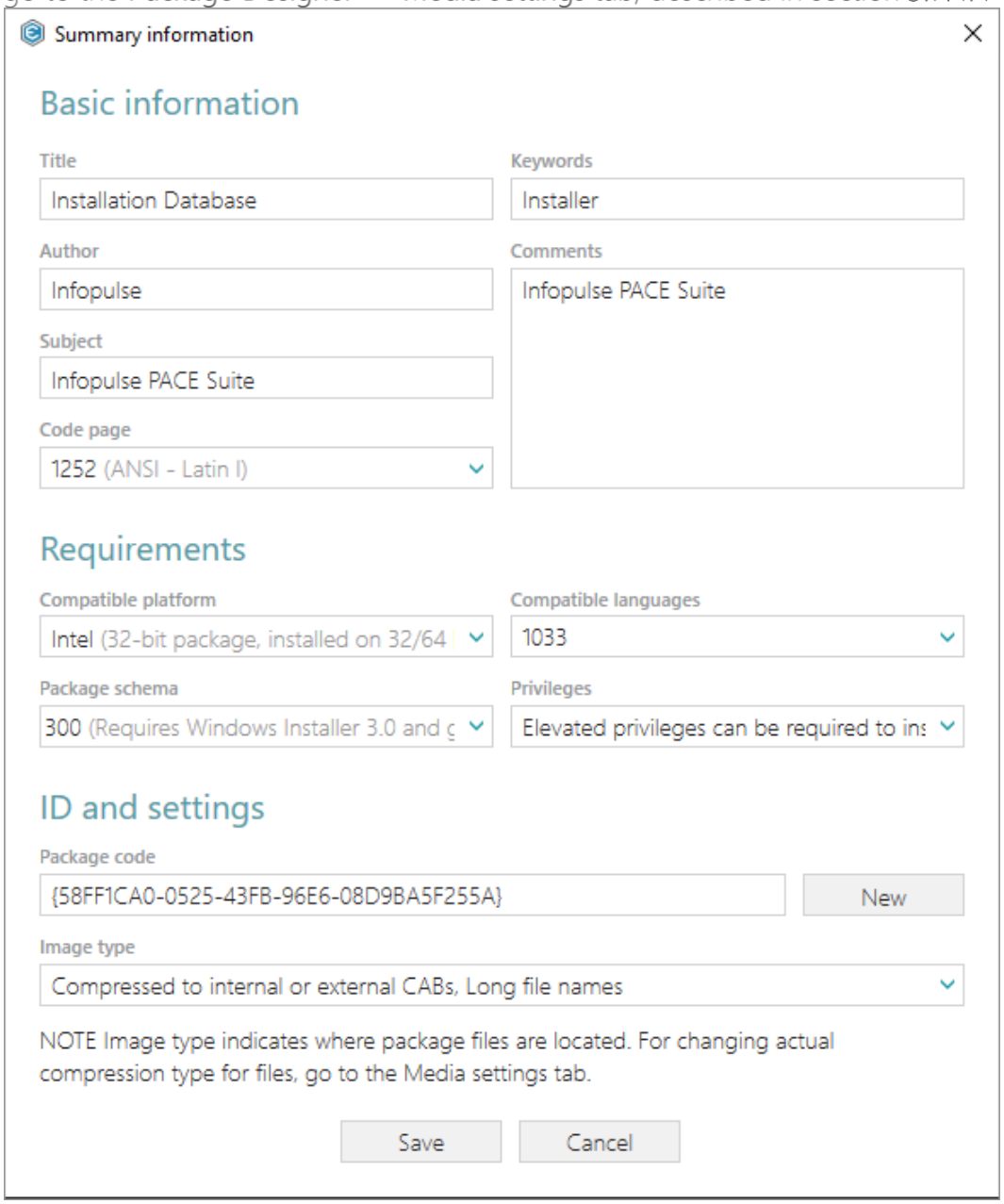
- Go to the Package Designer -> Summary Information tab.



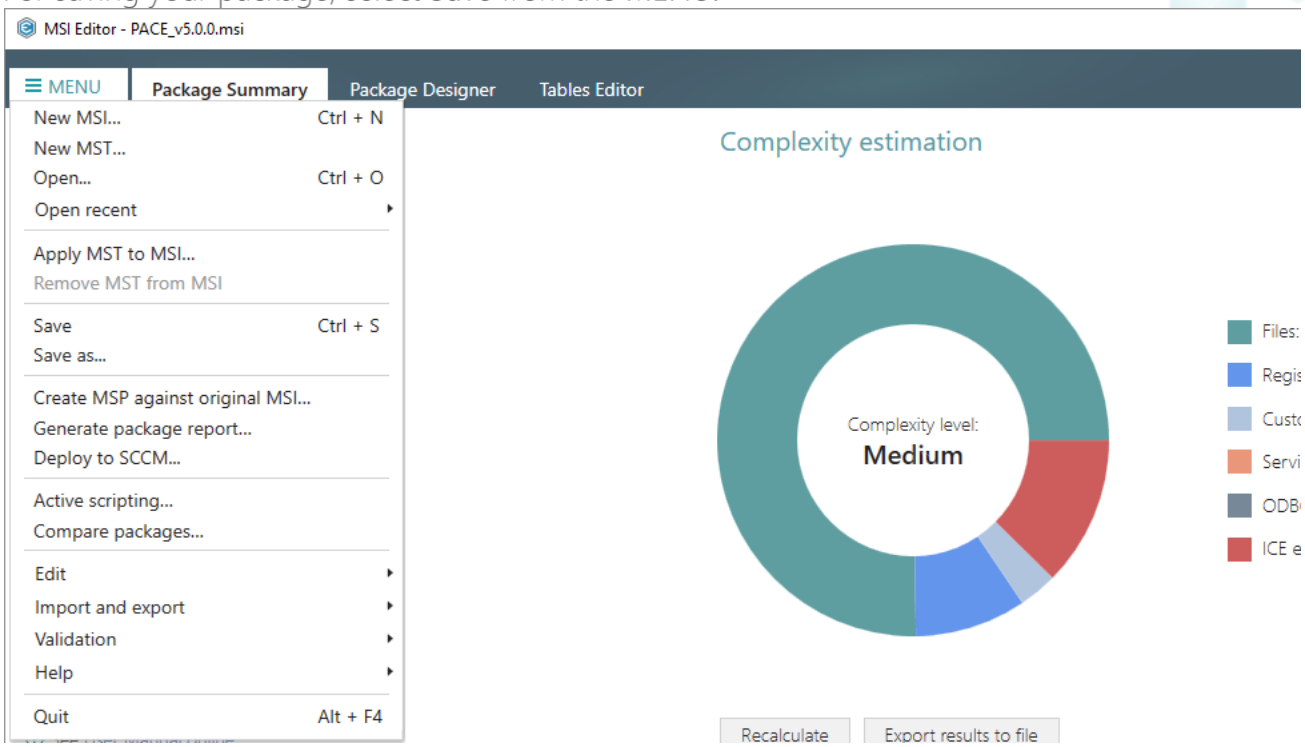
- Click Edit to modify the Summary Information properties.



- Update the necessary values such as title, author, compatible platform, package schema and the like. Afterwards, click Save.
  NOTE If you need to change a compression type for files, instead of editing the Image type, go to the Package Designer -> Media settings tab, described in section 3.7.4.4



- For saving your package, select Save from the MENU.



3.7.14 Tables

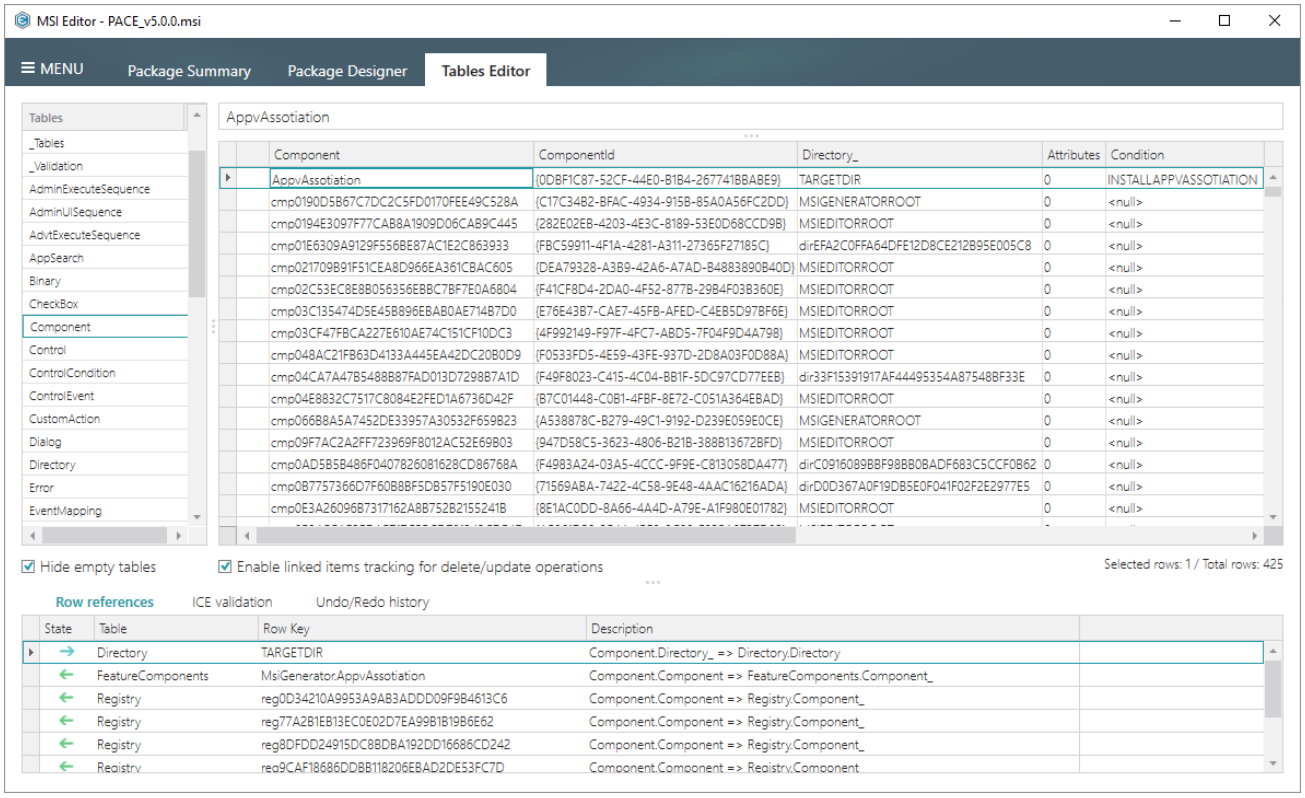
Choose a scenario that better suits your needs:

- Add Tables, described in section 3.7.14.1
  Add the standard MSI tables to the package.
- Drop Table, described in section 3.7.14.2
  Drop tables from the package controling which of linked resources also will be removed.
- Import Tables (IDT), described in section 3.7.14.3
  Import standard or custom tables with or without content from the IDT files.

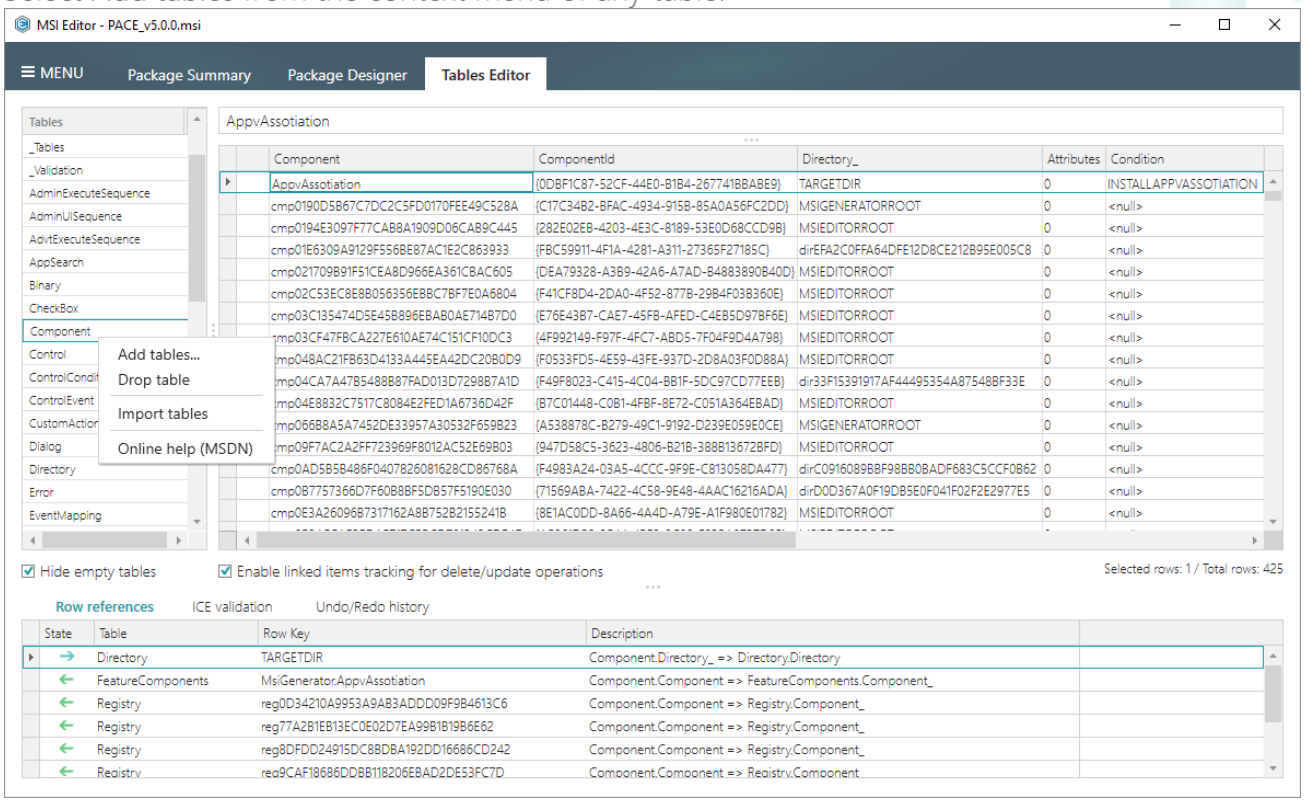
3.7.14.1 Add Tables

Add the standard MSI tables to the package.

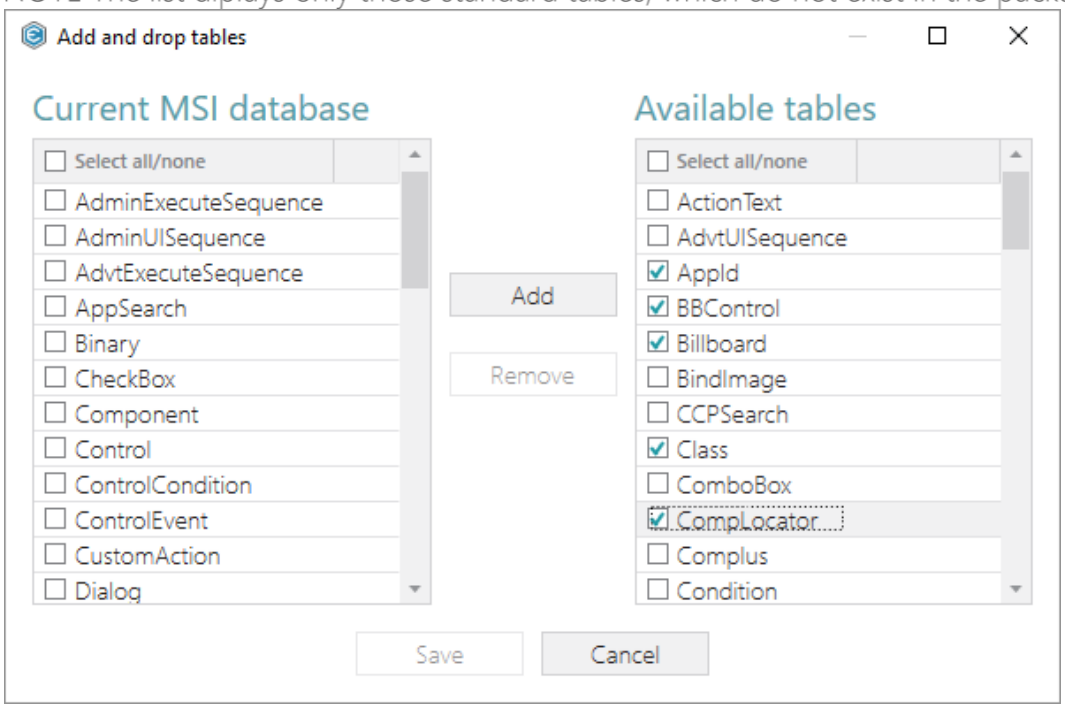
- Go to the Tables Editor tab.



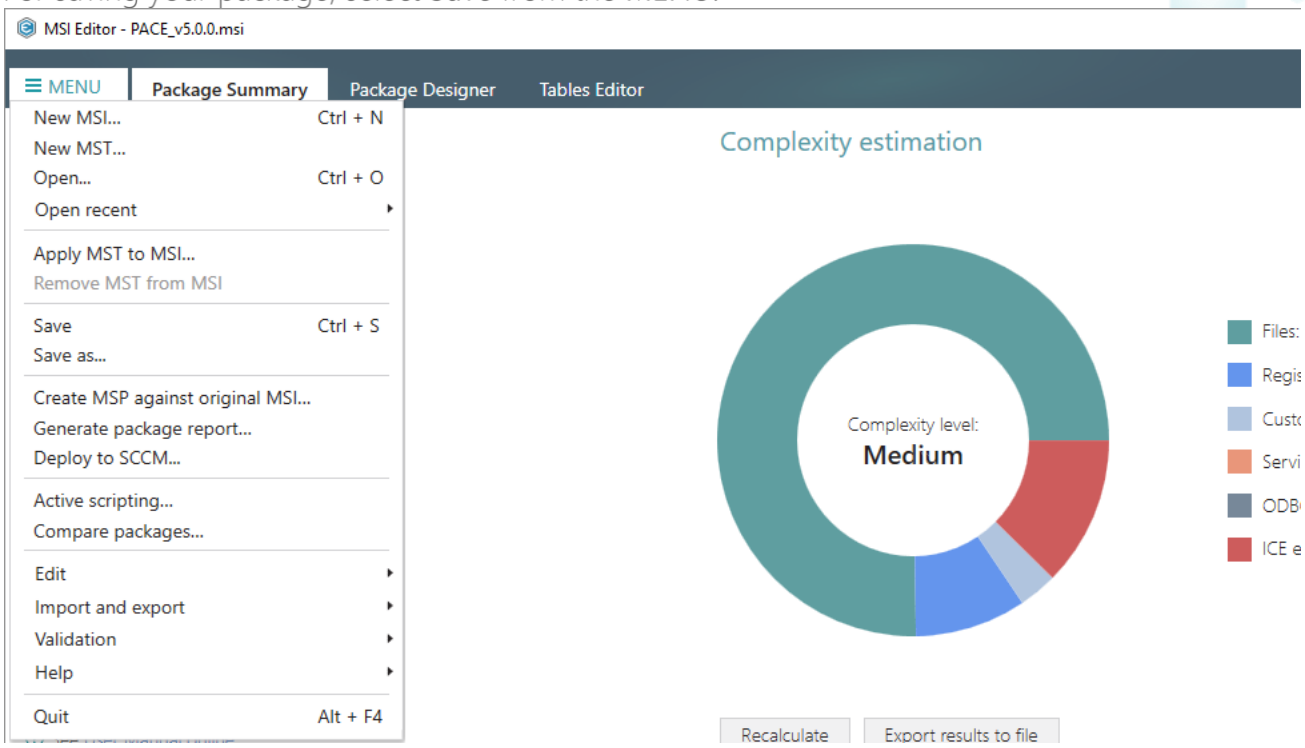
- Select **Add tables** from the context menu of any table.



- Select tables from the **Available tables** list, which you want to add to the package, click **Add** and **Save**.
NOTE The list diplays only those standard tables, which do not exist in the package.



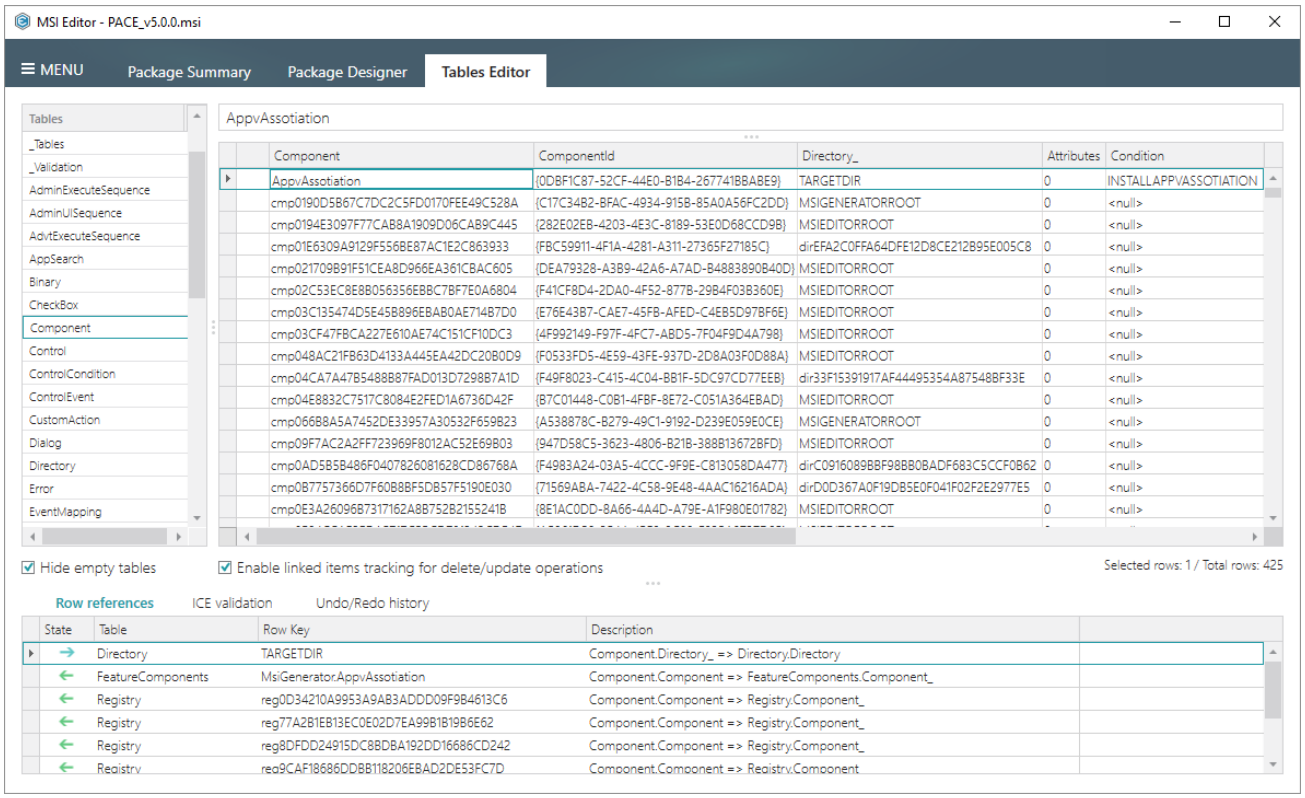
- For saving your package, select Save from the MENU.



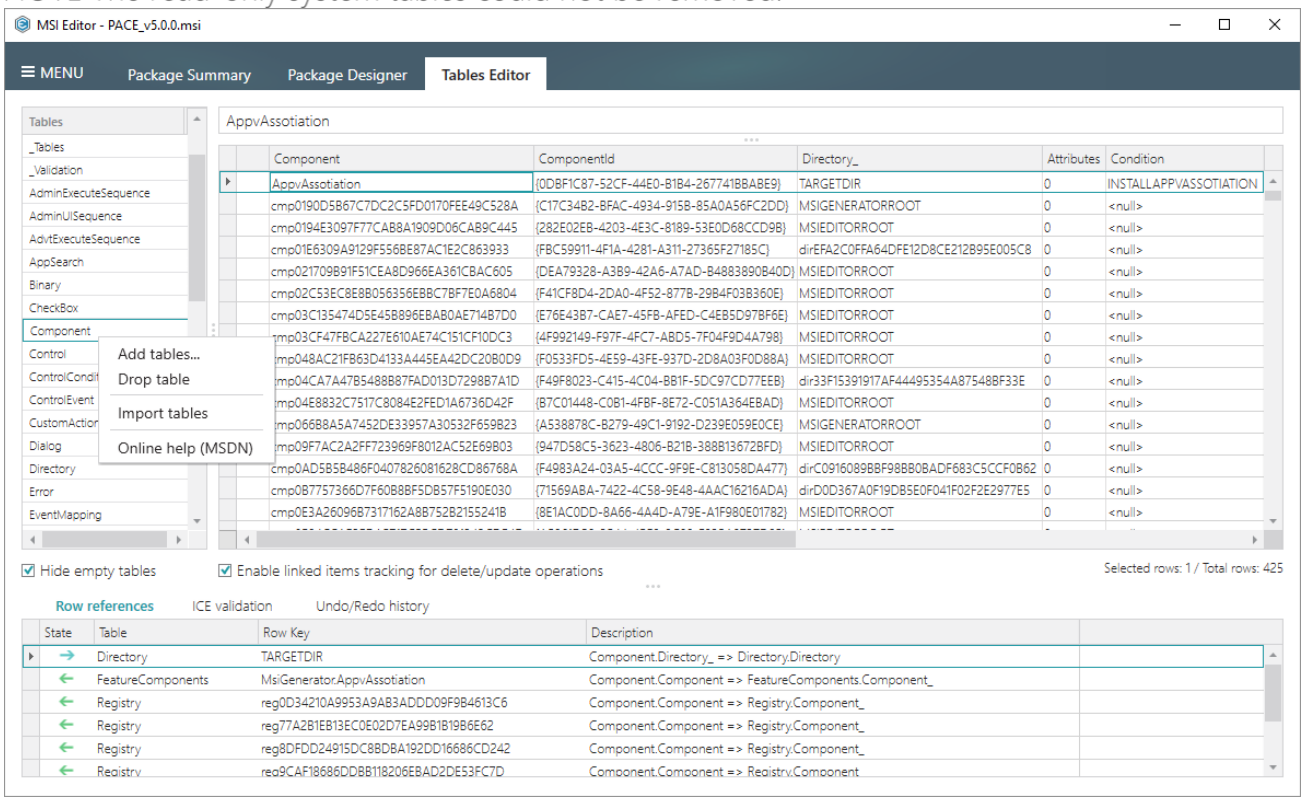
3.7.14.2 Drop Table

Drop tables from the package controling which of linked resources also will be removed.

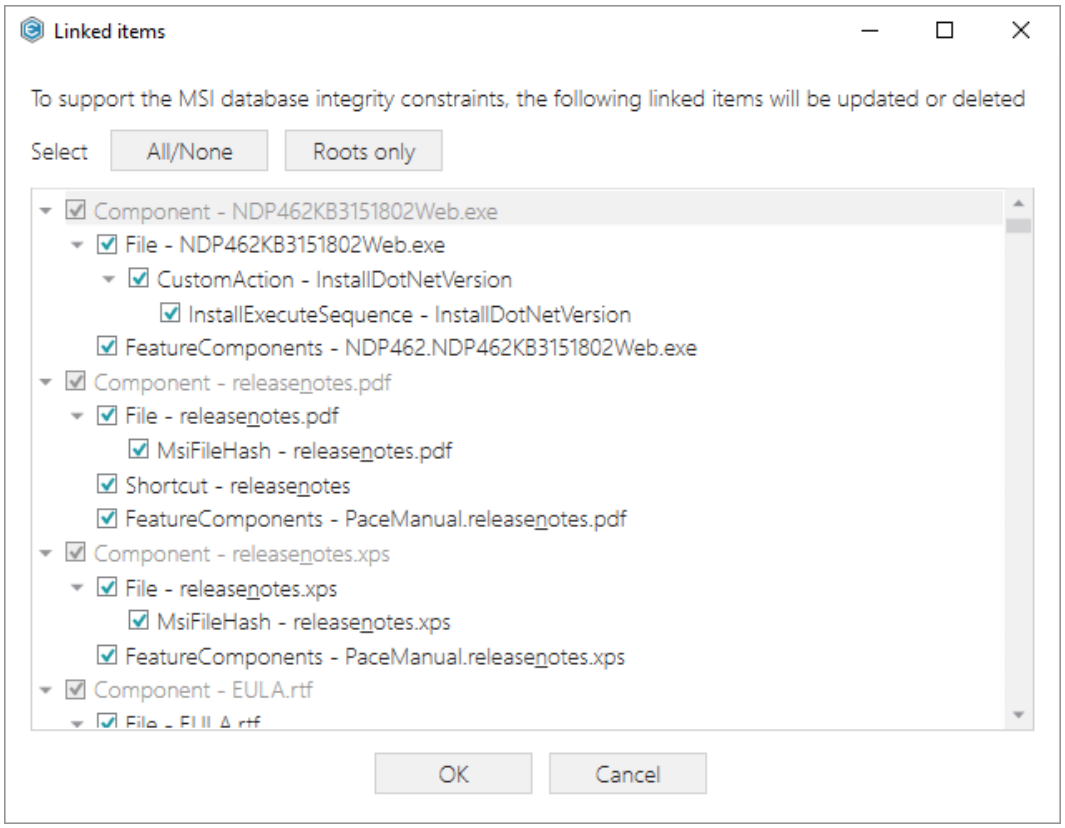
- Go to the Tables Editor tab.



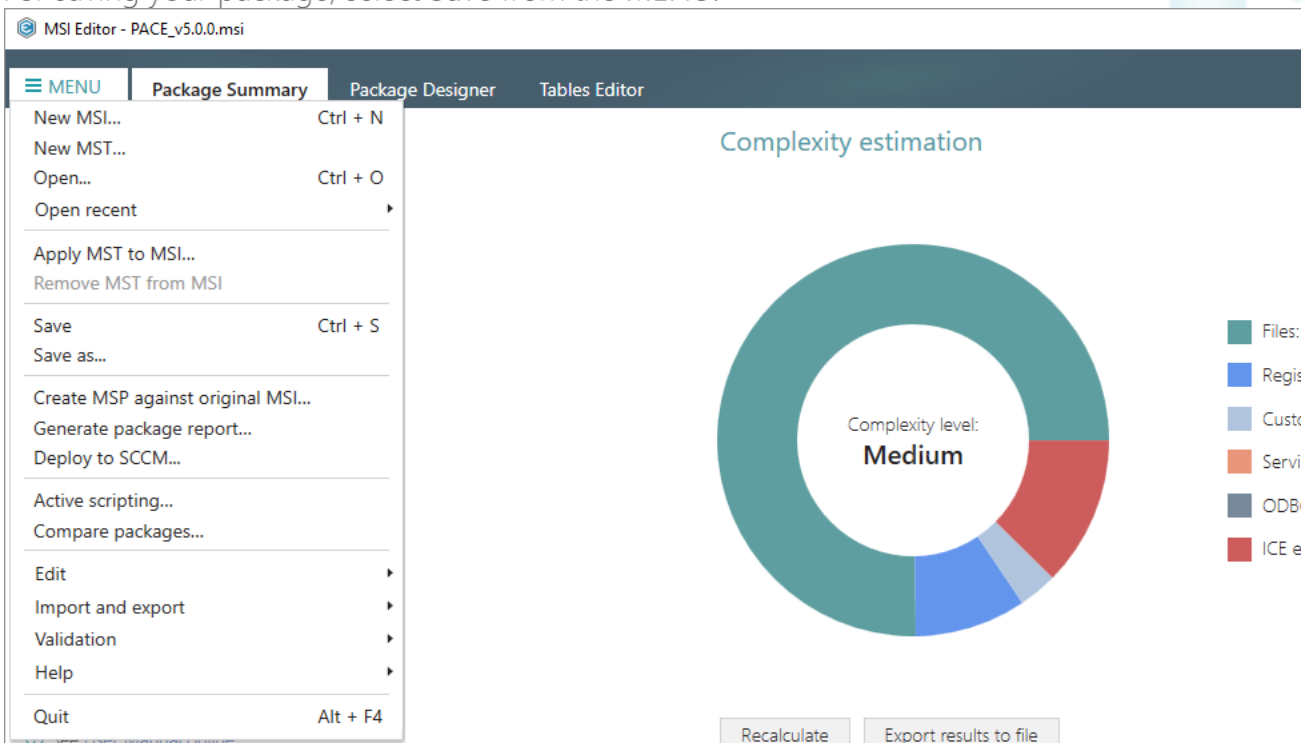
- Select **Drop table** from the context menu of a table, which you want to remove.
NOTE The read-only system tables could not be removed.



- In case other non-empty tables refer to the selected table, you will get a compete list of all linked items, which will be affected by the removed tables. By default, all linked items will be removed along with the selected table in order to support the MSI database integrity constraints. If needed, untick those linked items, which you do not want to remove and click OK to continue.



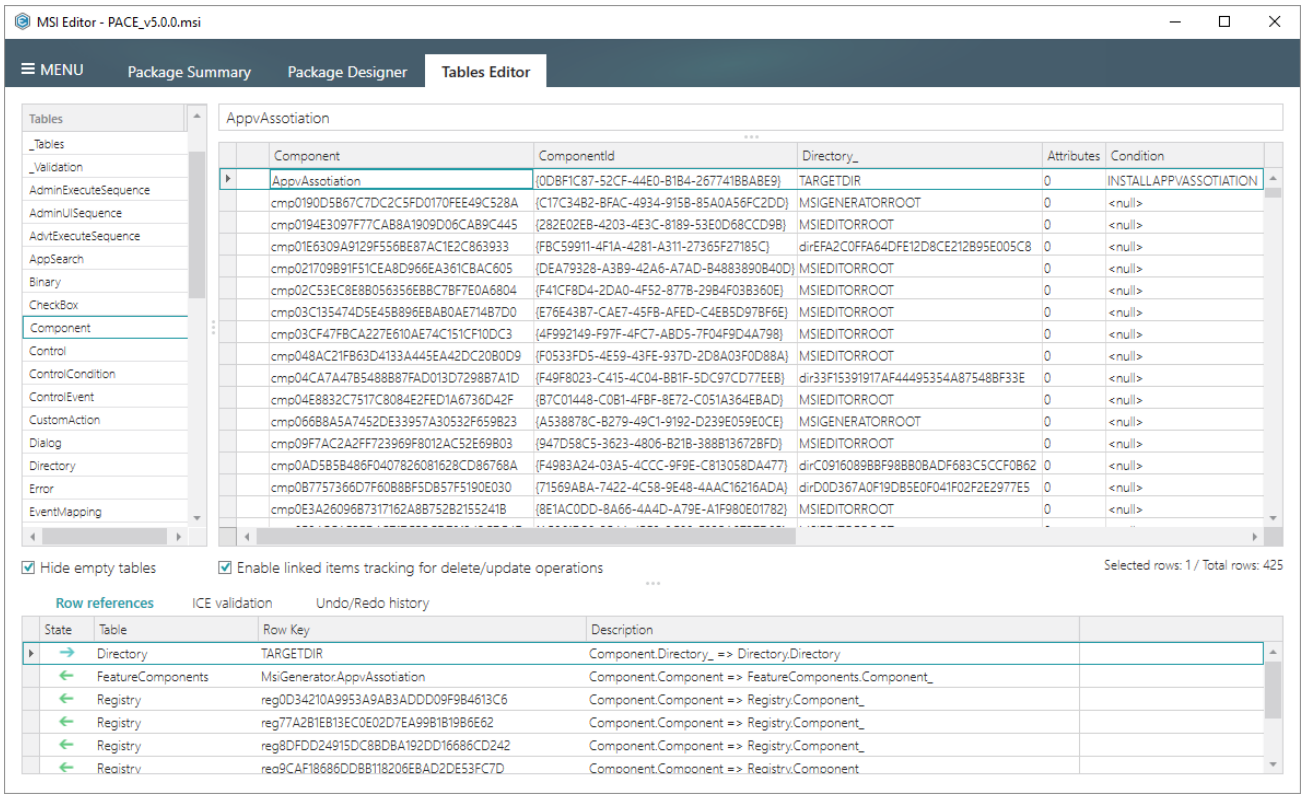
- For saving your package, select Save from the MENU.



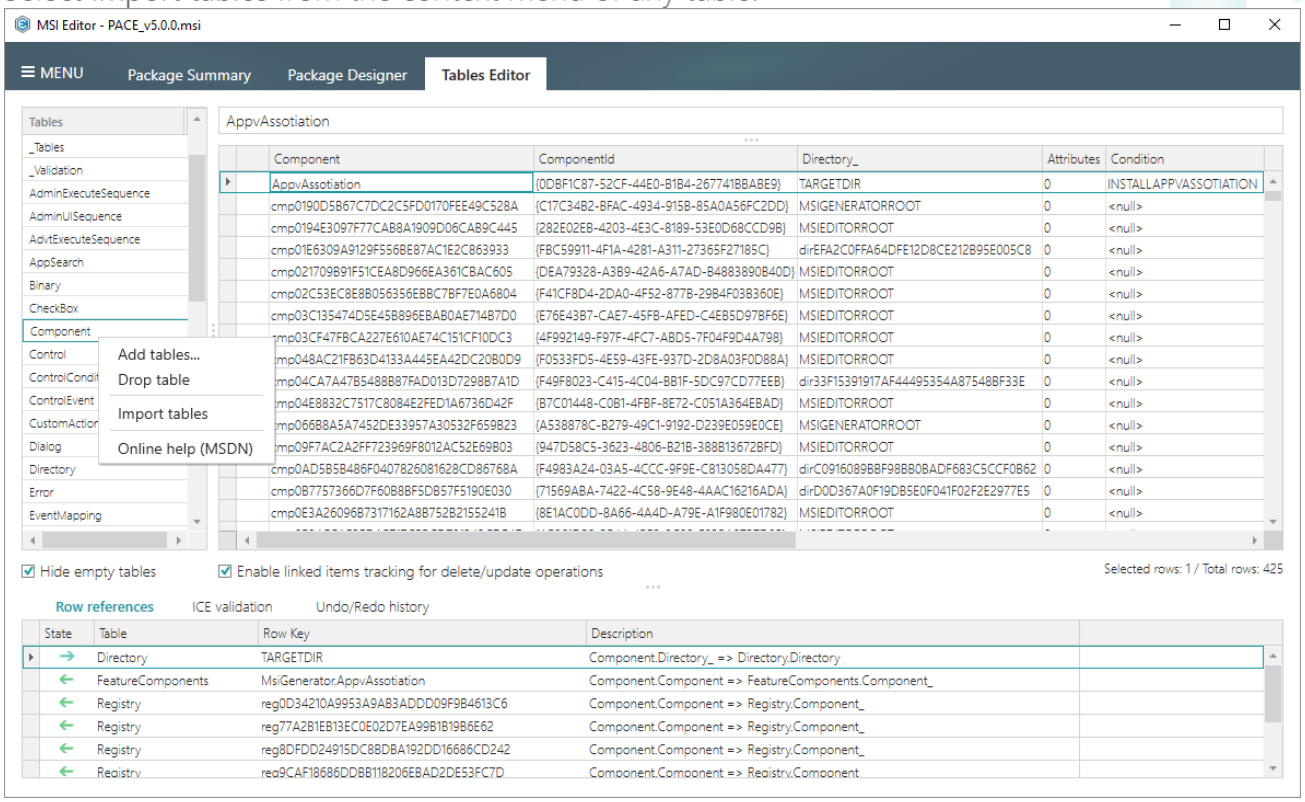
3.7.14.3 Import Tables (IDT)

Import standard or custom tables with or without content from the IDT files.

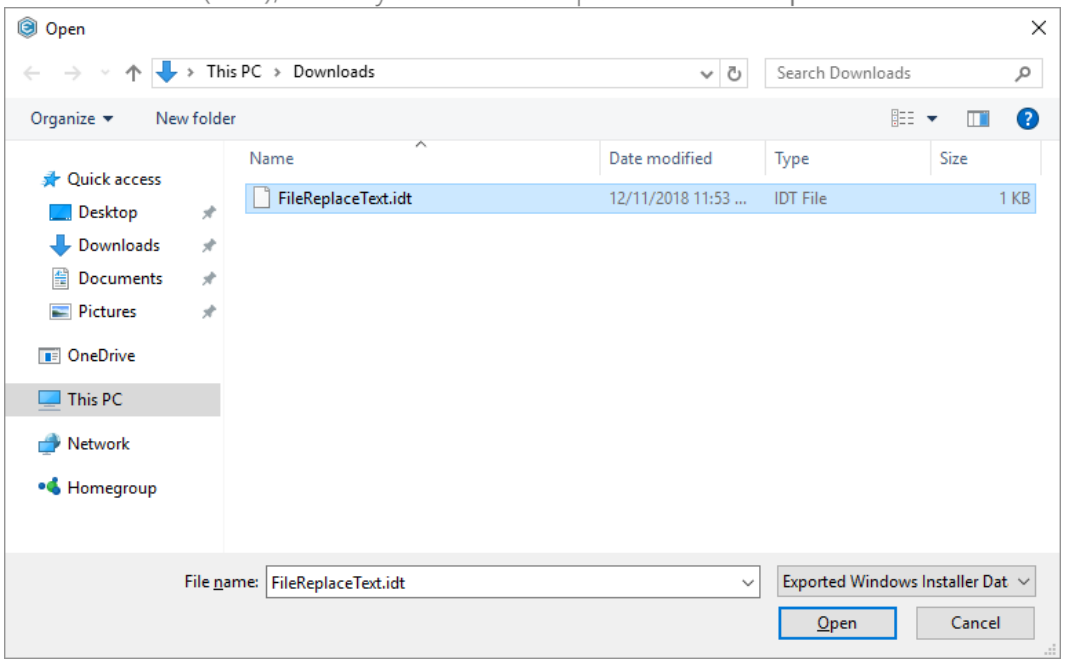
- Go to the Tables Editor tab.



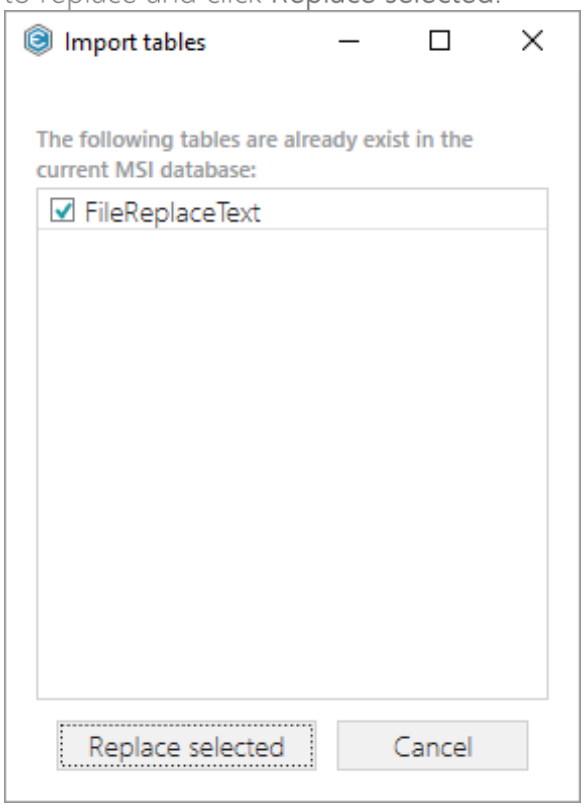
- Select Import tables from the context menu of any table.



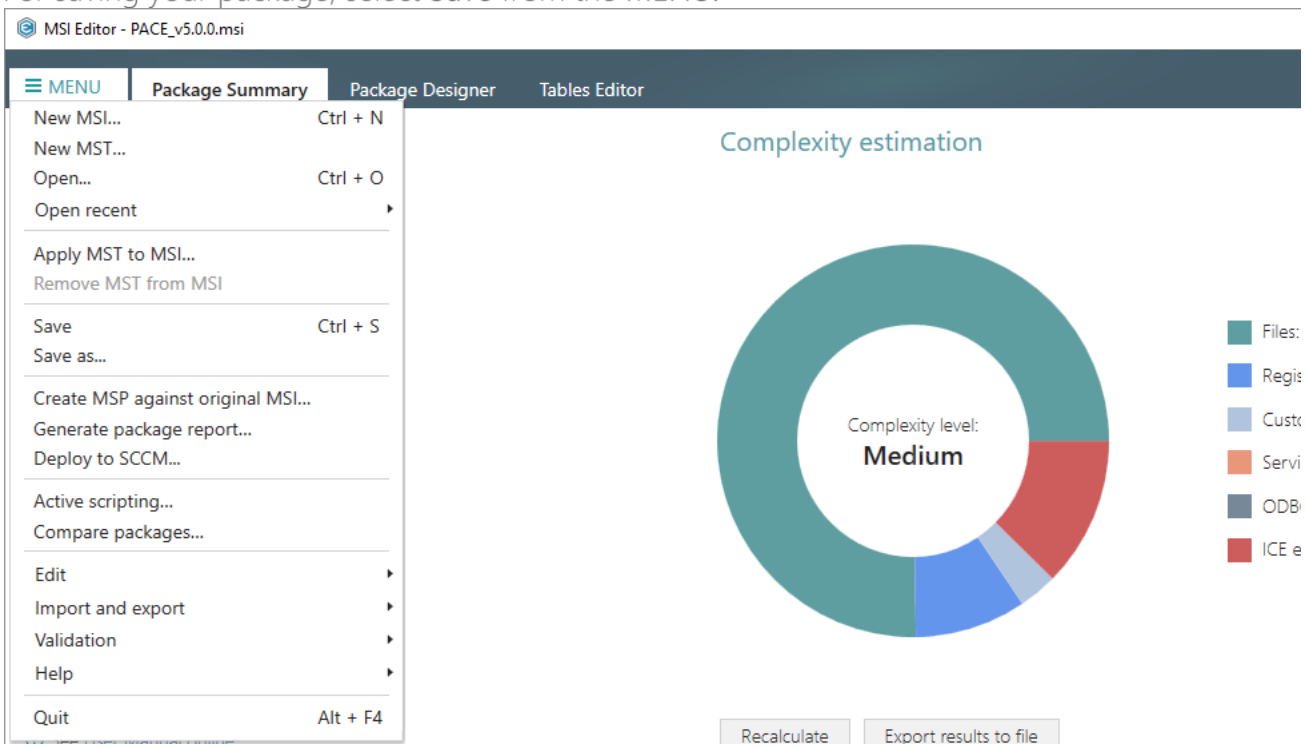
- Choose tables (*.idt), which you want to import and click Open.



- If the package already contains a table with the same name, you will be informed about this by the Import tables window. To replace the existing tables, tick tables from the list you want to replace and click Replace selected.



- For saving your package, select Save from the MENU.



3.7.15 Merge Modules

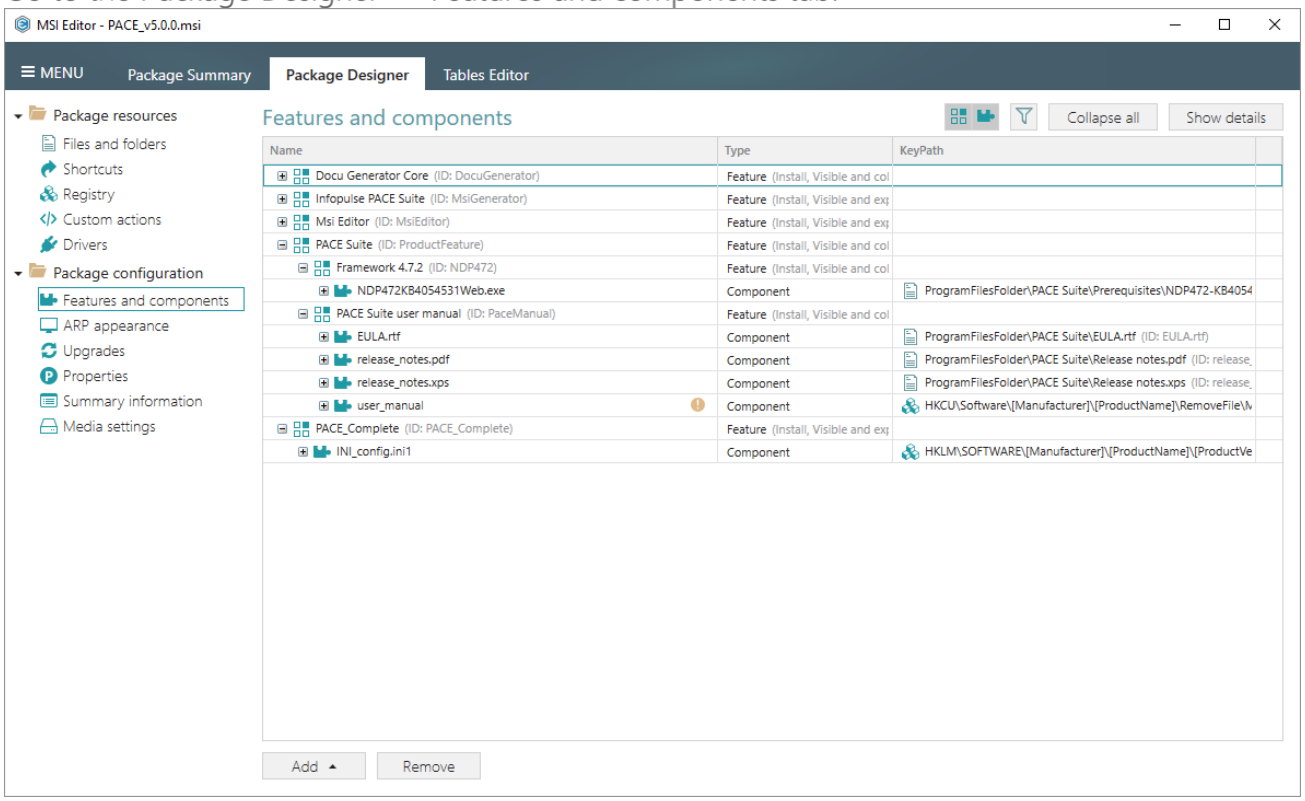
Choose a scenario that better suits your needs:

- Create MSM, described in section 3.7.15.1
  Export component(s) from the opened MSI package to the new merge module (MSM).
- Import MSM to Package, described in section 3.7.15.2
  Import merge modules (MSM) from the file system to the opened MSI package to deliver the shared resources properly.

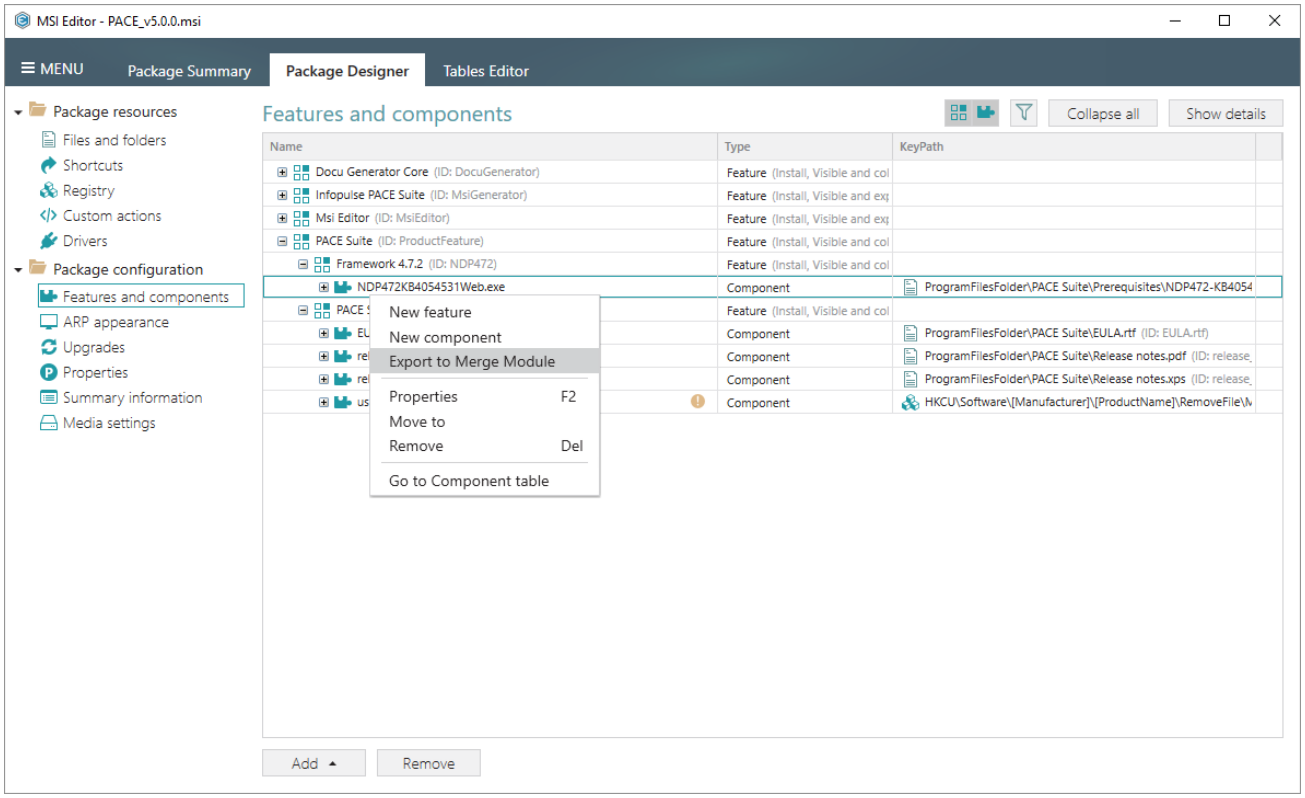
3.7.15.1 Create MSM

Export component(s) from the opened MSI package to the new merge module (MSM).

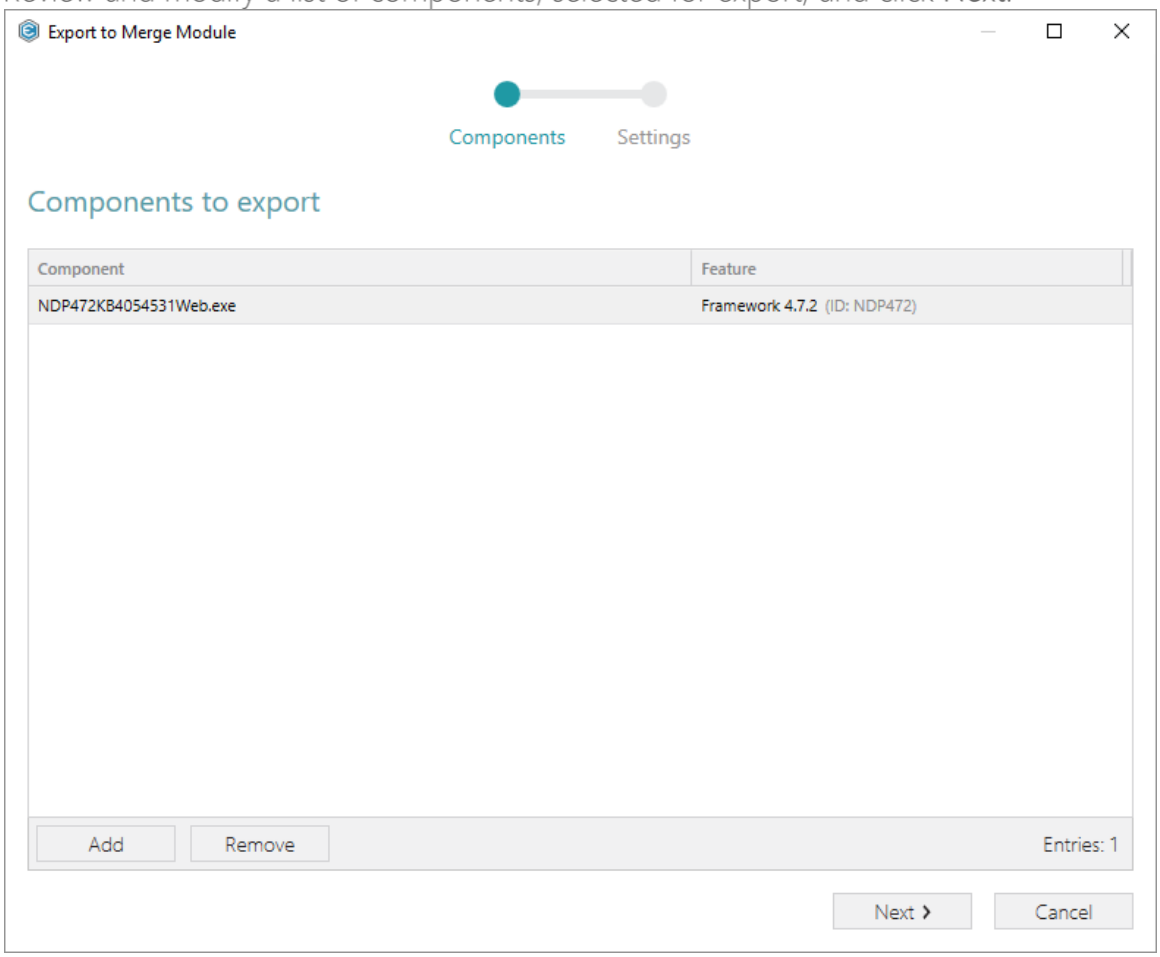
- Go to the Package Designer -> Features and components tab.



- Select **Export to Merge Module** from the context menu of a component, which you want to export to MSM, or from the context menu of a feature to export all components of this feature.

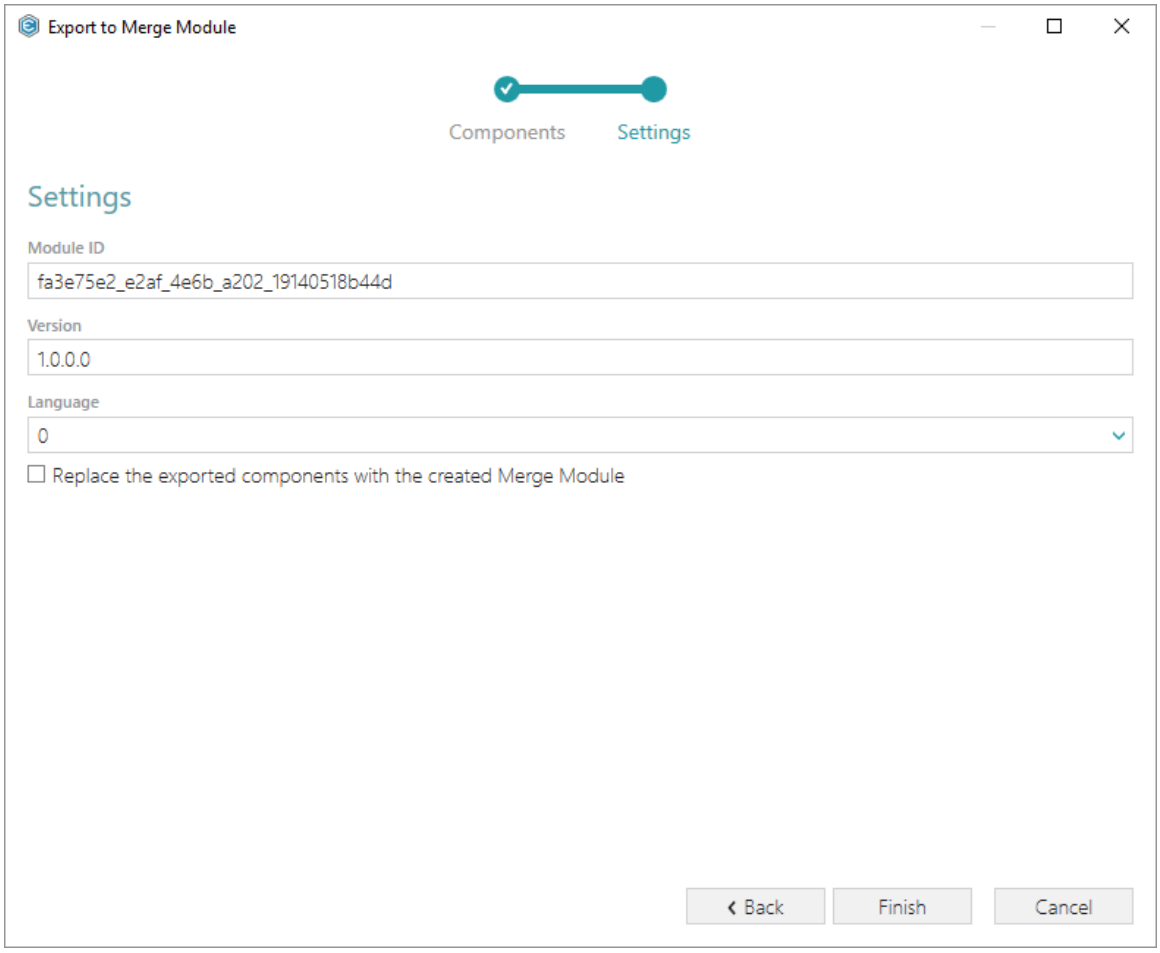


- Review and modify a list of components, selected for export, and click **Next**.

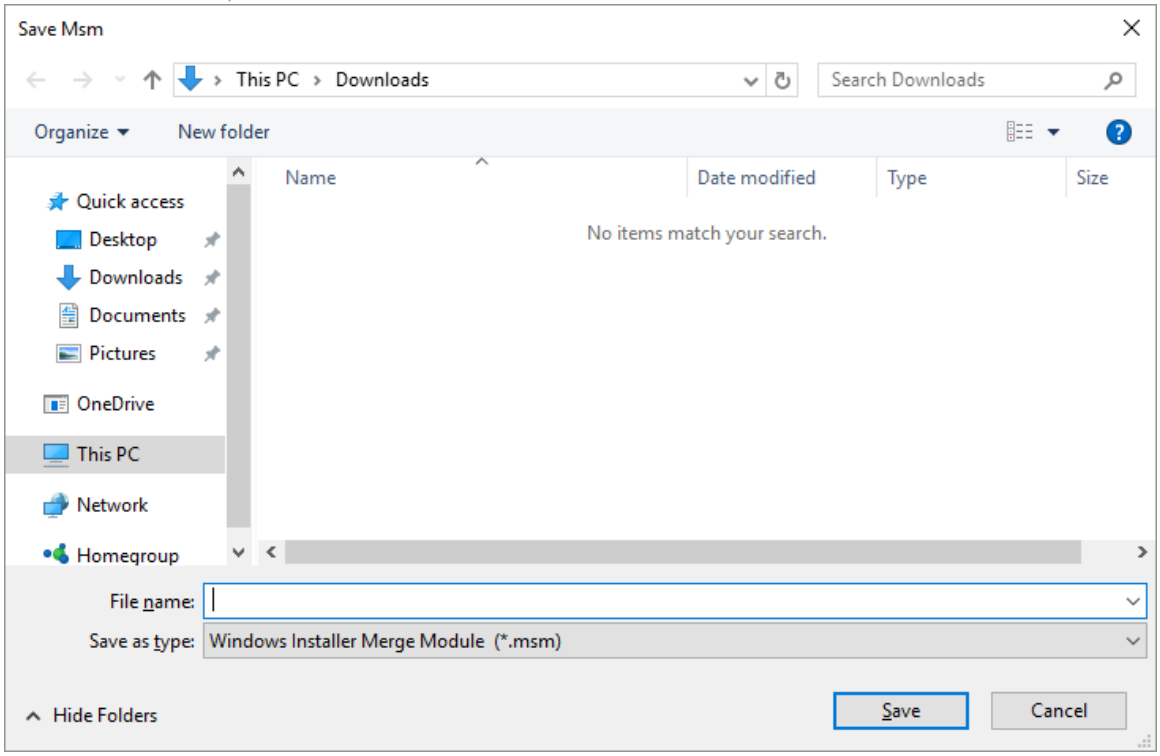


- If needed, update a new module ID, version and language. To replace the exported components of MSI package with the created MSM, select the respective option. Afterwards,

click Finish.



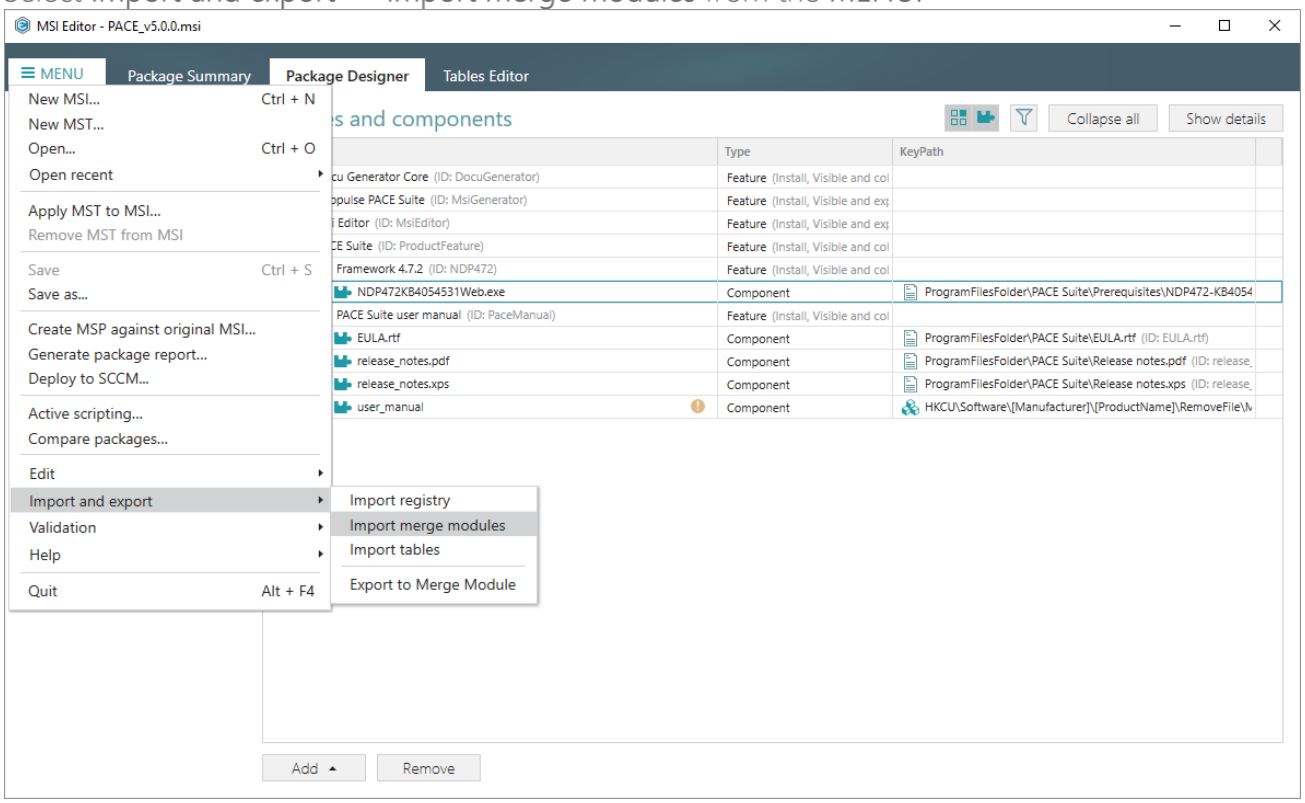
- Enter a file name, select a destination location and click Save.



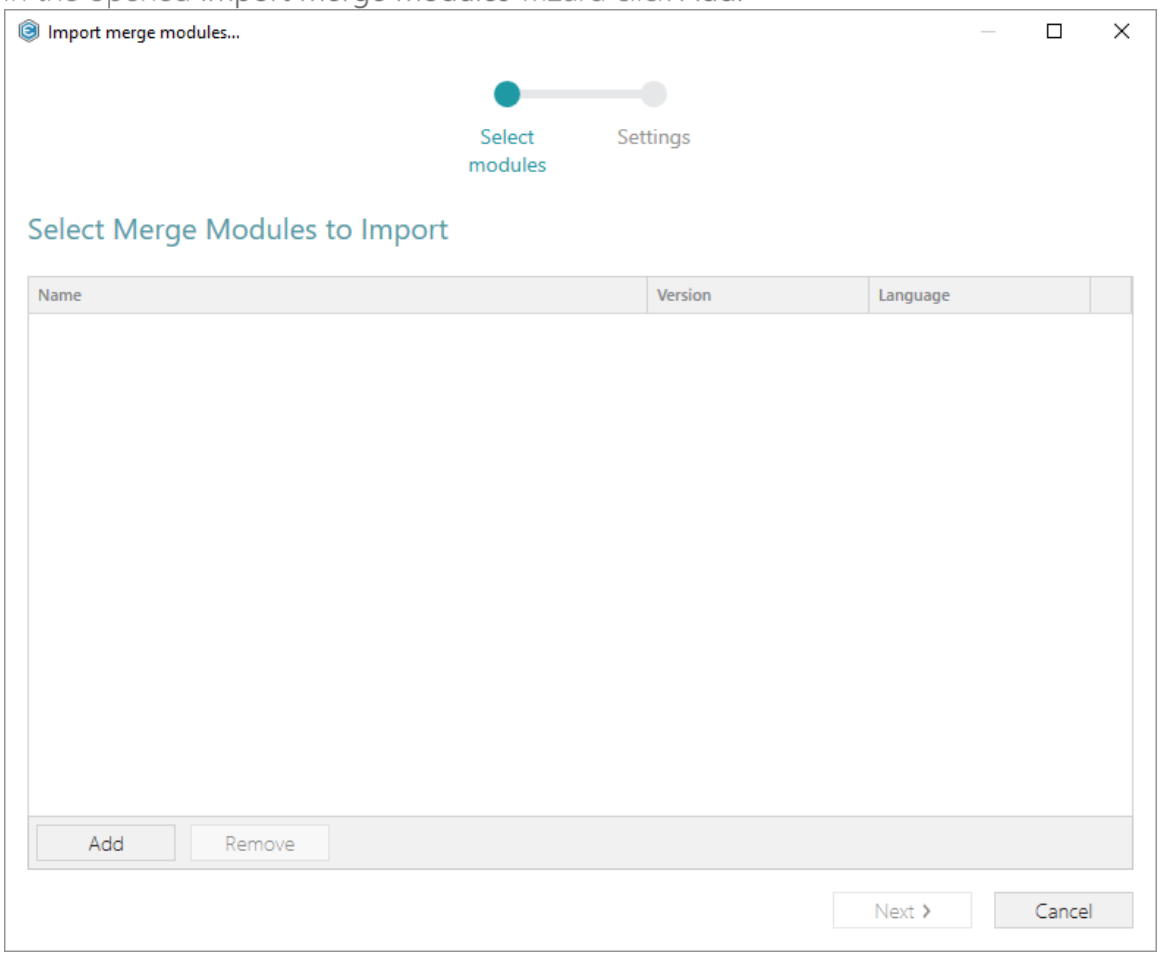
3.7.15.2 Import MSM to Package

Import merge modules (MSM) from the file system to the opened MSI package to deliver the shared resources properly.

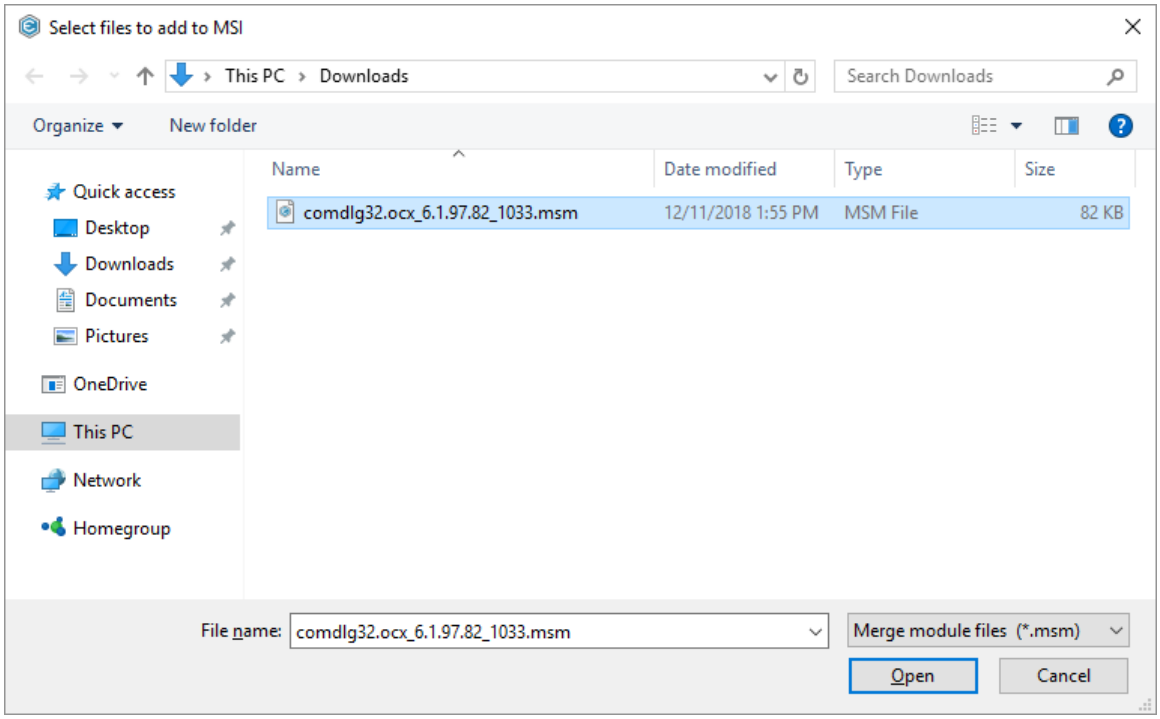
- Select Import and export -> Import merge modules from the MENU.



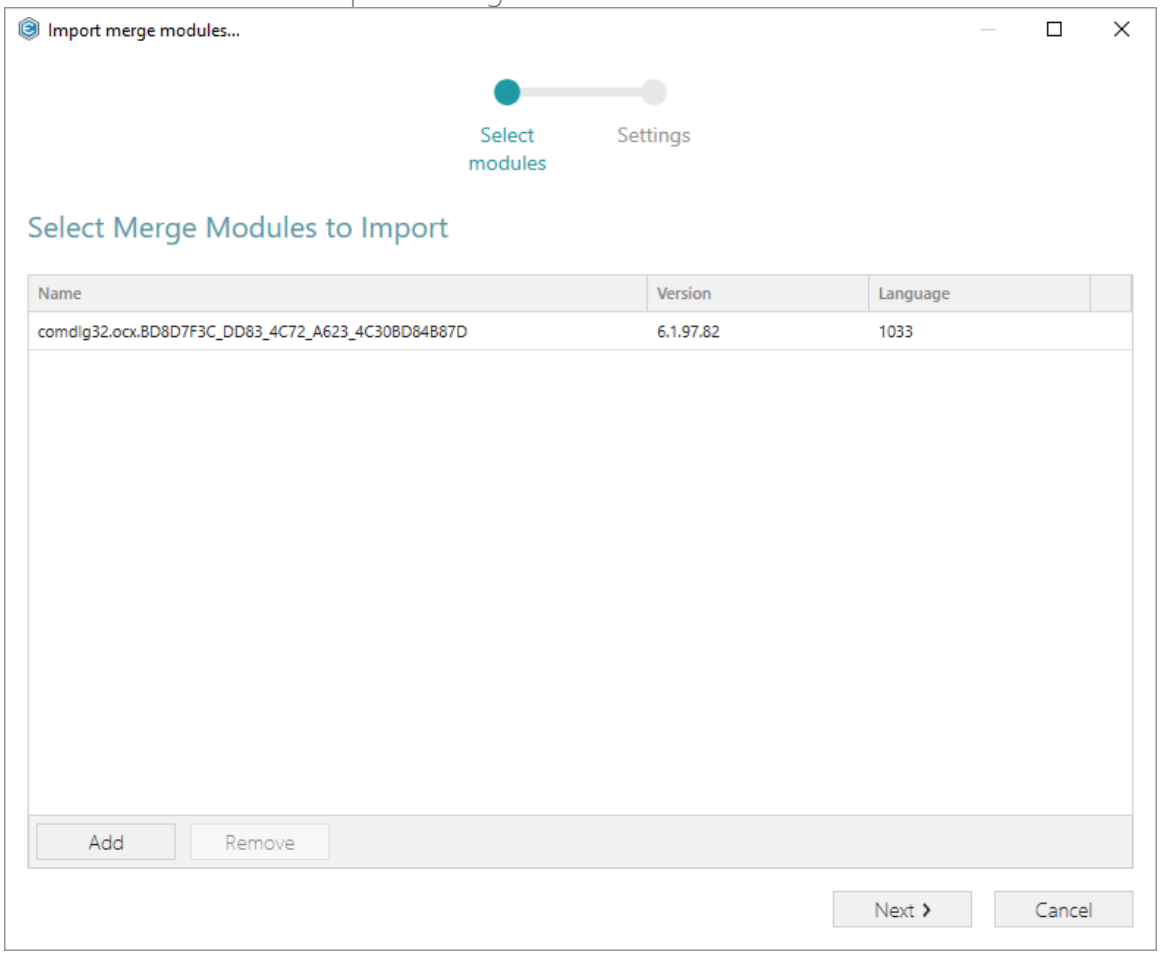
- In the opened Import merge modules wizard click Add.



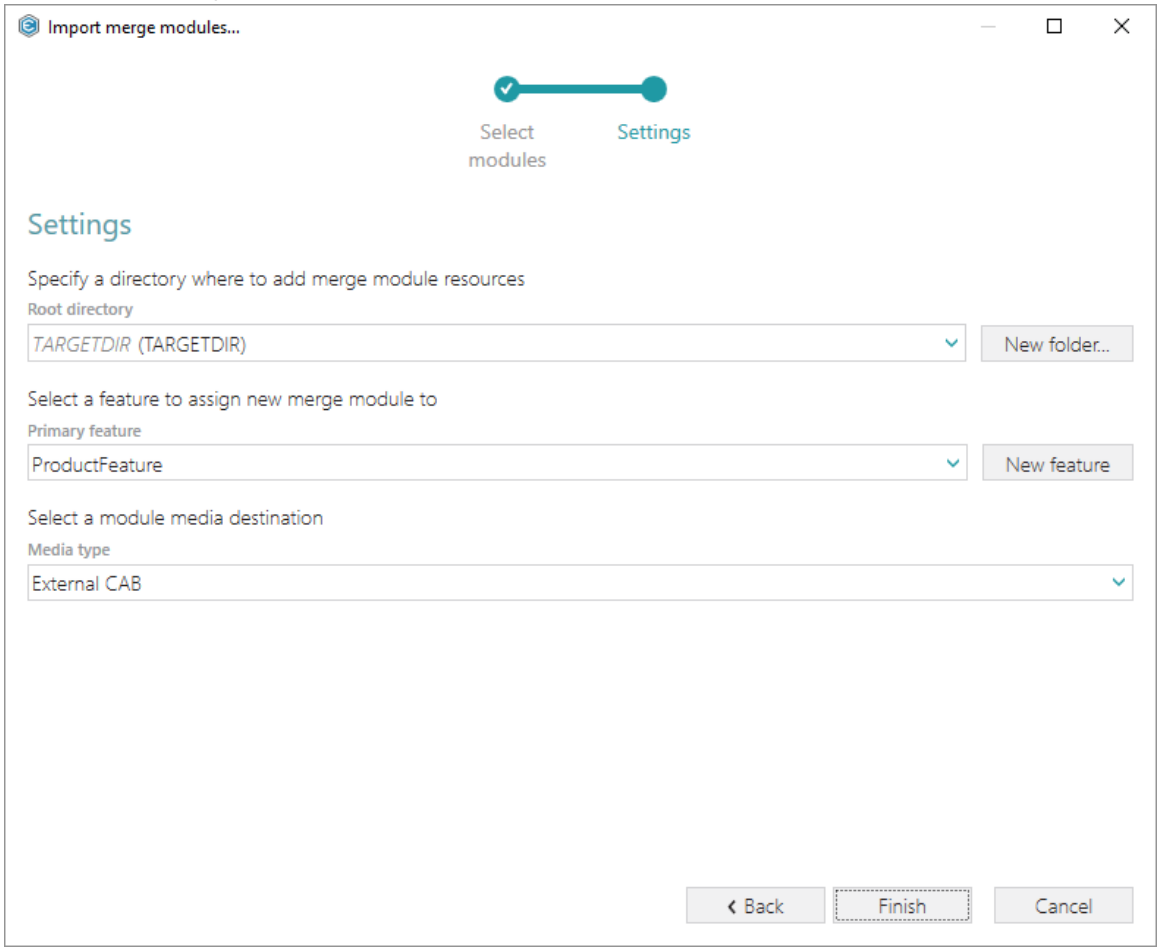
- Choose merge modules (*.msm), which you want to import and click Open. More than one file could be selected at once.



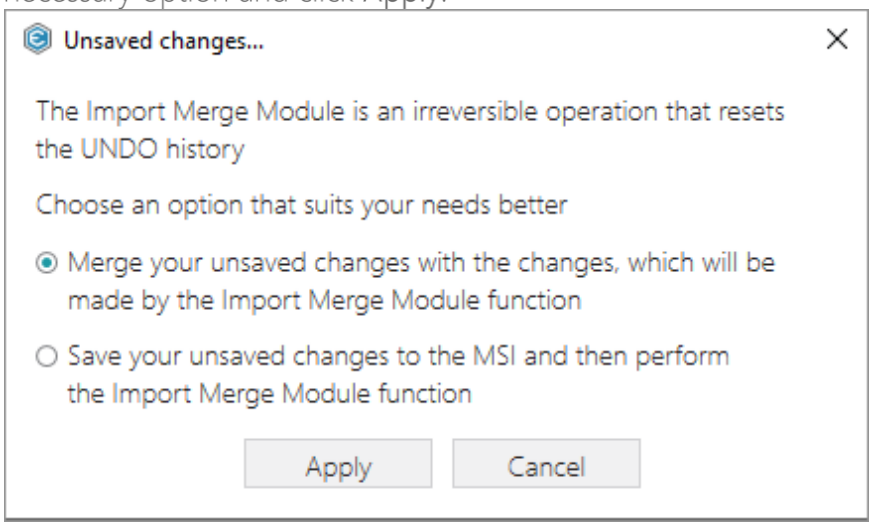
- Click Next to select the import settings.



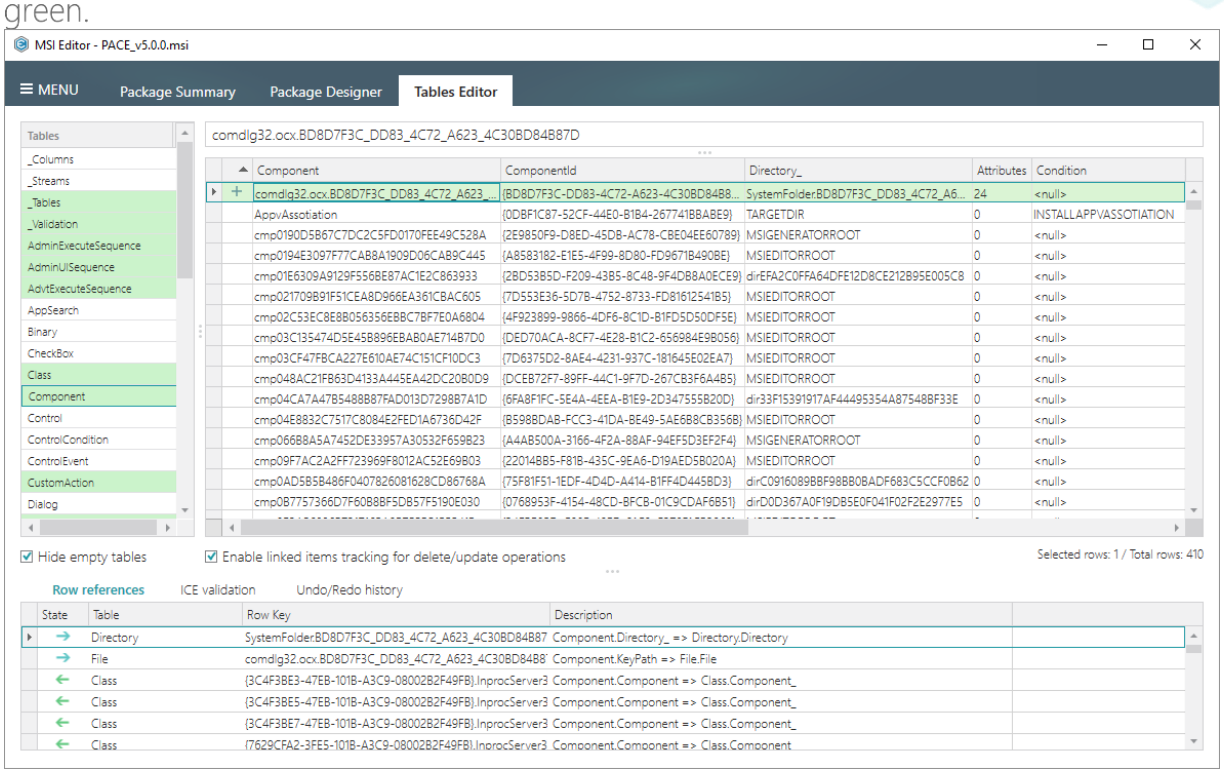
- Select a parent folder for the merge module resources from the **Root directory** field (in most cases this path is re-defined via the Custom action inside the MSM), feature from the **Primary feature** field, to which MSM components will be assigned, and compression type for the MSM files. Afterwards, click **Finish**.



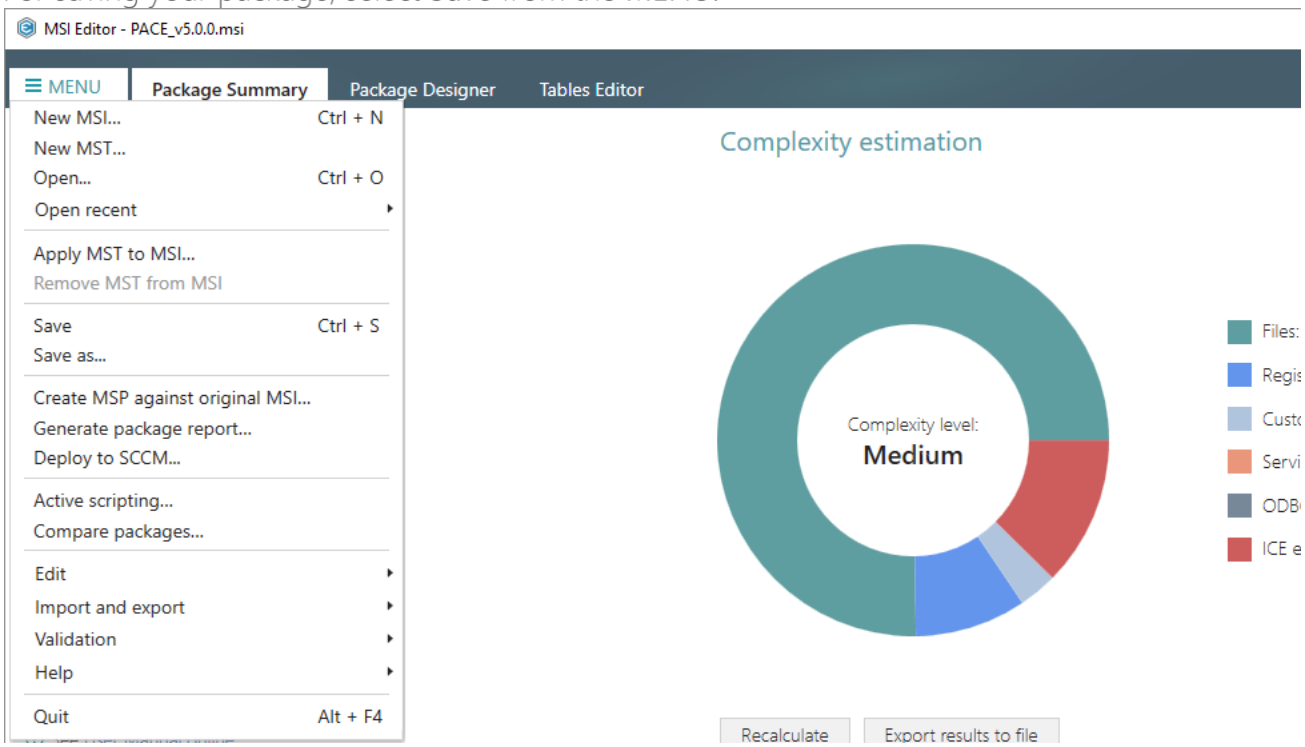
- If you have the unsaved changes, you will be asked what to do with them because the **Import merge modules** is an irreversible operation that resets whole undo/redo history. Select the necessary option and click **Apply**.



- Find the resouces from the imported merge modues on the Tables Editor tab, highlighted in green.



- For saving your package, select Save from the MENU.



3.8 Edit Project

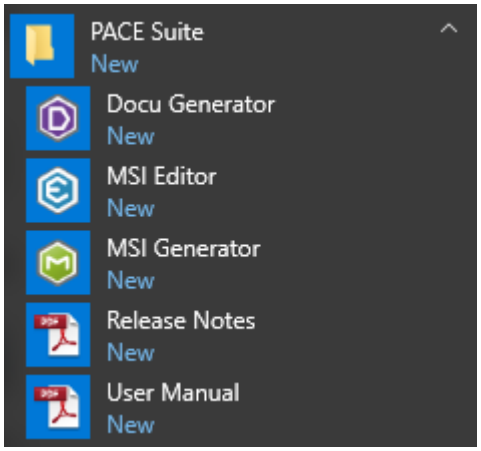
Choose a scenario that better suits your needs:

- For managing File and Folders see section 3.8.2
  Exclude/include captured files, add new folders, files.
- For managing Registry see section 3.8.3
  Exclude/include captured registry entries, import and export registry.
- For managing Services see section 3.8.4
  Exclude/include captured services and manage their details.
- For managing Environment Variables see section 3.8.5
  Manage captured environment variables, manage their details, create new ones.
- For managing MSI Settings see section 3.8.6
  Select the MSI specific settings for creting the MSI packages.
- For managing MSIX/APPX Settings see section 3.8.7
  Select the MSIX/APPX specific settings for creting the MSIX/APPX packages.
- For managing APPV Settings see section 3.8.8
  Select the APPV specific settings for creting APPV packages.
- For managing THINAPP Settings see section 3.8.9
  Select the THINAPP specific settings for creting THINAPP packages.

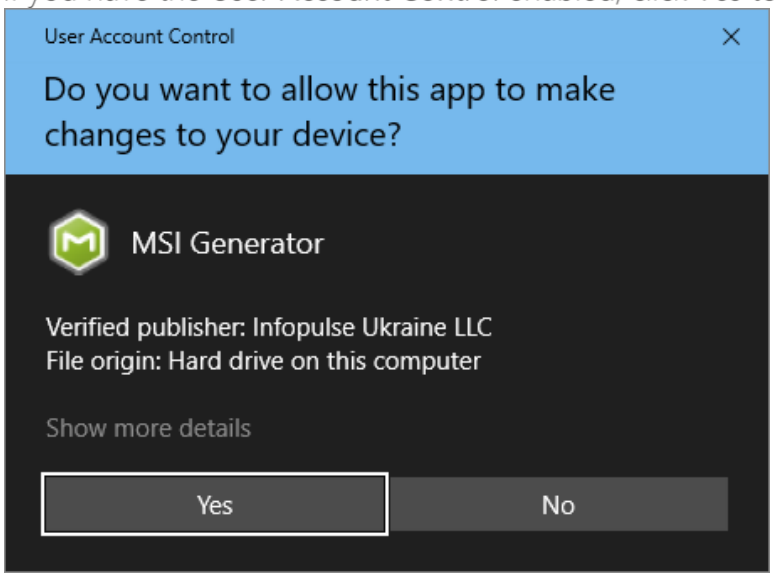
3.8.1 Pre-Condition

Open a project file following the instructions below:

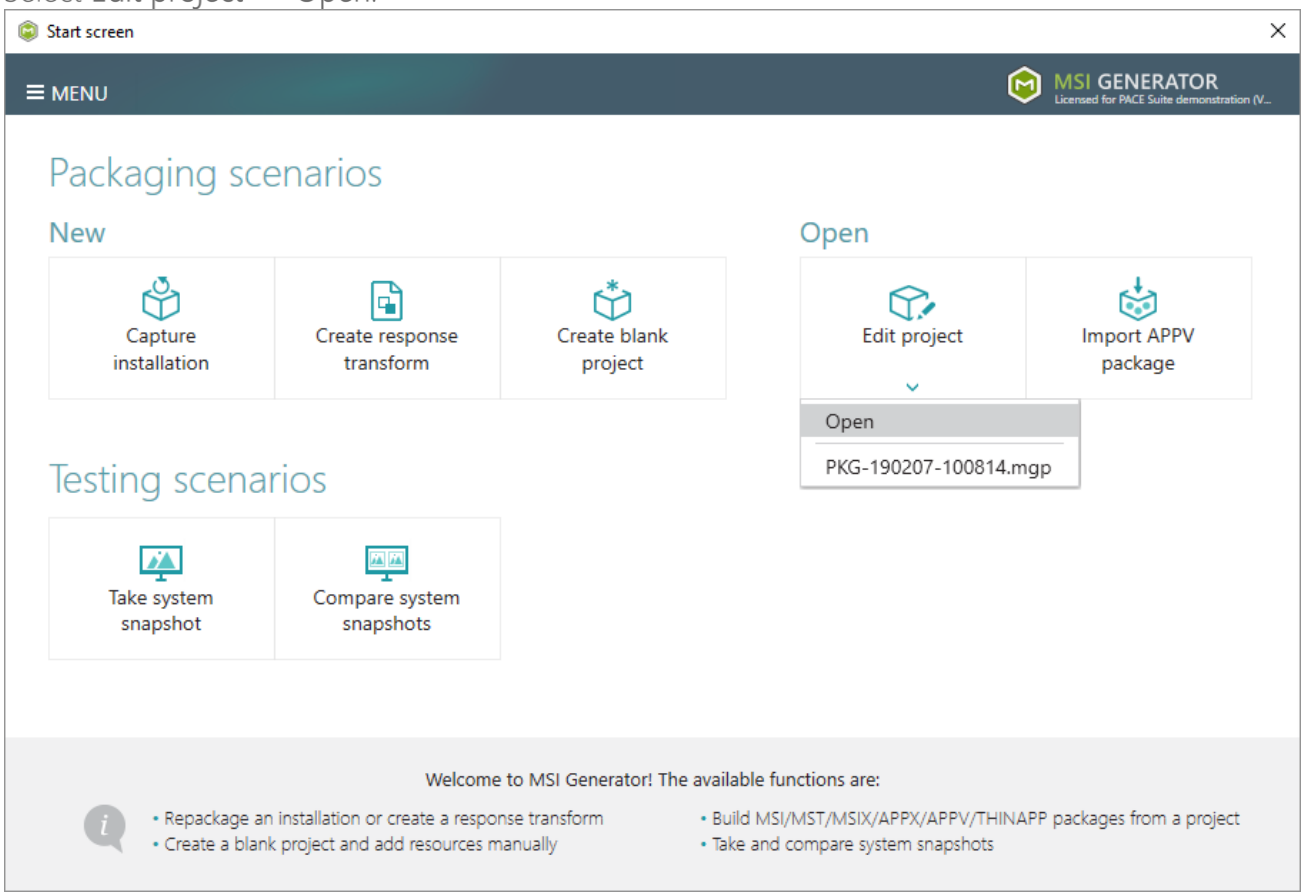
- Launch MSI Generator from the desktop of the start menu shortcut.



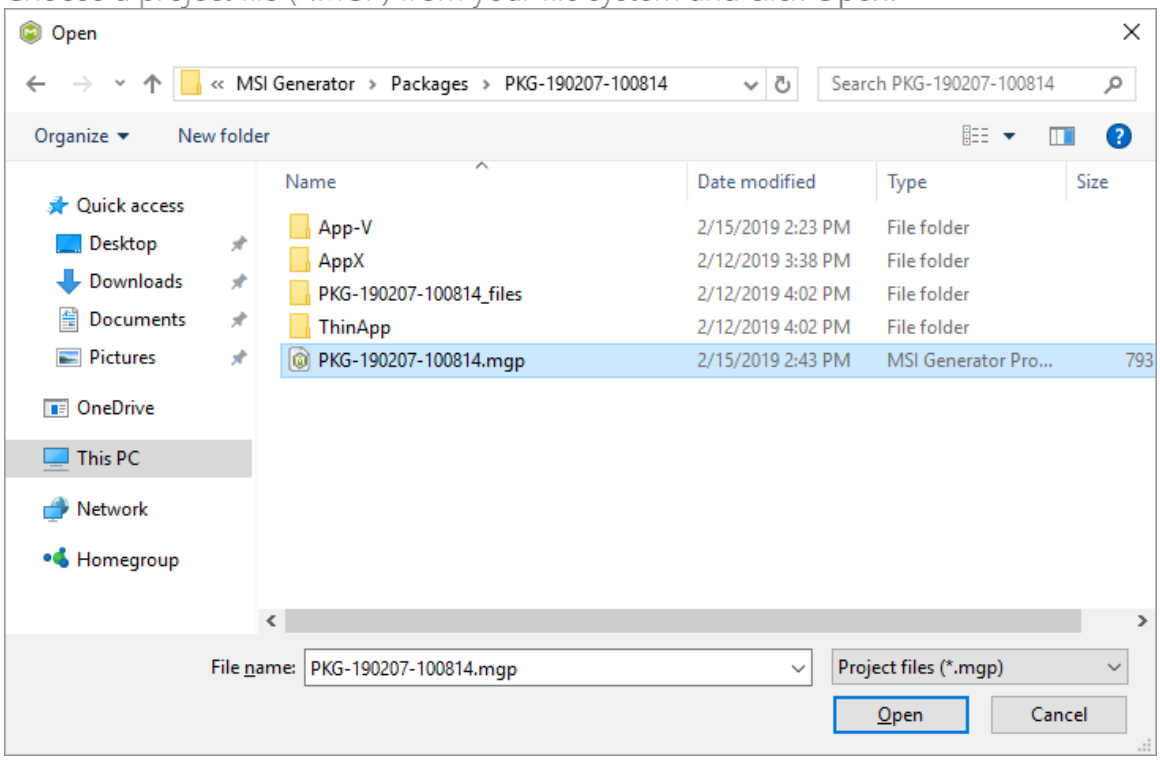
- If you have the User Account Control enabled, click Yes to launch the application.



- Select Edit project -> Open.



- Choose a project file (*.MGP) from your file system and click Open.



3.8.2 File and Folders

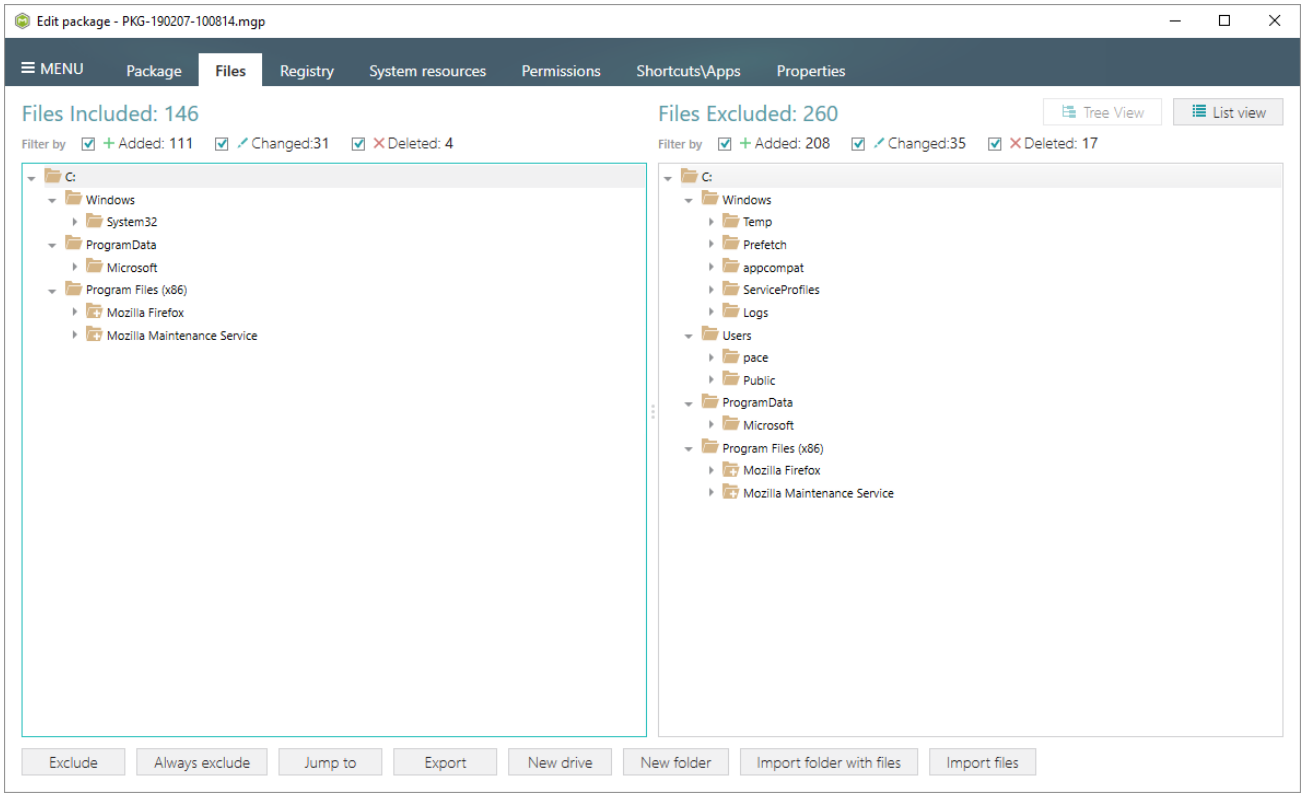
Choose a scenario that better suits your needs:

- Exclude File/Folder, described in section 3.8.2.1
  Exclude the unnecessary files and folders from the project.
- Include Excluded File/Folder, described in section 3.8.2.2
  Include the necessary resources, previously excluded from the project.
- Add Folder, described in section 3.8.2.3
  Create a new empty folder.
- Import Files, described in section 3.8.2.4
  Import files from the file system to the project.

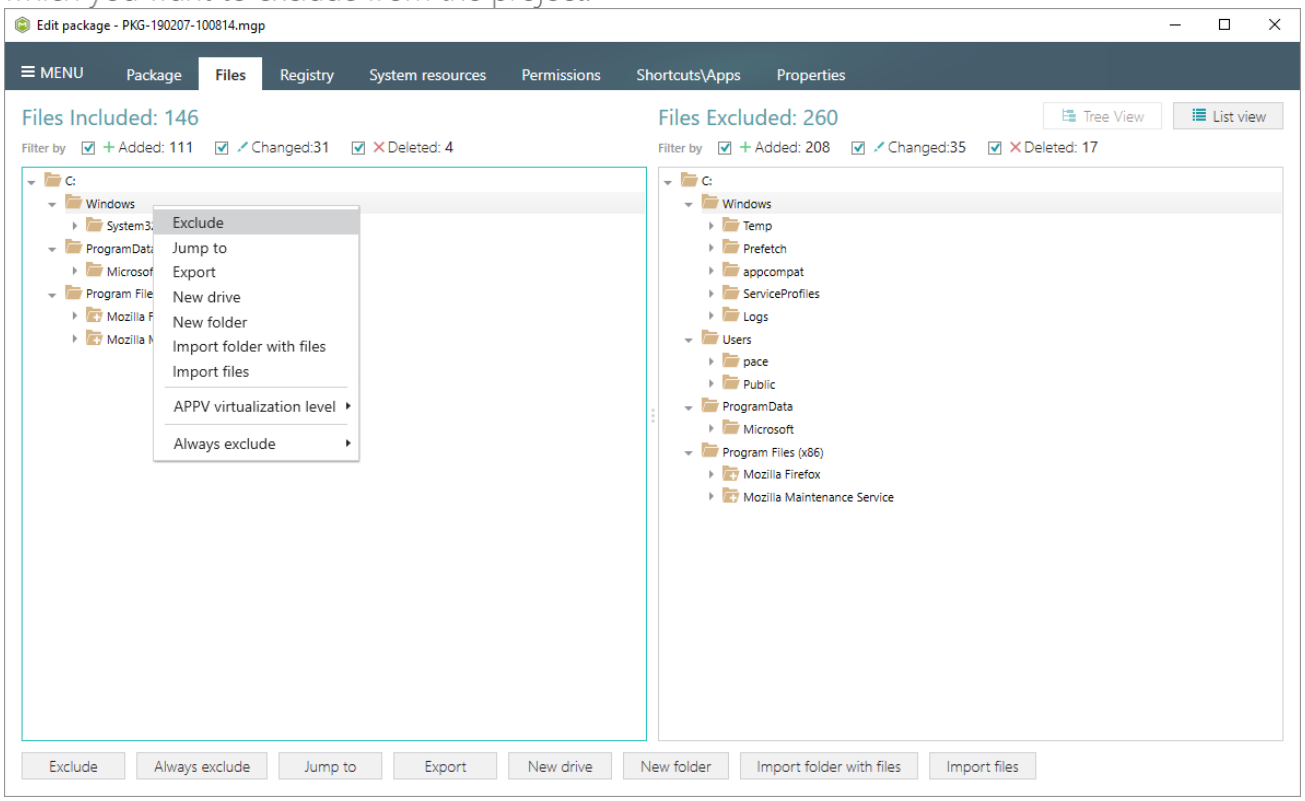
3.8.2.1 Exclude File/Folder

Exclude the unnecessary files and folders from the project.

- Go to the Files tab.



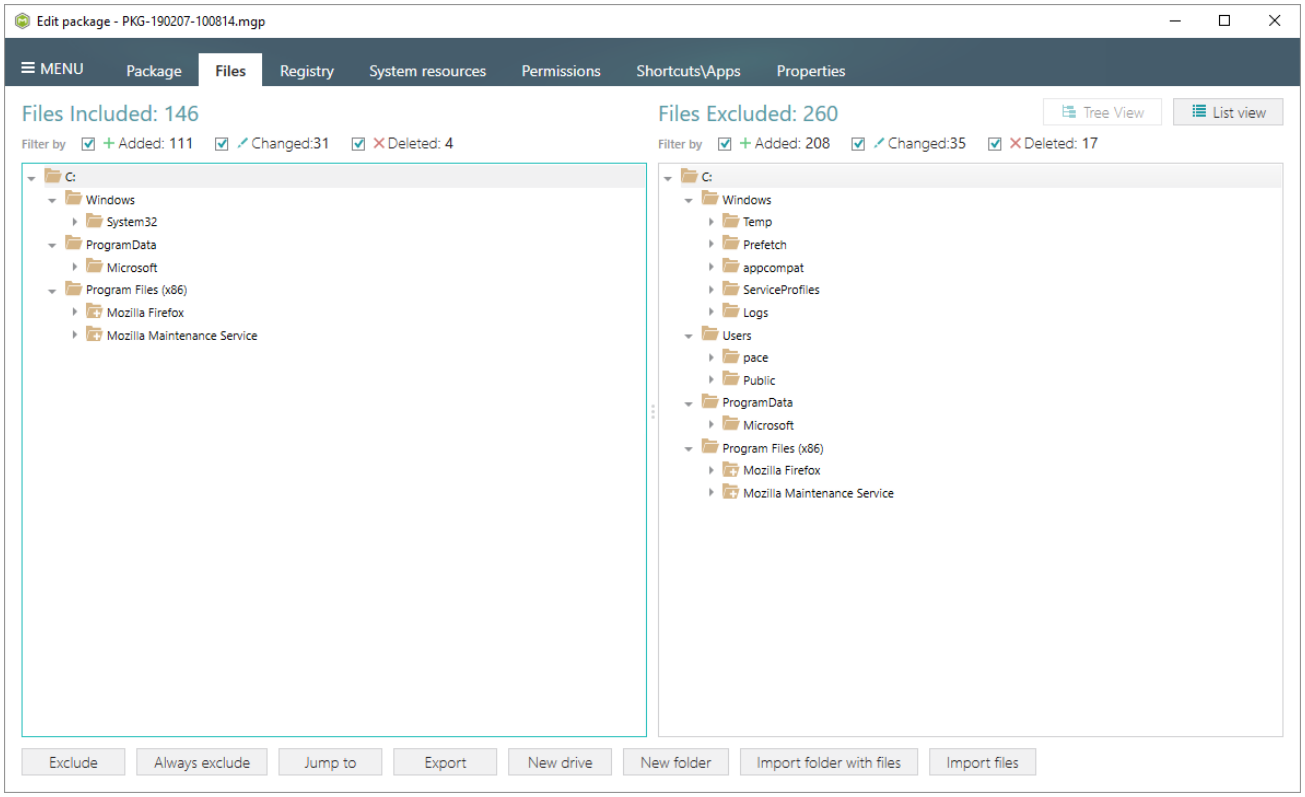
- Select **Exclude** from the context menu of an item, located on the left '**Files Included**' pane, which you want to exclude from the project.



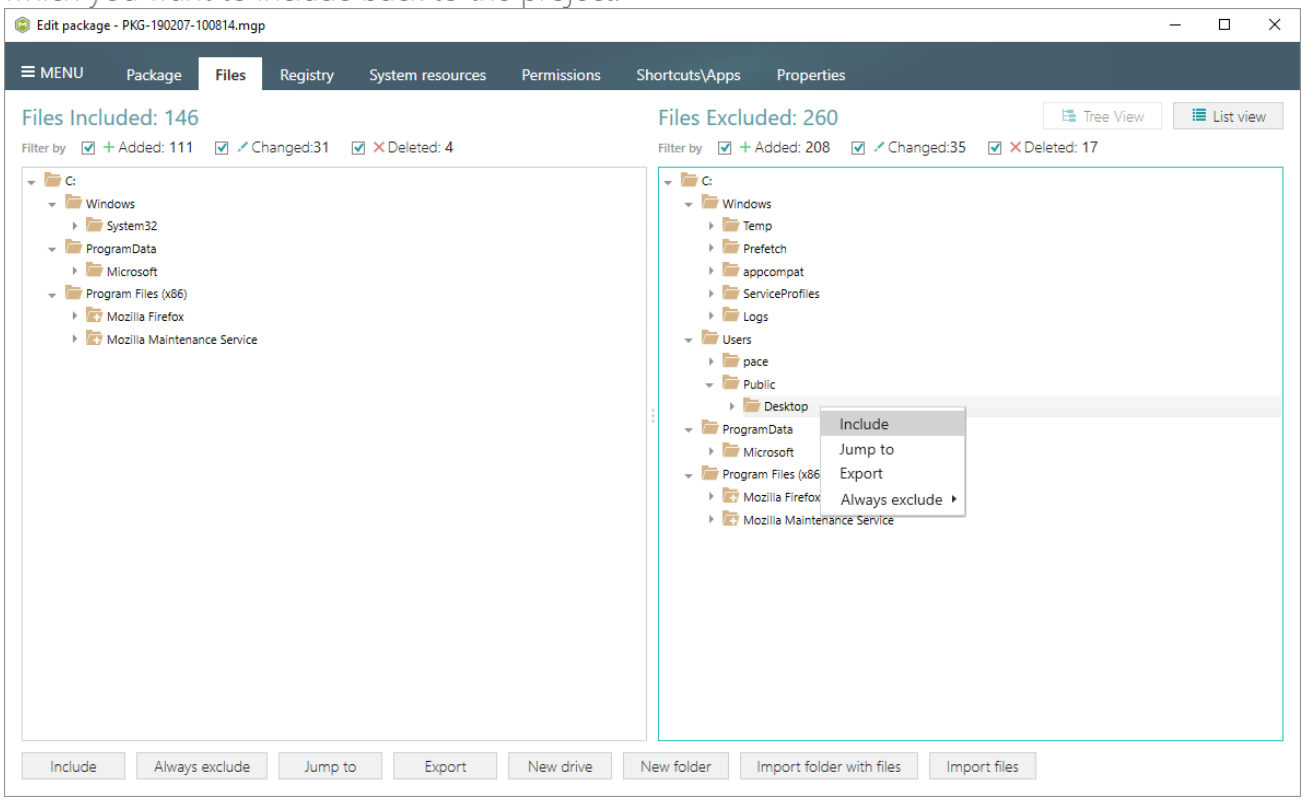
3.8.2.2 Include Excluded File/Folder

Include the necessary files and folders, previously excluded from the project.

- Go to the **Files** tab.



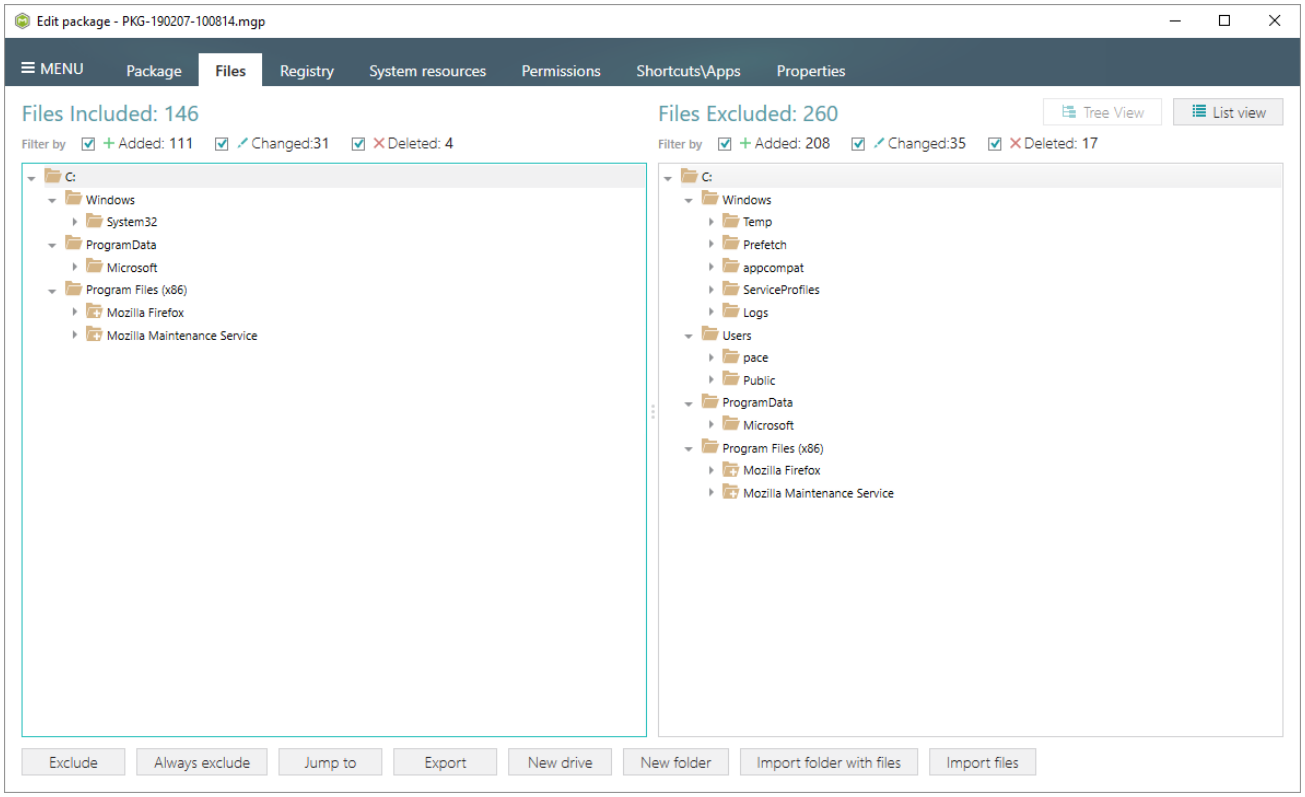
- Select **Include** from the context menu of an item, located on the right '**Files Excluded**' pane, which you want to include back to the project.



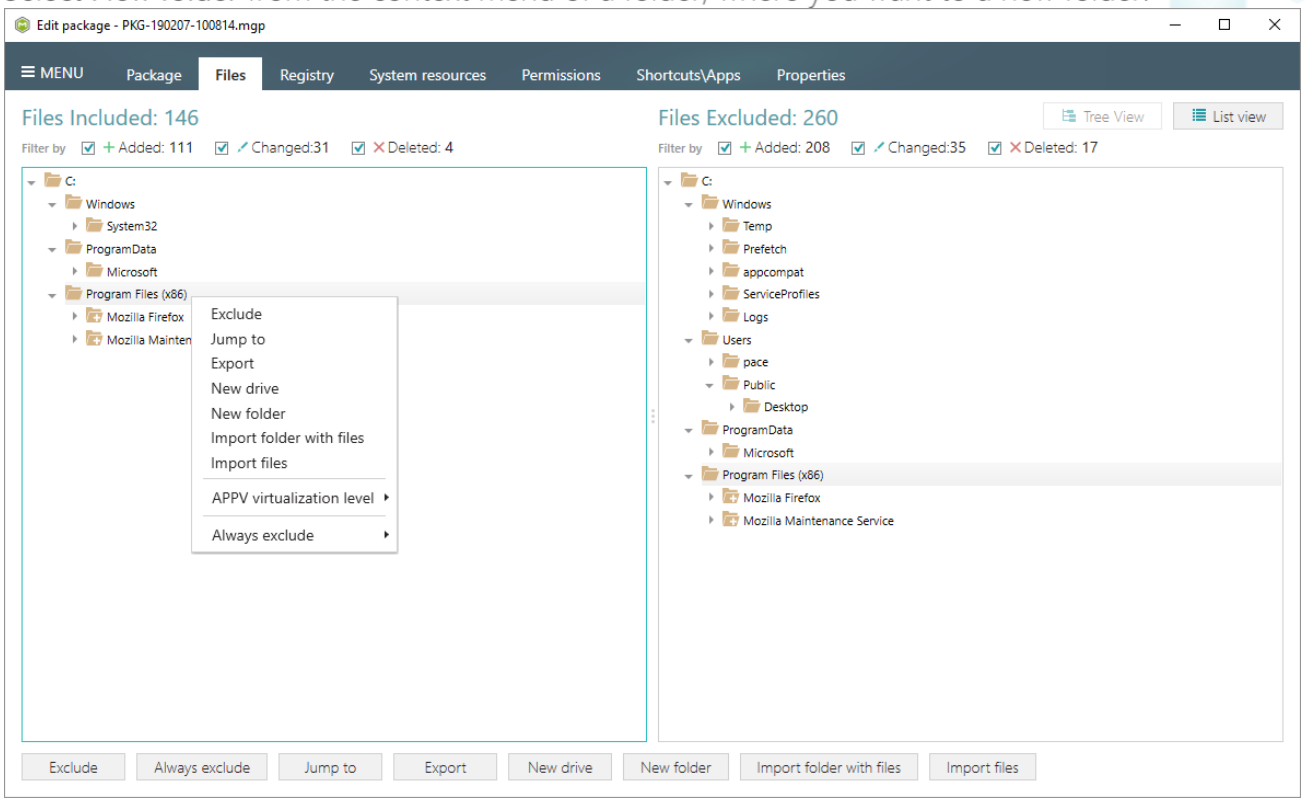
3.8.2.3 Add Folder

Create a new empty folder.

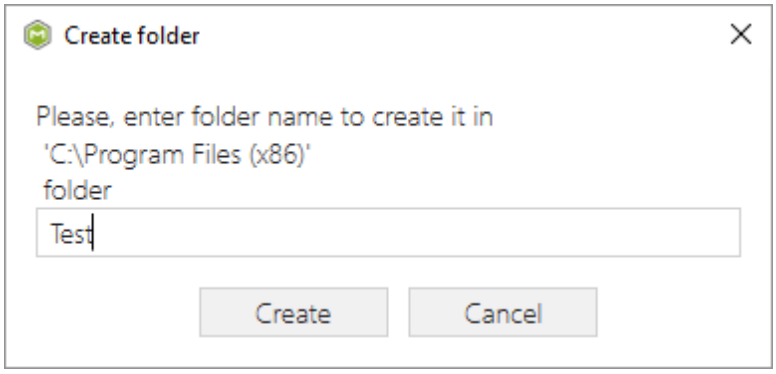
- Go to the **Files** tab.



- Select **New folder** from the context menu of a folder, where you want to a new folder.



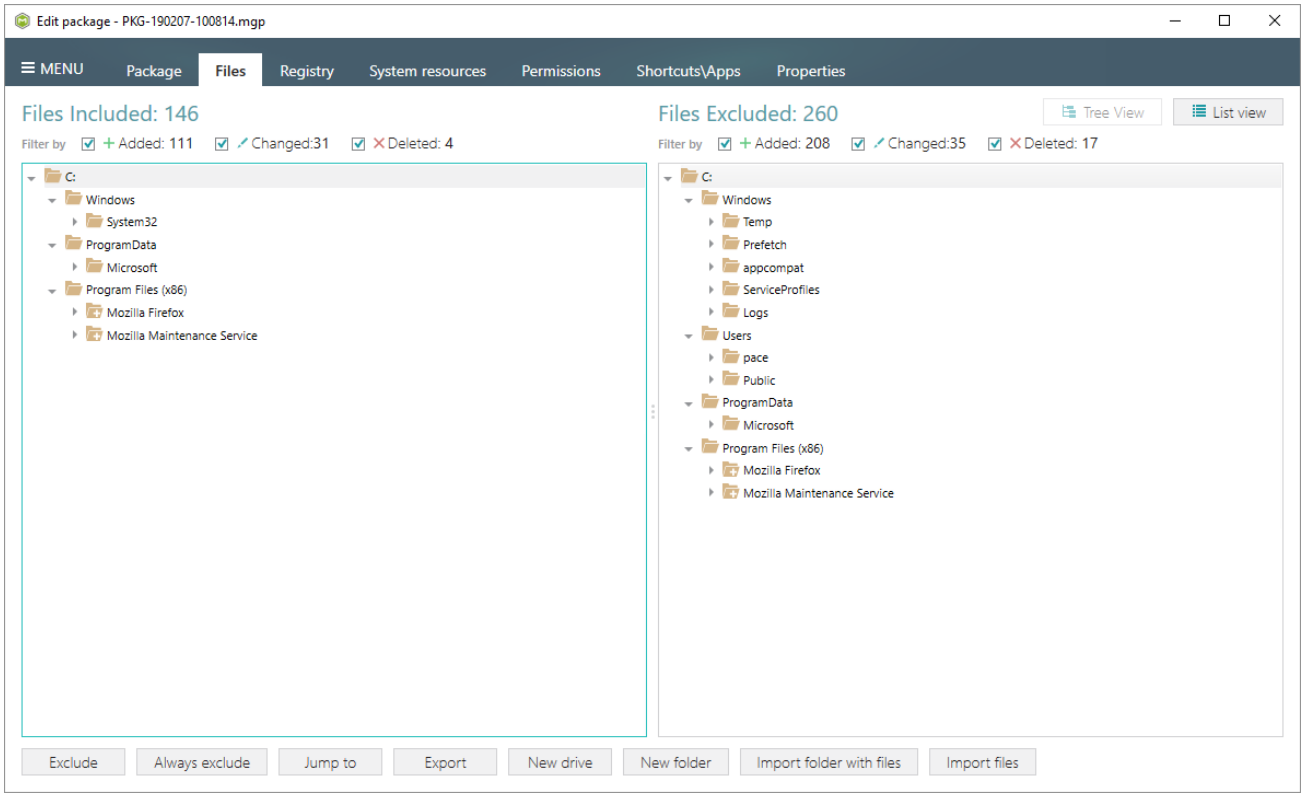
- Enter a folder name and click **Create**.



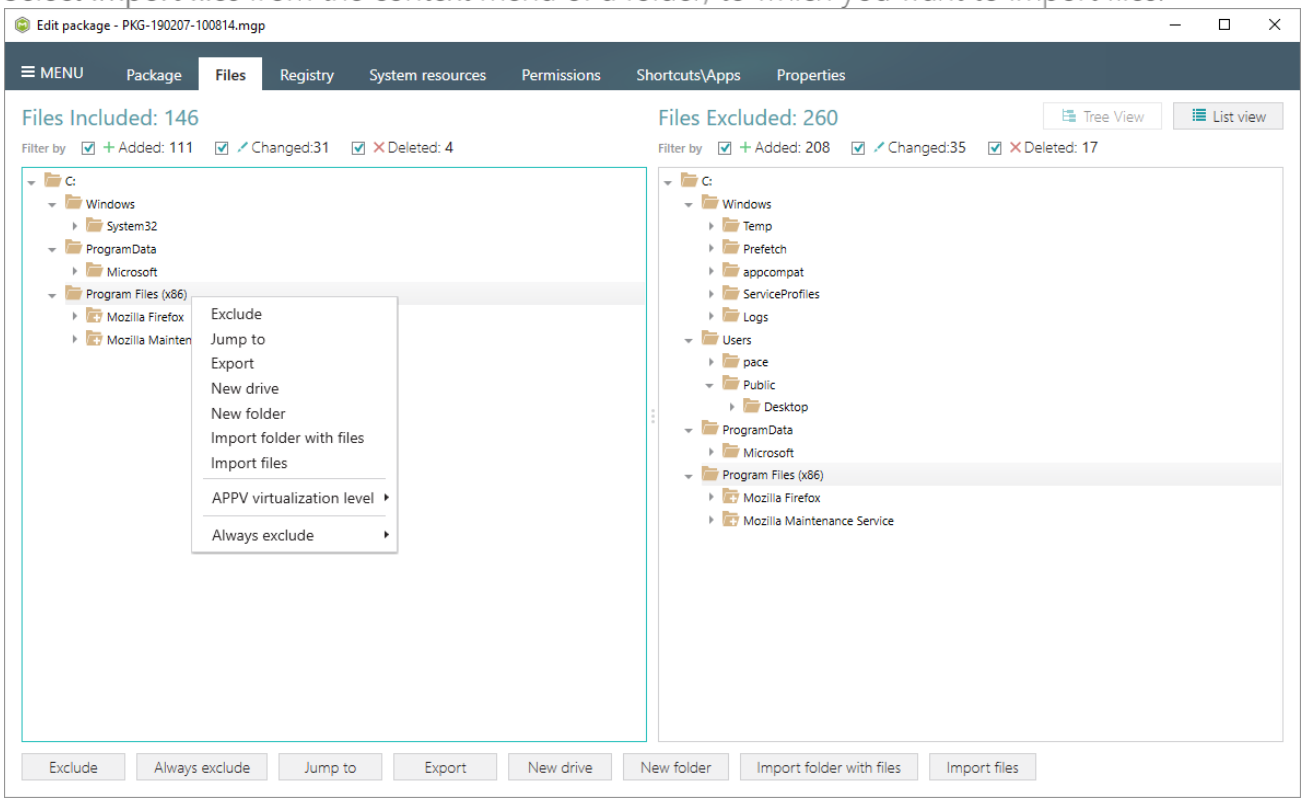
3.8.2.4 Import Files

Import files from the file system to the project.

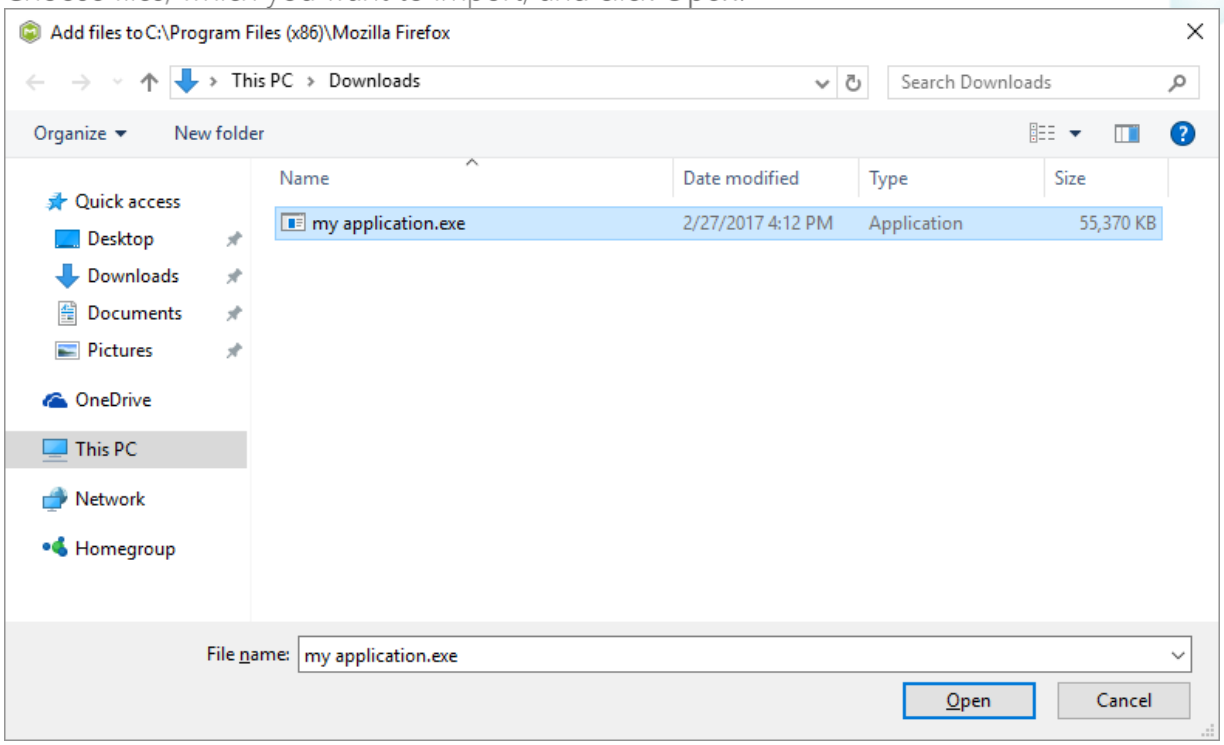
- Go to the **Files** tab.



- Select **Import files** from the context menu of a folder, to which you want to import files.



- Choose files, which you want to import, and click Open.



3.8.3 Registry

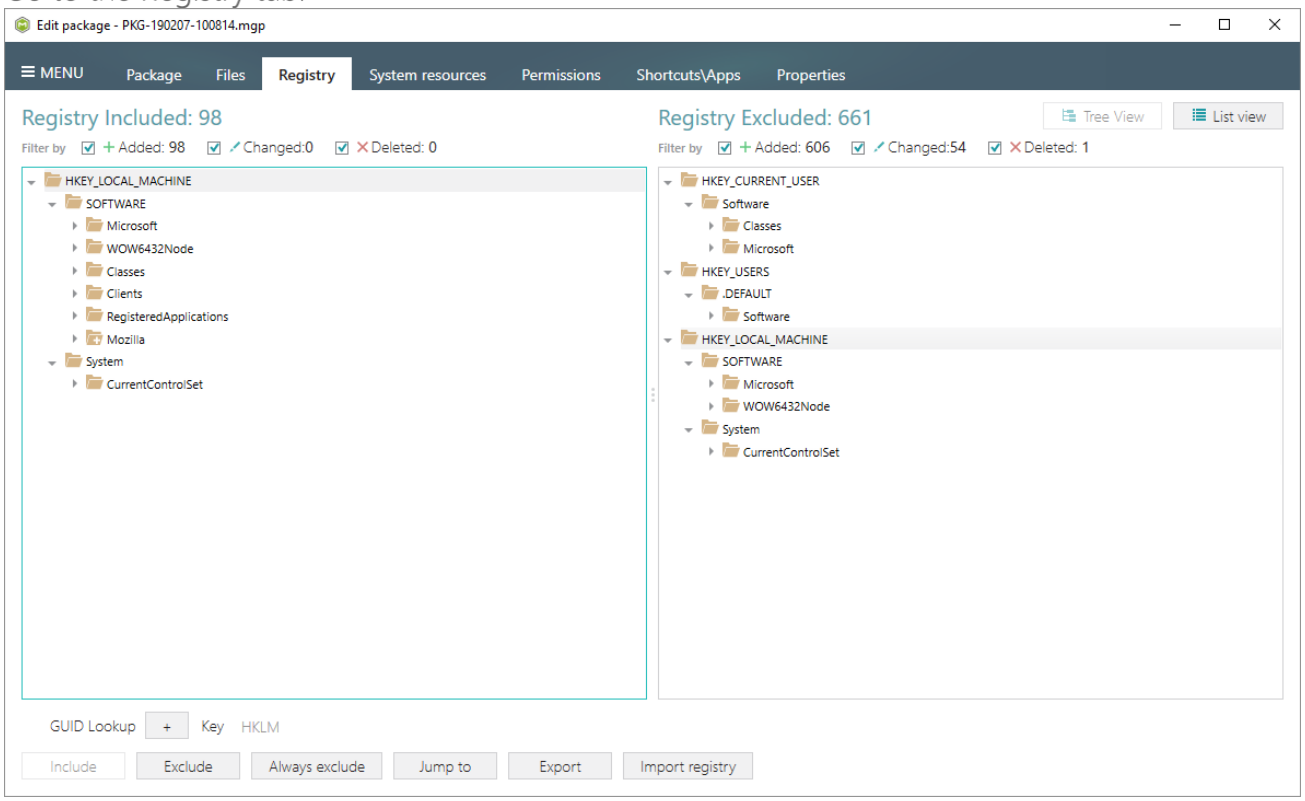
Choose a scenario that better suits your needs:

- Exclude Registry, described in section 3.8.3.1
  Exclude the unnecessary registry keys and values from the project.
- Include Excluded Registry, described in section 3.8.3.2
  Include the necessary registry keys and values, previously excluded from the project.
- Import Registry, described in section 3.8.3.3
  Import registry from the REG file to the project.
- Export Registry, described in section 3.8.3.4
  Export registry from the project to the REG file.

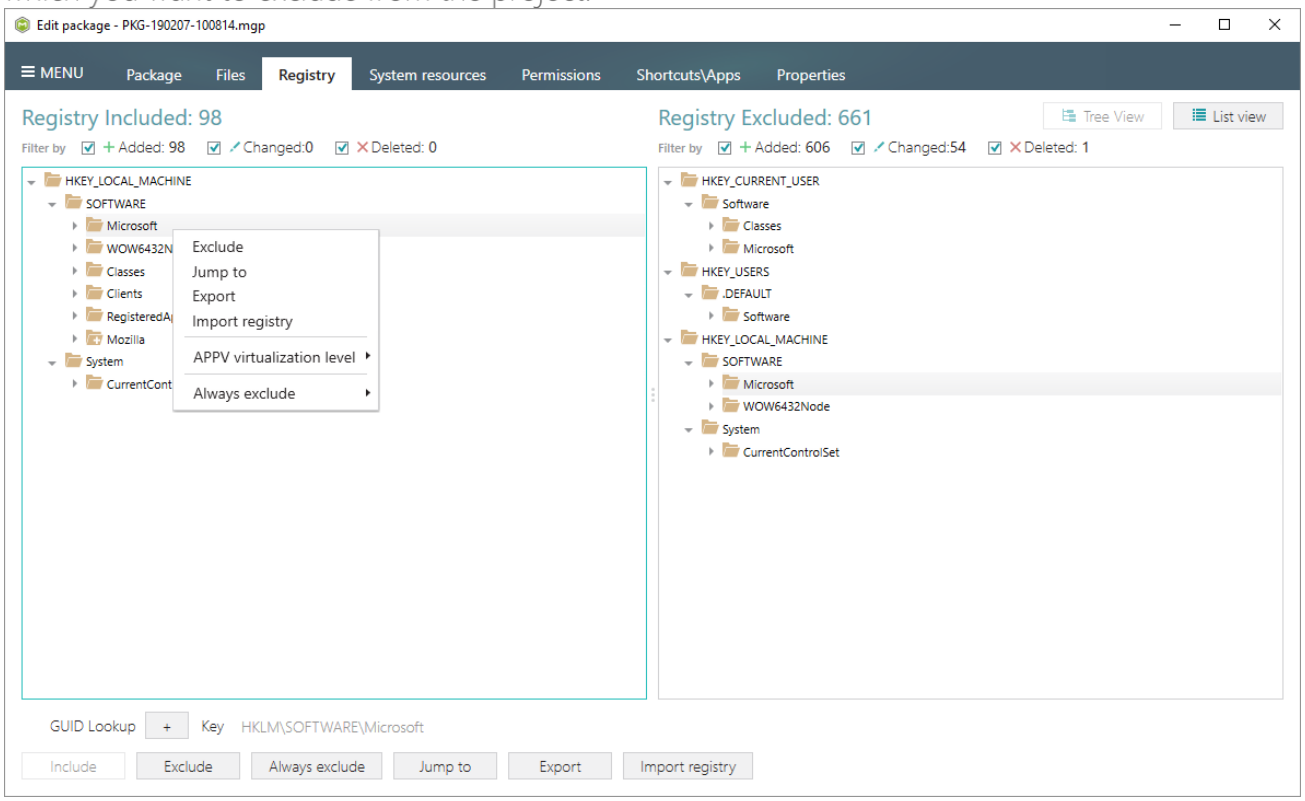
3.8.3.1 Exclude Registry

Exclude the unnecessary registry keys and values from the project.

- Go to the Registry tab.



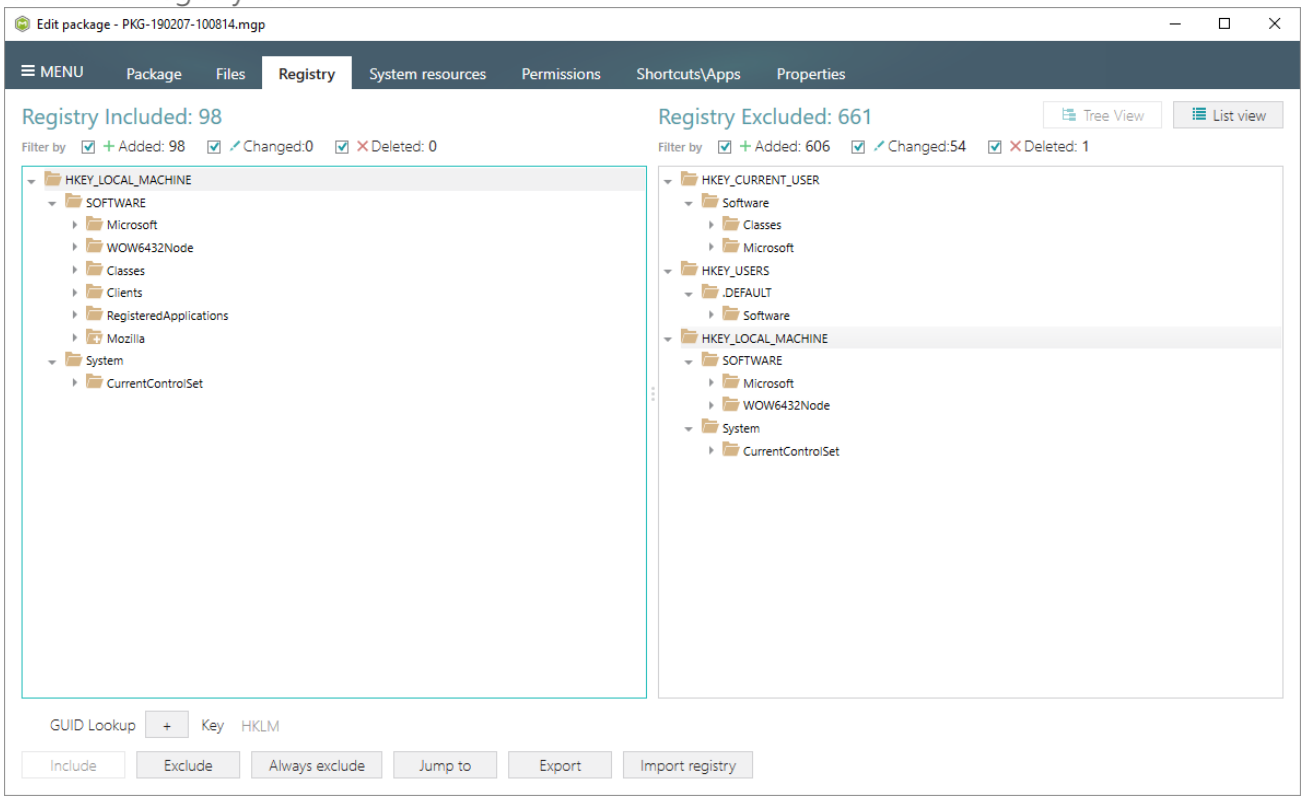
- Select **Exclude** from the context menu of an item, located in the left '**Registry Included**' pane, which you want to exclude from the project.



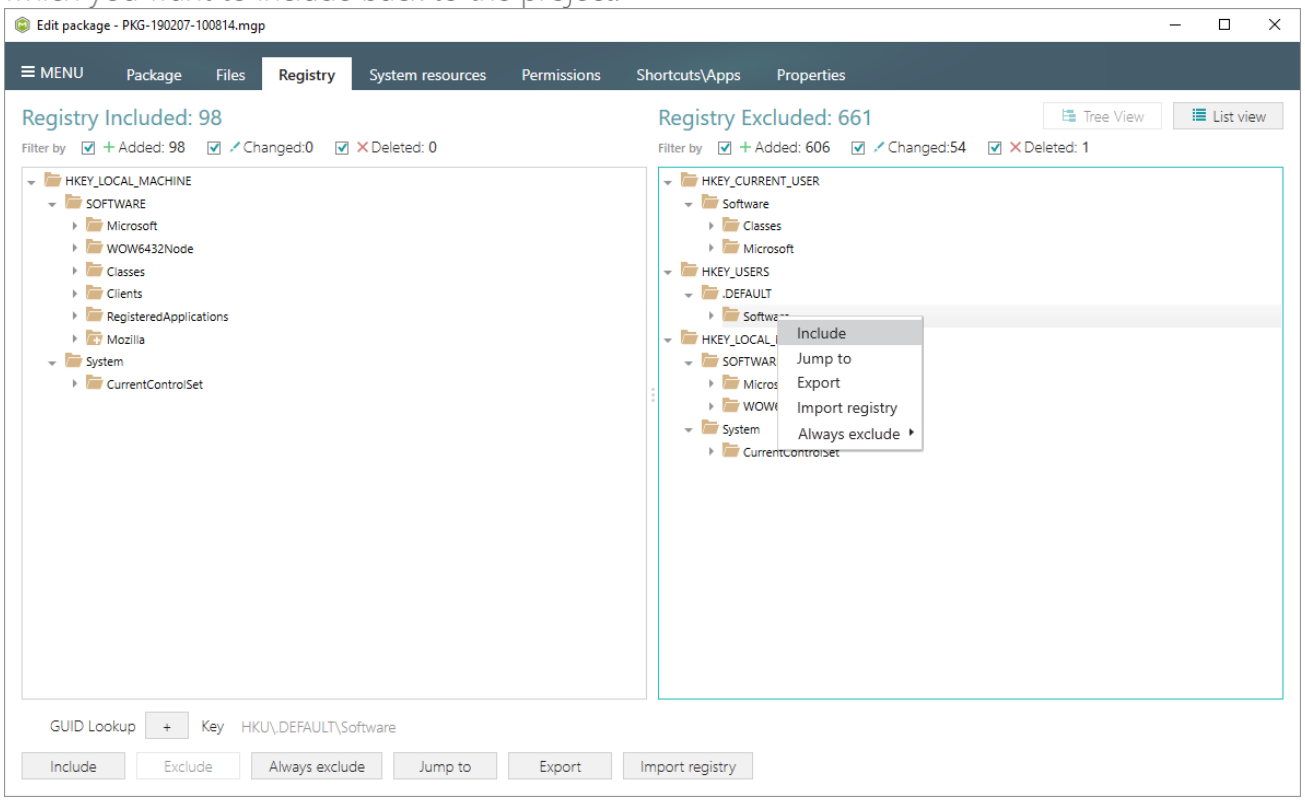
3.8.3.2 Include Excluded Registry

Include the necessary registry keys and values, previously excluded from the project.

- Go to the **Registry** tab.



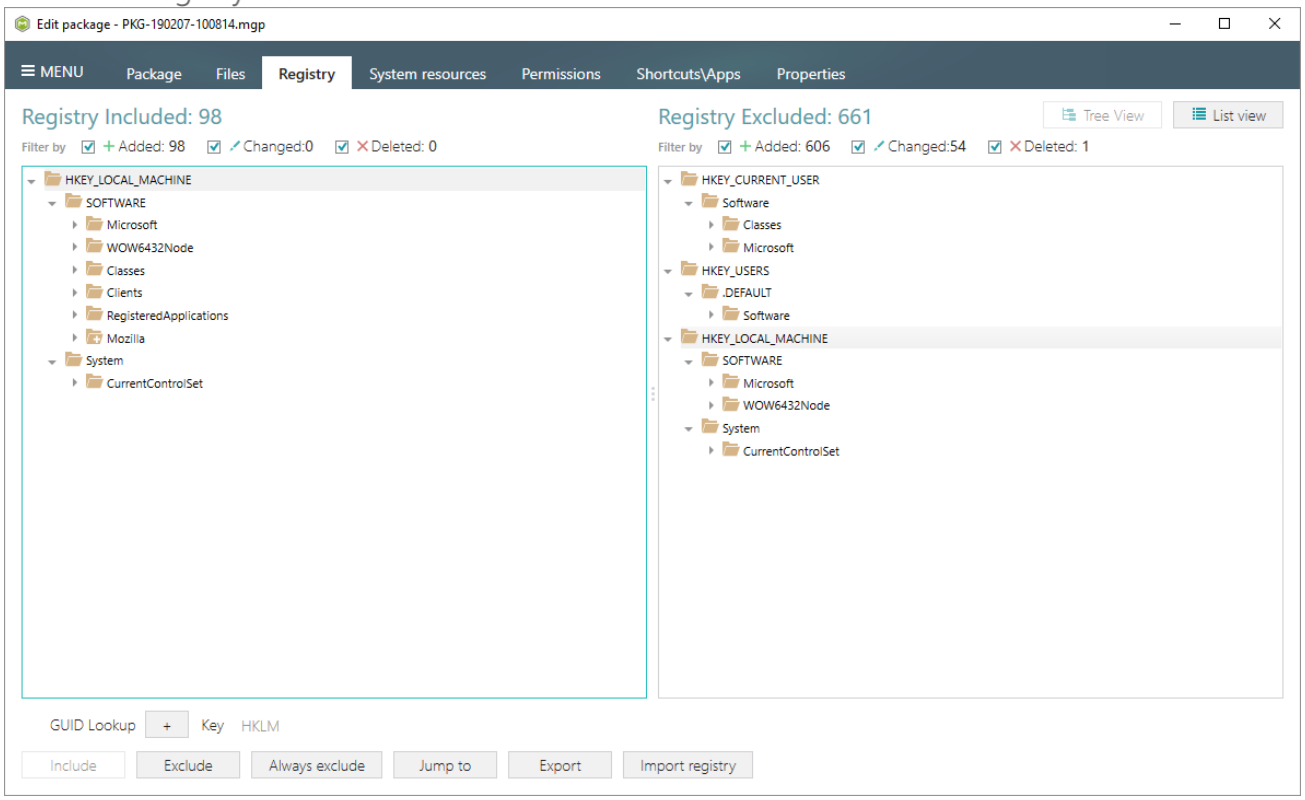
- Select **Include** from the context menu of an item, located in the right '**Registry Excluded**' pane, which you want to include back to the project.



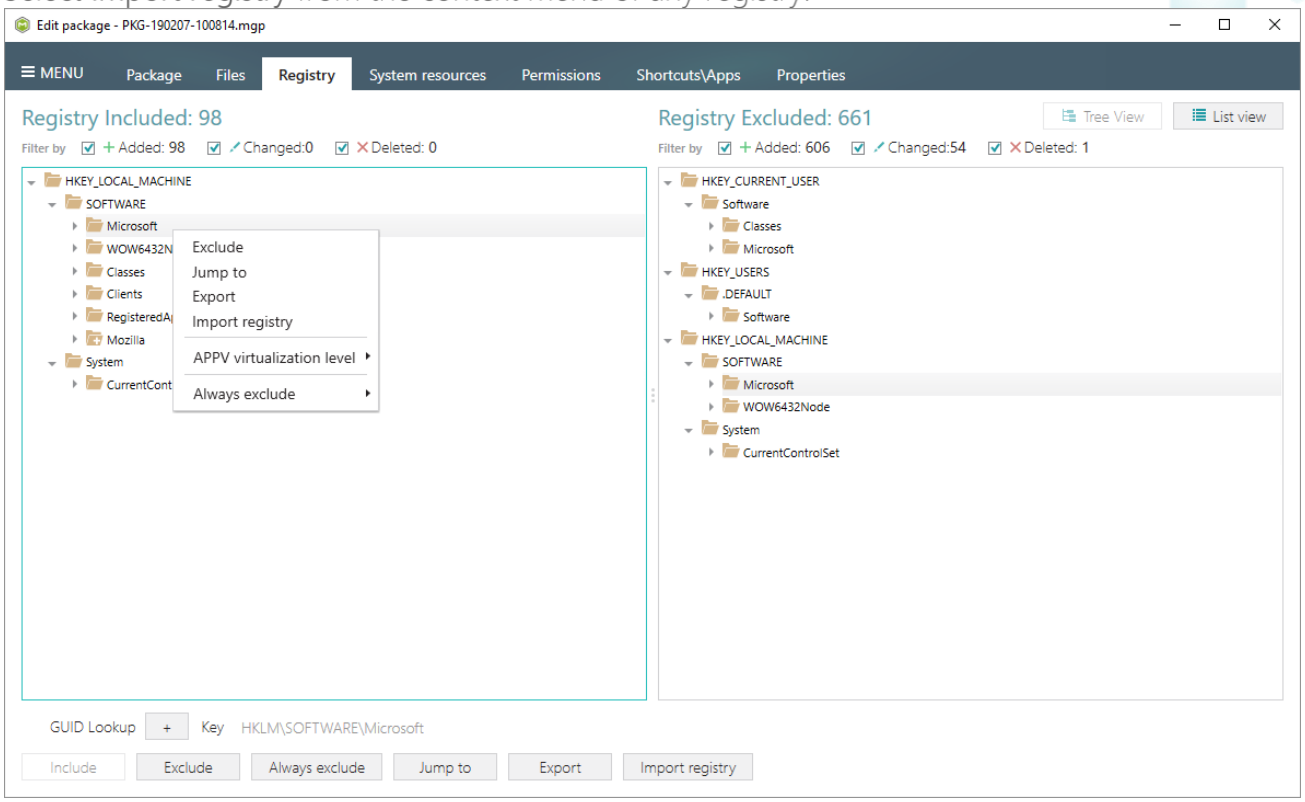
3.8.3.3 Import Registry

Import registry from the REG file to the project.

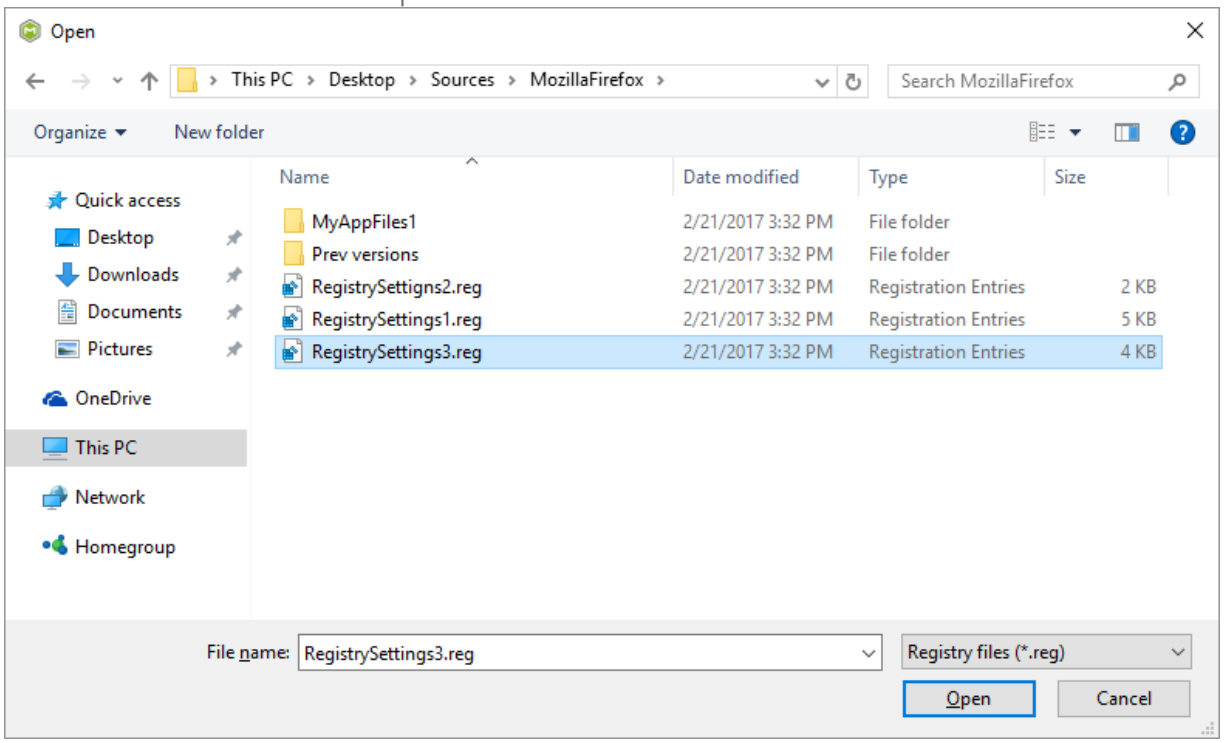
- Go to the **Registry** tab.



- Select **Import registry** from the context menu of any registry.



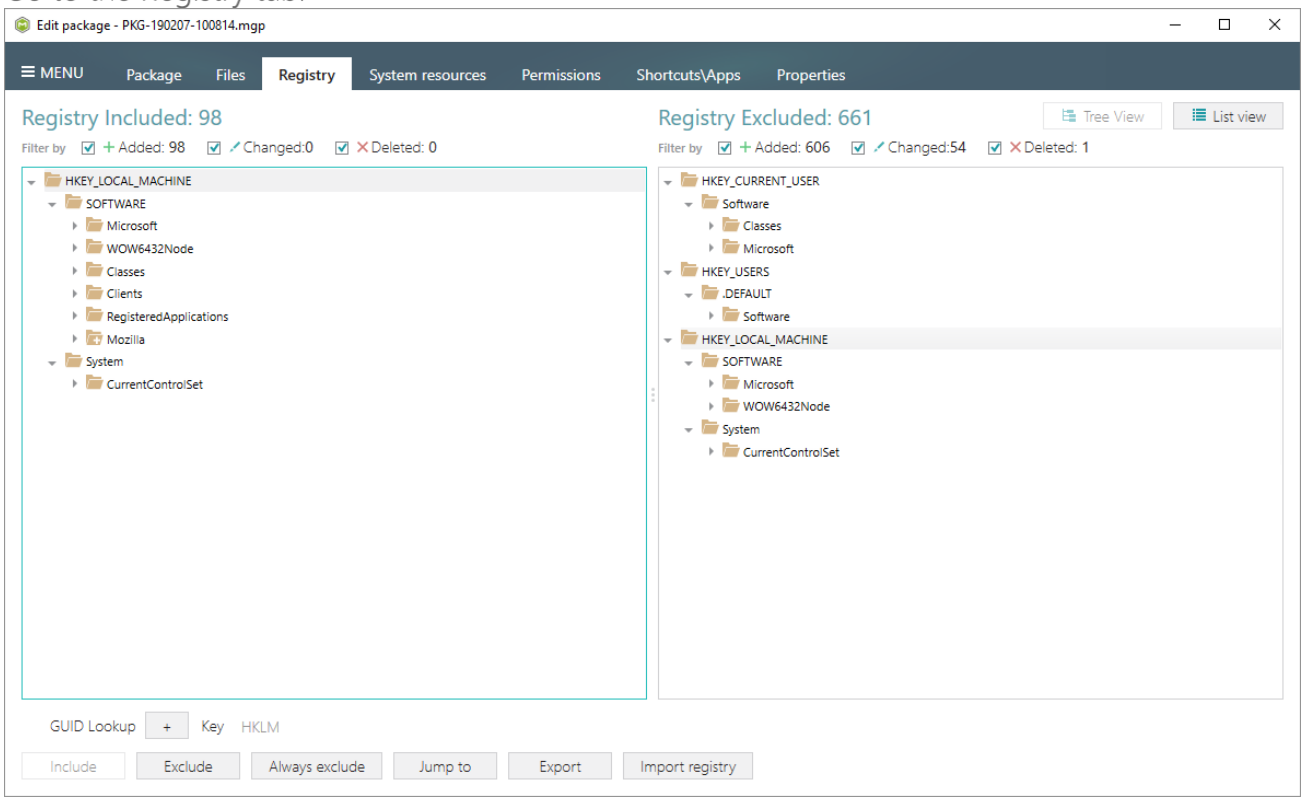
- Choose a REG file and click **Open**.



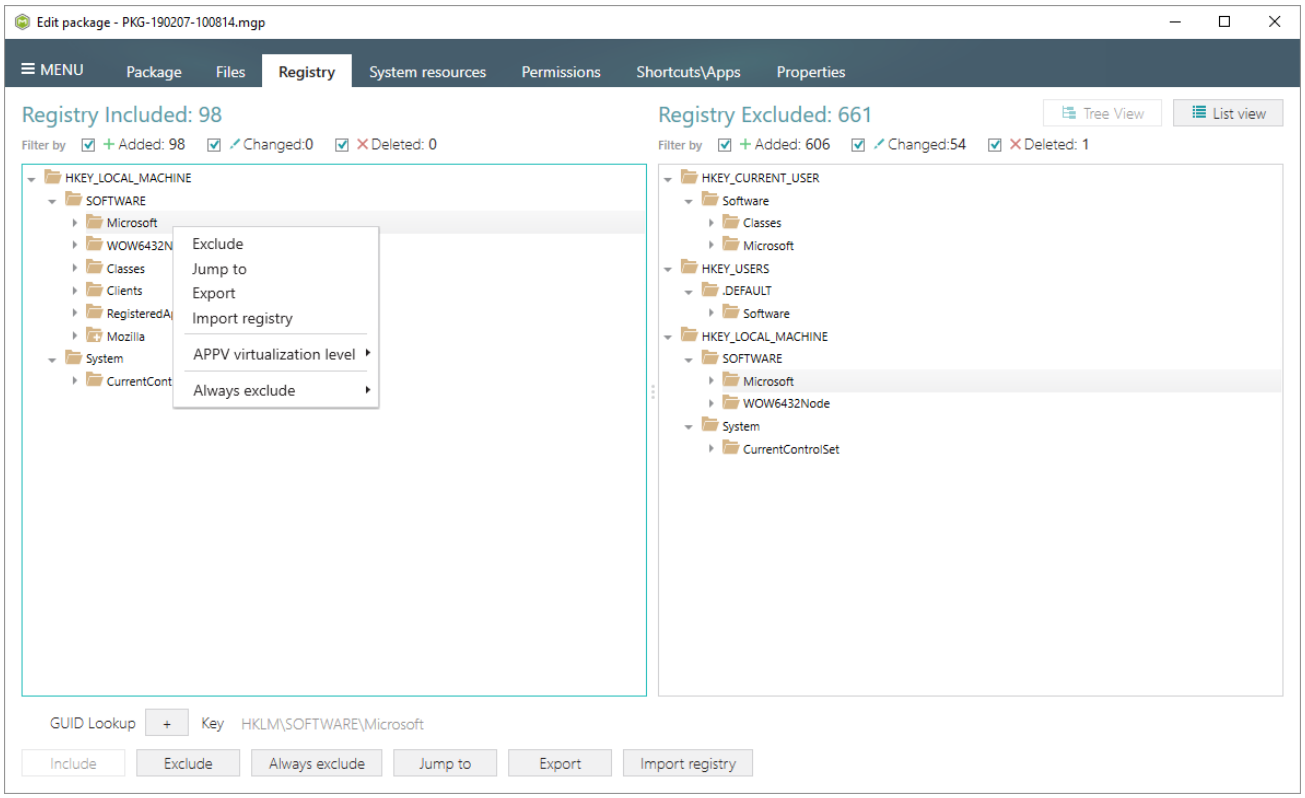
3.8.3.4 Export Registry

Export registry from the project to the REG file

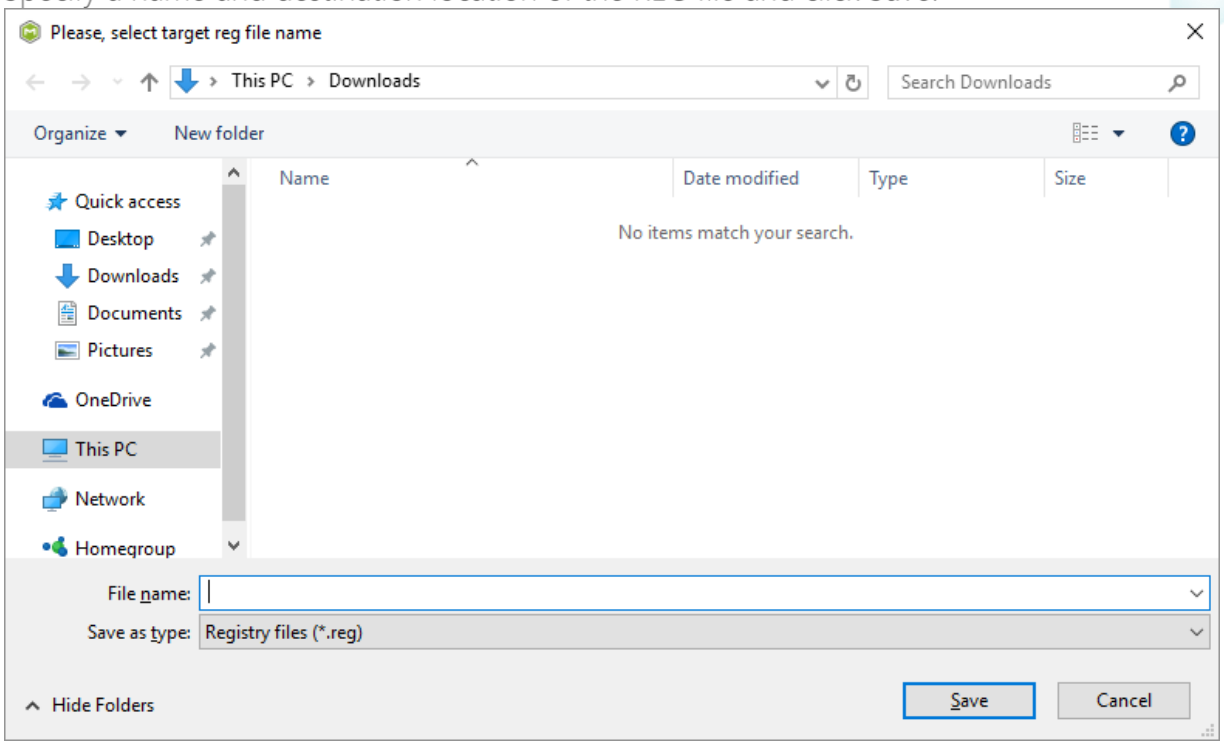
- Go to the **Registry** tab.



- Select **Export** from the context menu of a registry key or value, which you want to export to the REG file.



- Specify a name and destination location of the REG file and click Save.



3.8.4 Services

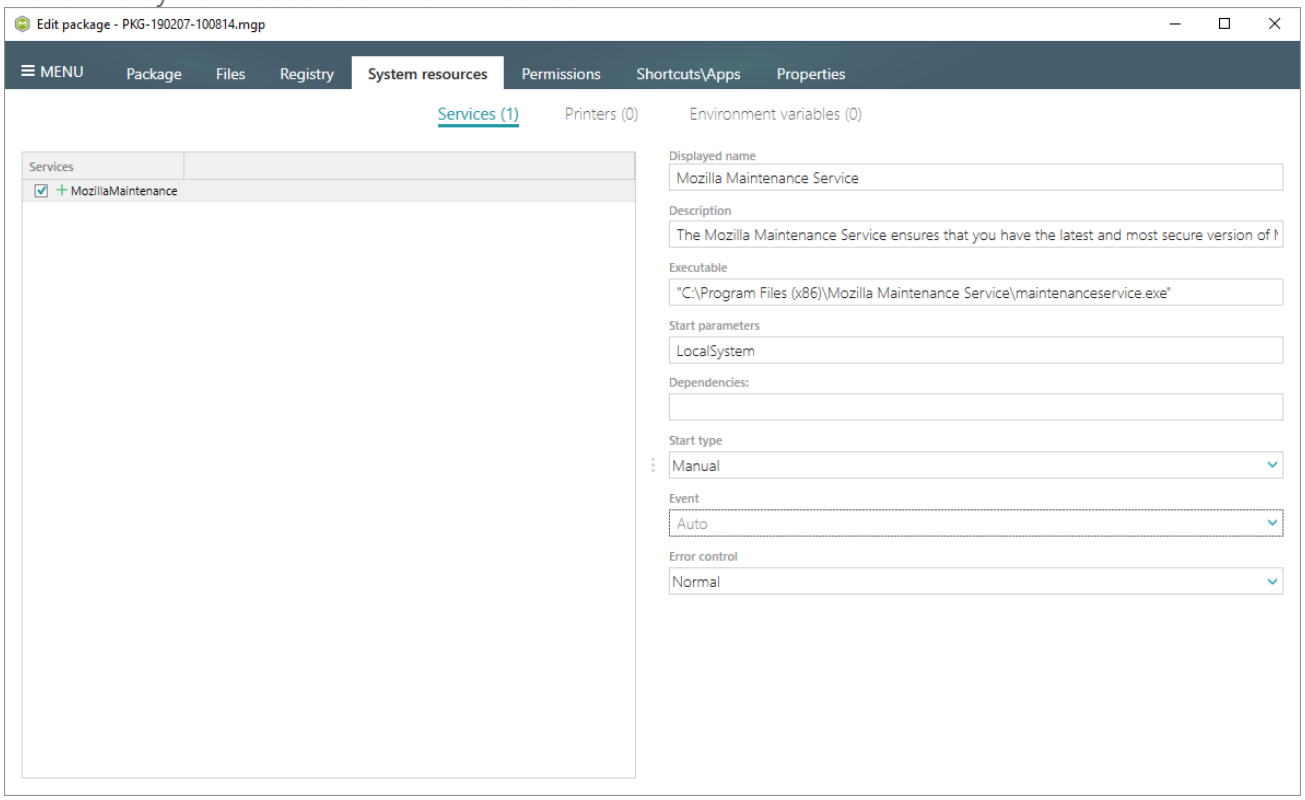
Choose a scenario that better suits your needs:

- Exclude Service, described in section 3.8.4.1
  Exclude the unnecessary services from the project.
- Include Excluded Service, described in section 3.8.4.2
  Include the necessary services, previously excluded from the project.
- Edit Service, described in section 3.8.4.3
  Edit the service details such as display name, description, start type, parameters, event and error control.

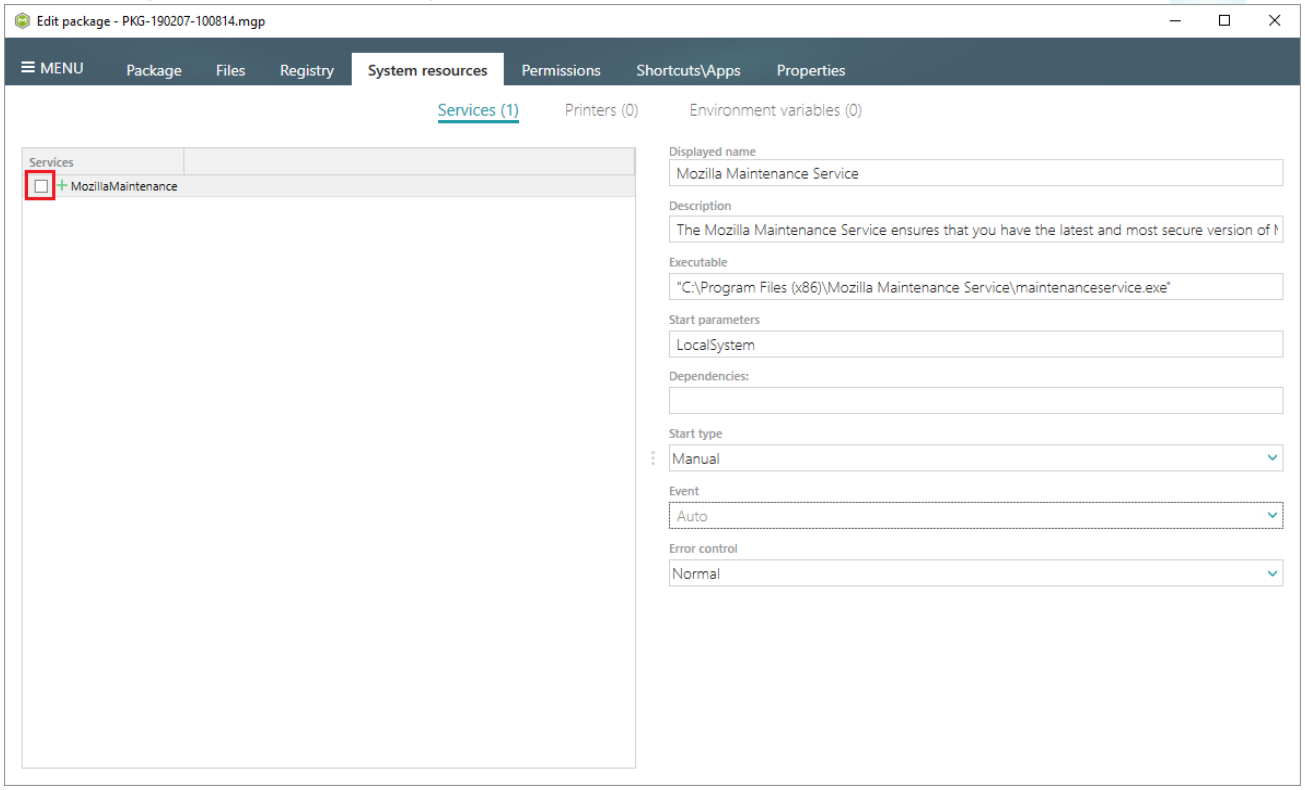
3.8.4.1 Exclude Service

Exclude the unnecessary services from the project.

- Go to the System resources -> Services tab.



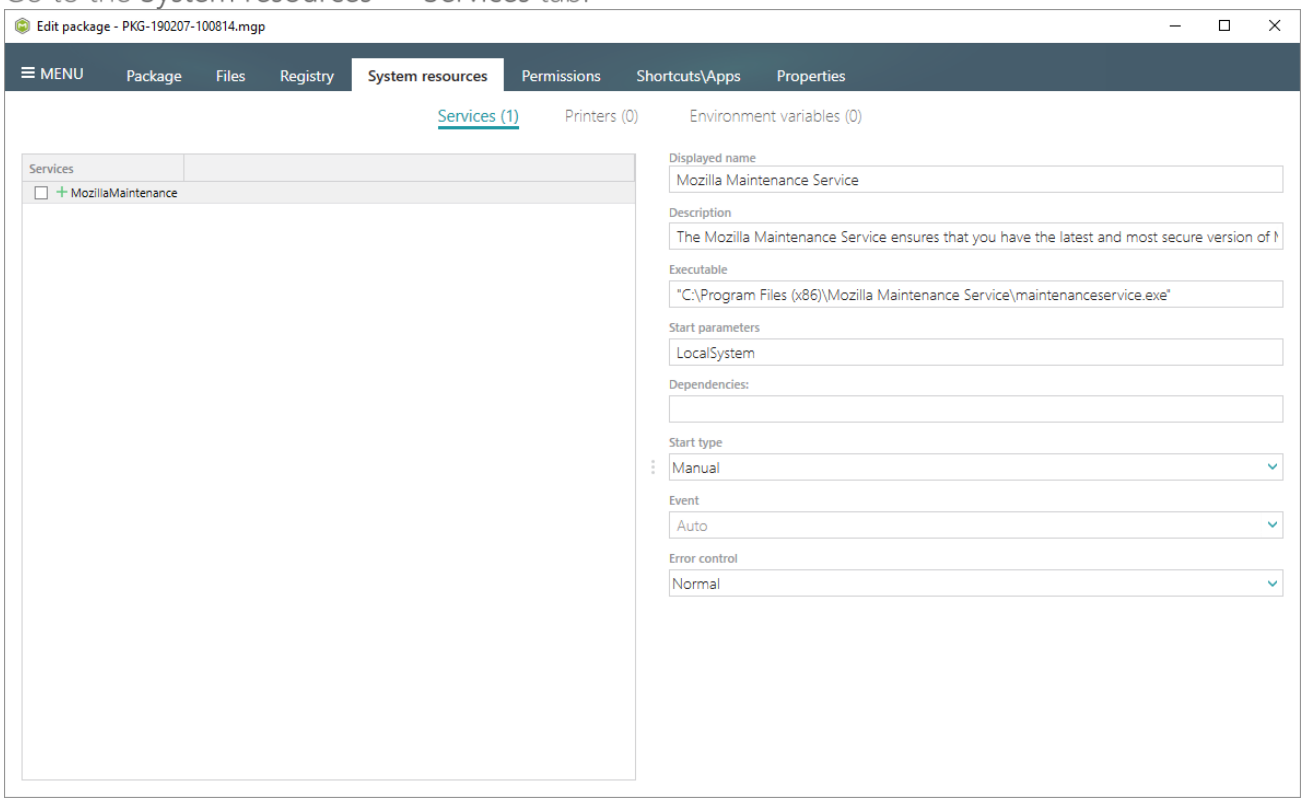
- To exclude, untick the checkbox, located before the service name.



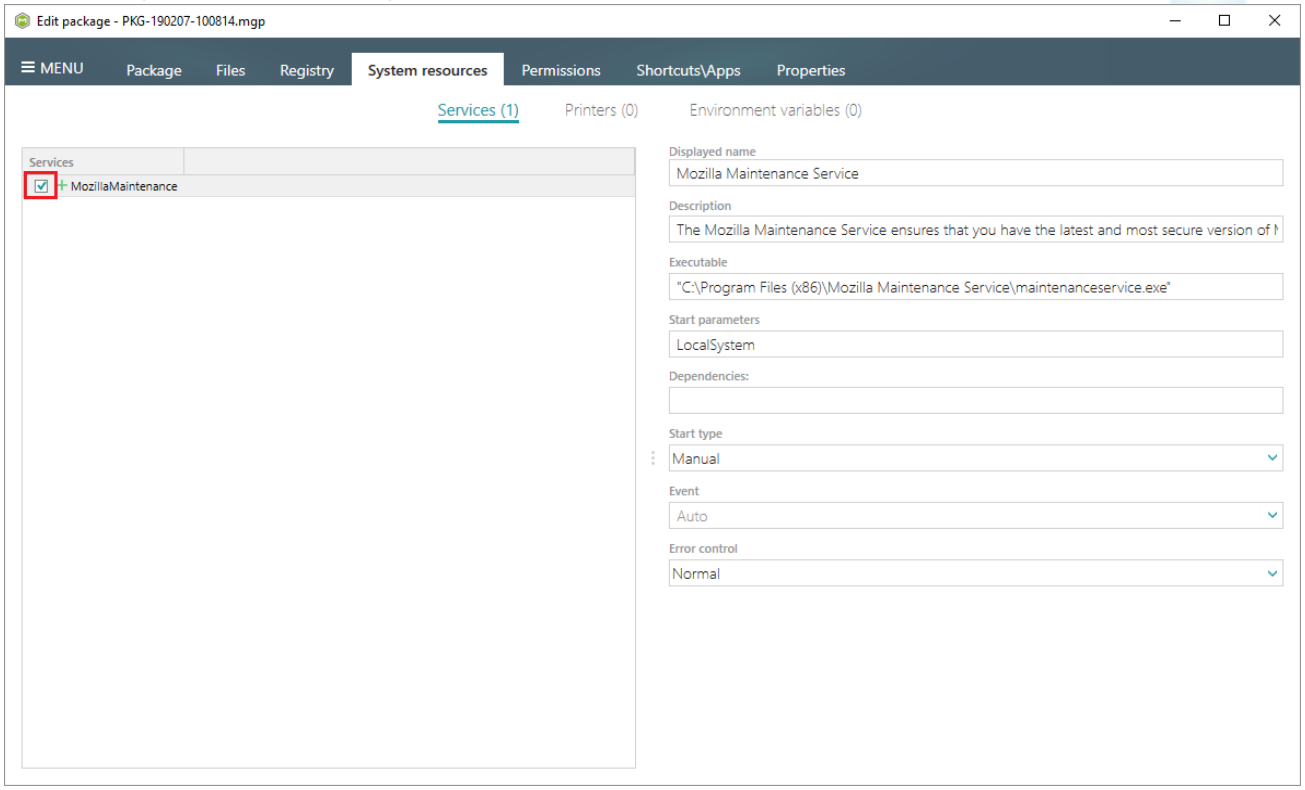
3.8.4.2 Include Excluded Service

Include the necessary services, previously excluded from the project.

- Go to the System resources -> Services tab.



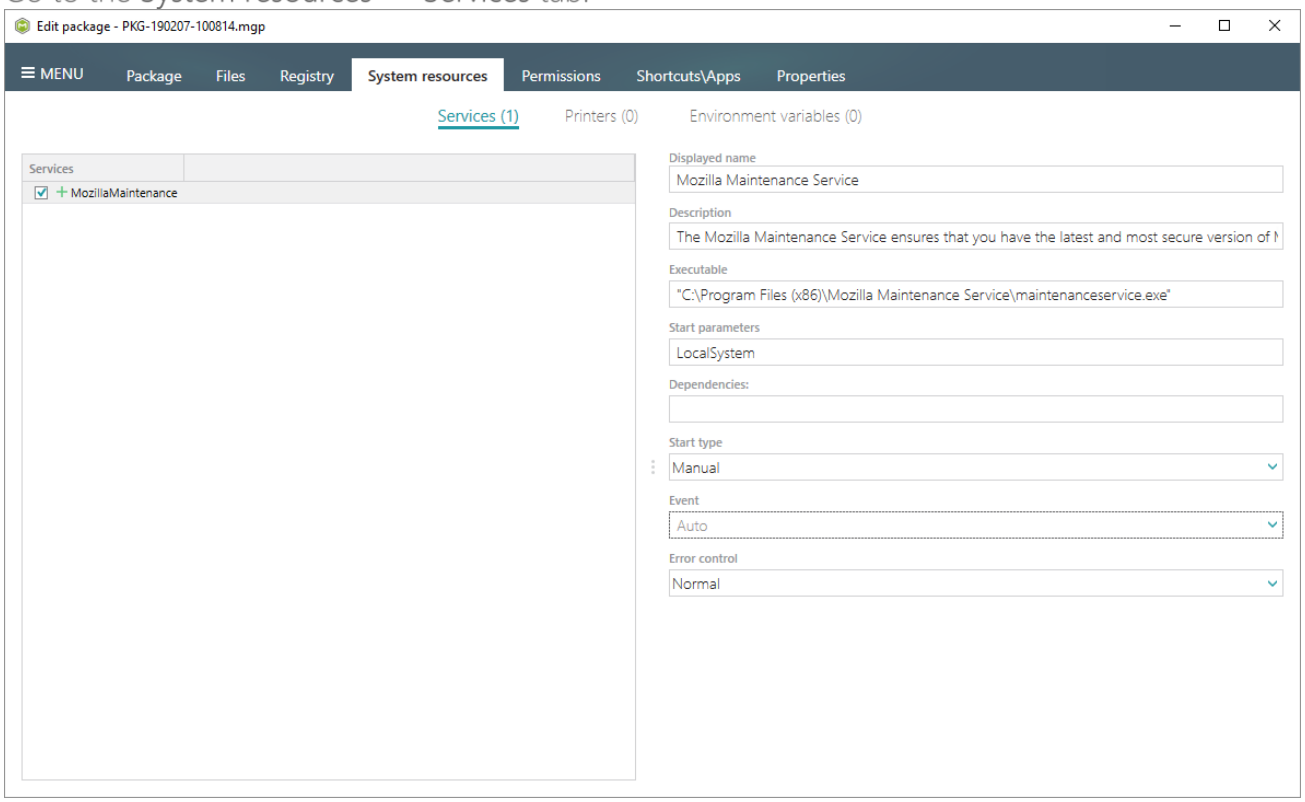
- To include, tick the checkbox, located before the service name.



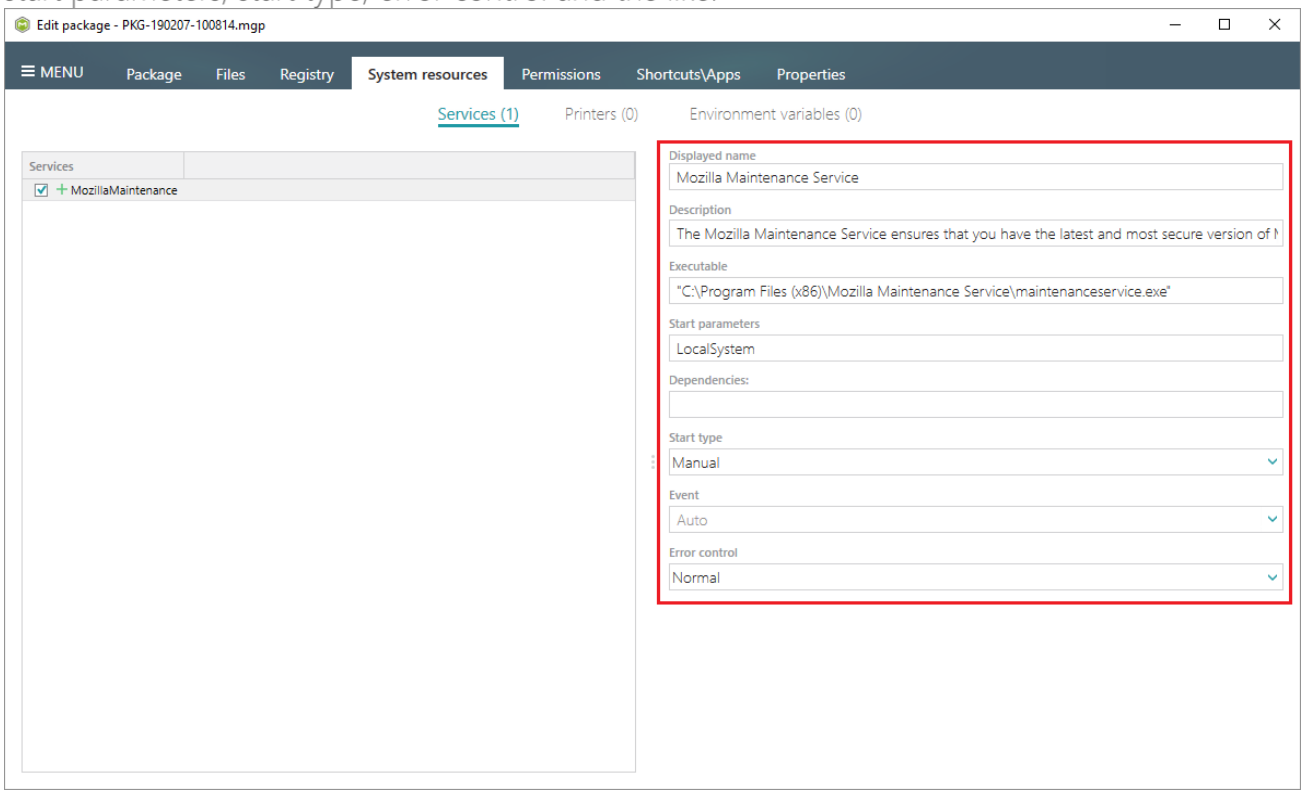
3.8.4.3 Edit Service

Edit the service details such as display name, description, start type, parameters, event and error control.

- Go to the System resources -> Services tab.



- Update the necessary values such as service display name, description, path to executable, start parameters, start type, error control and the like.



3.8.5 Environment Variables

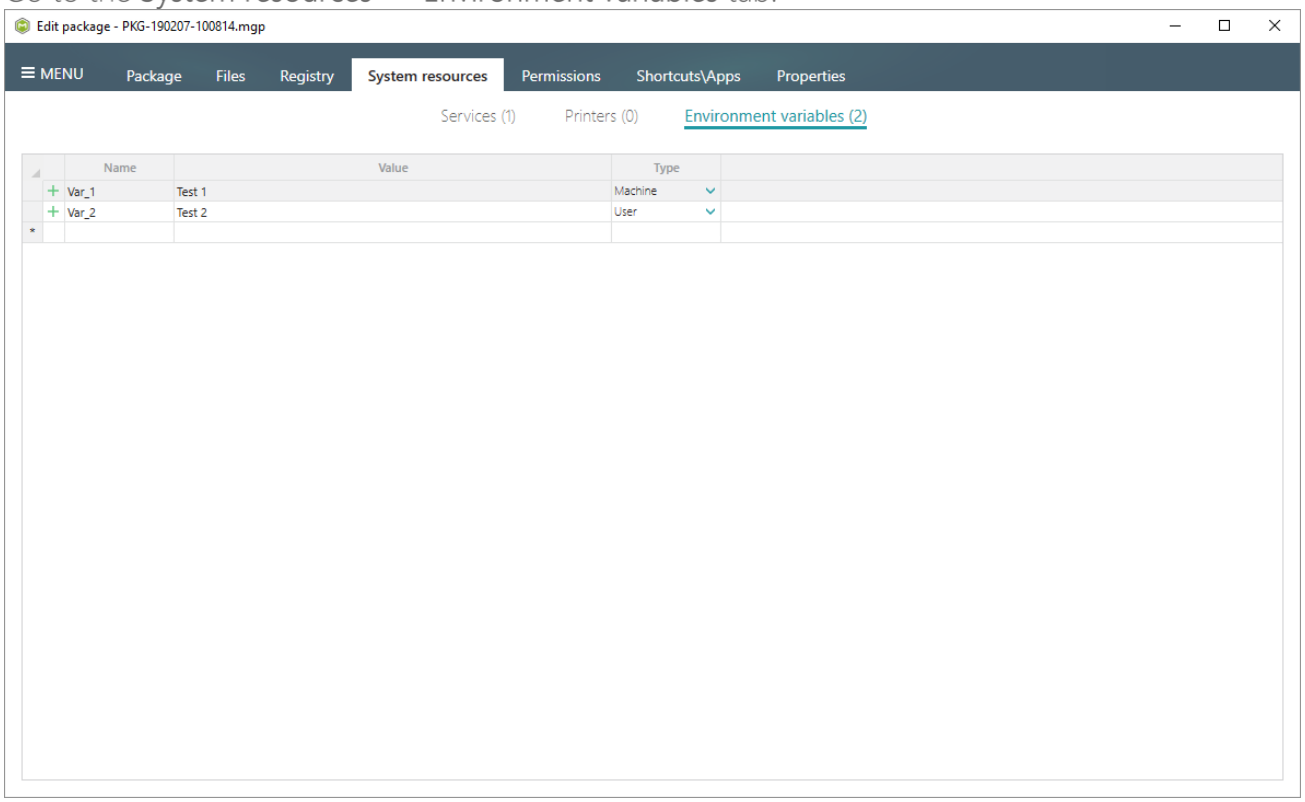
Choose a scenario that better suits your needs:

- Add Environment Variable, described in section 3.8.5.1
  Add a new environment variable to the project.
- Remove Environment Variable, described in section 3.8.5.2
  Remove an environment variable from the project.

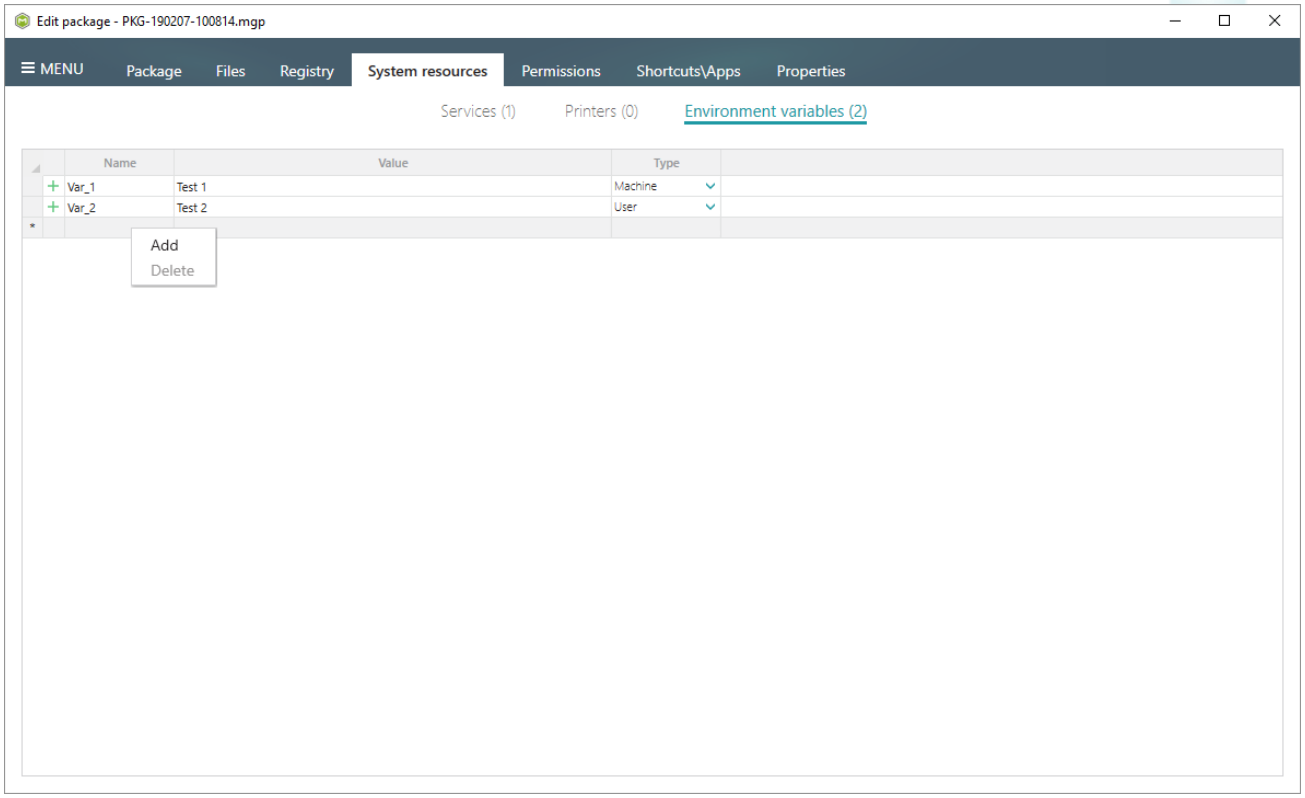
3.8.5.1 Add Environment Variable

Add a new environment variable to the project.

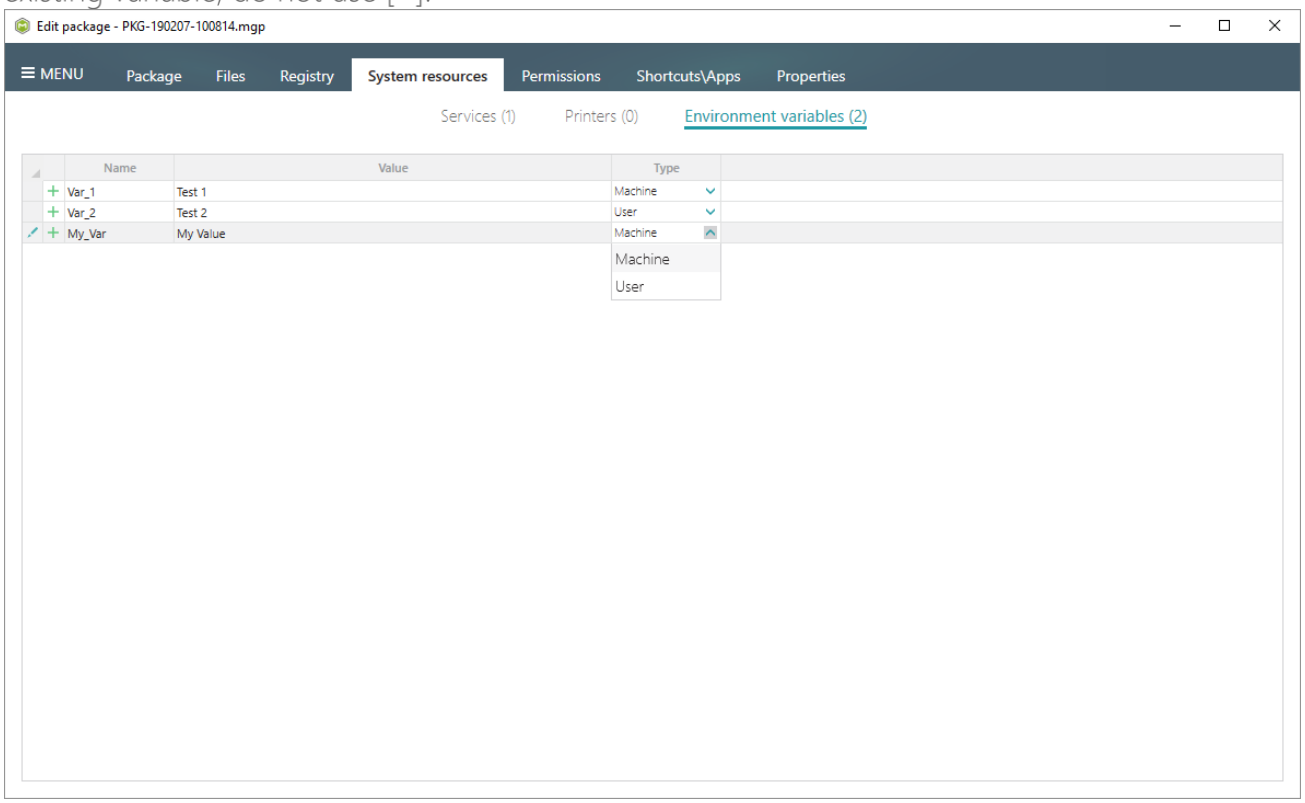
- Go to the System resources -> Environment variables tab.



- Select **Add** from the context menu.



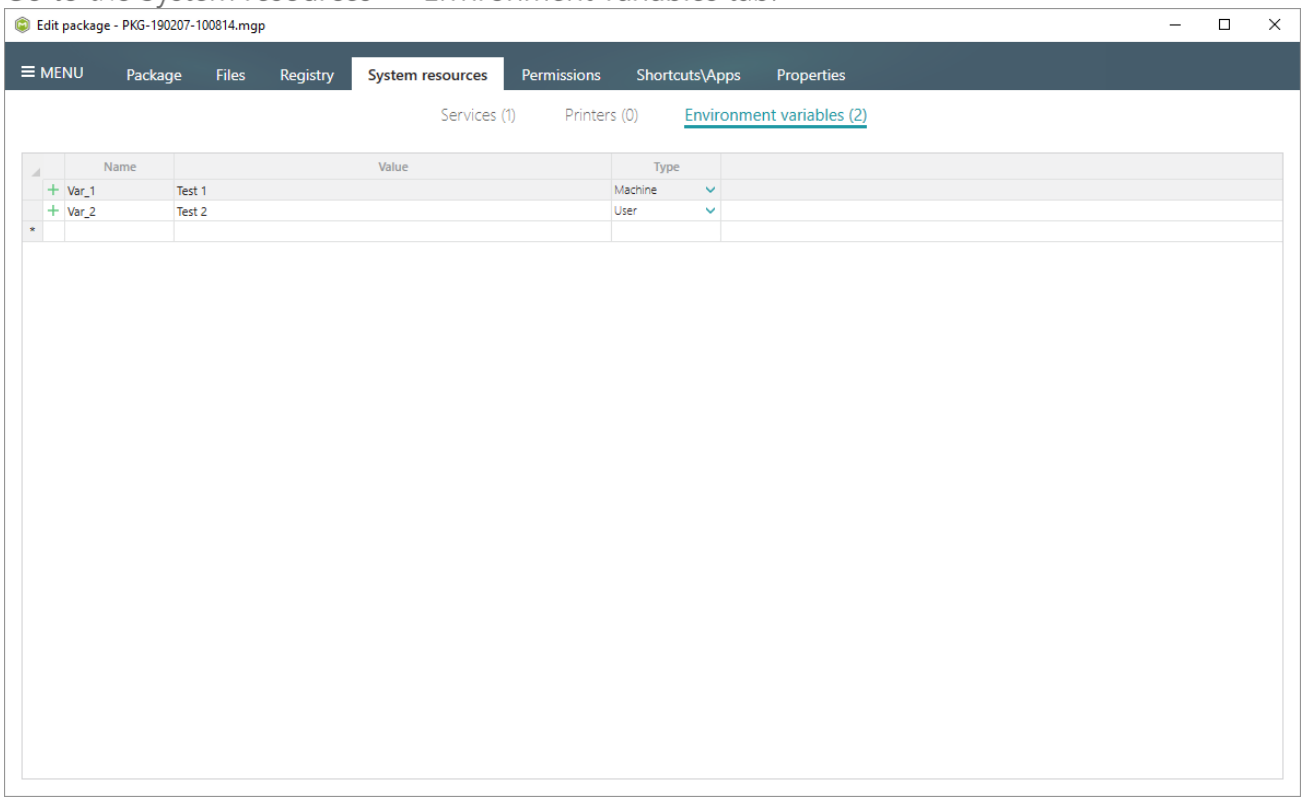
- Enter a variable name, value and choose its type: per-machine or per-user.
  NOTE To append the entered value to the end of an existing variable, put the "**[~];**" string before your value (e.g. **[~];Your value**). To prefix the entered value to the front of an existing variable, put the "**;[~]**" string after your value (e.g. **Your value;[~]**). To replace a value of the existing variable, do not use [~].



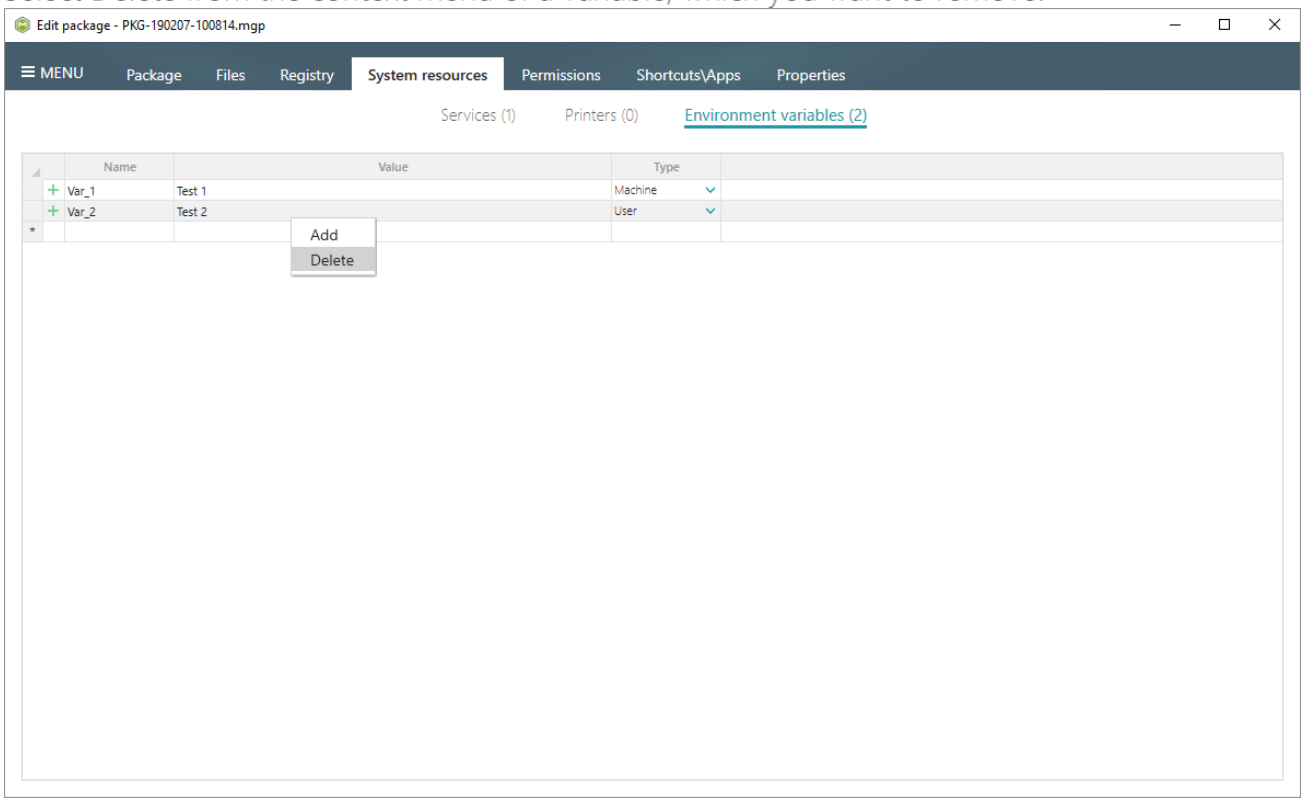
3.8.5.2 Remove Environment Variable

Remove an environment variable from the project.

- Go to the System resources -> Environment variables tab.



- Select Delete from the context menu of a variable, which you want to remove.



3.8.6 MSI Settings

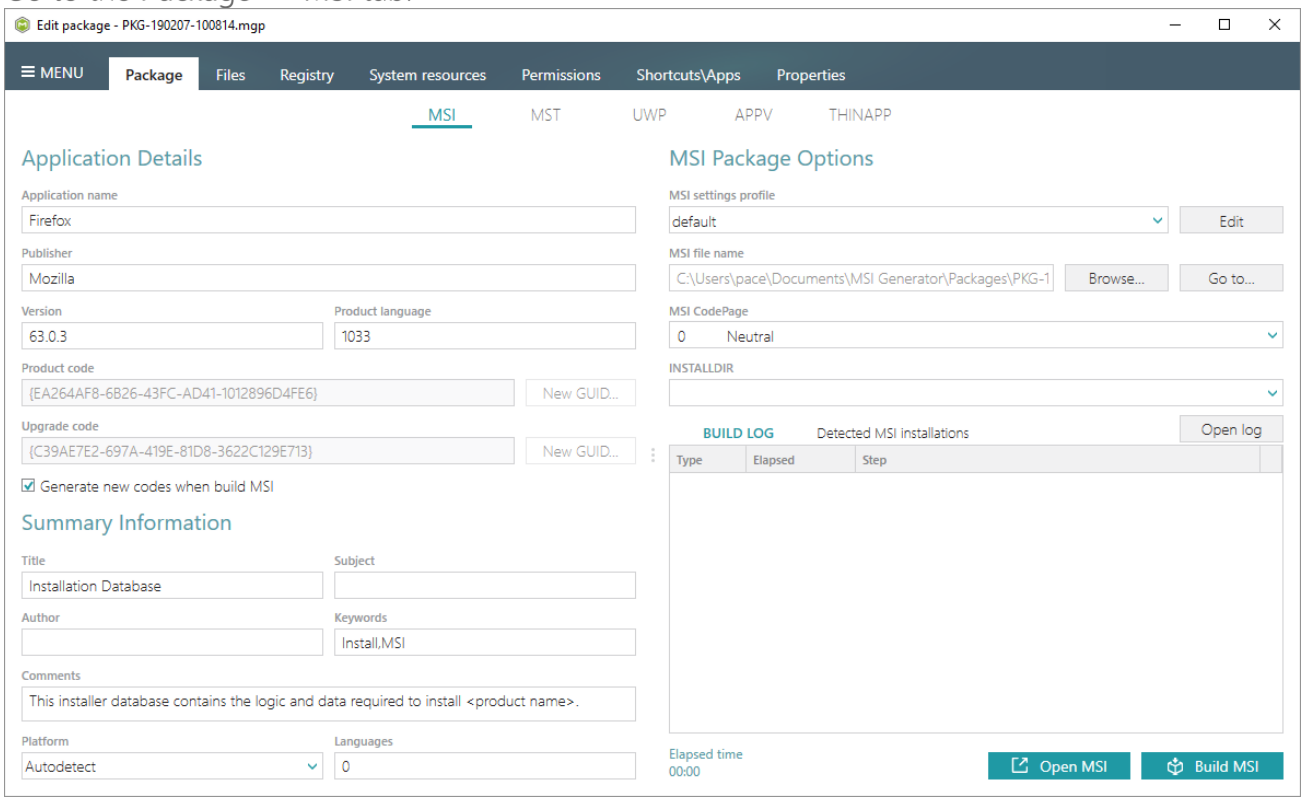
Choose a scenario that better suits your needs:

- Set Application Details, described in section 3.8.6.1
  Control how the package will be displayed in the App list (Add & Remove Programs).
- Set INSTALLDIR, described in section 3.8.6.2
  Set a folder of the project as the INSTALLDIR.
- Manage Shortcuts, described in section 3.8.6.3
  Exclude/include, import new ones and edit existing shortcuts.
- Control File/Folder Permissions, described in section 3.8.6.4
  Control which of captured permission changes for files and folders will be included into the package.
- Control Registry Permissions, described in section 3.8.6.5
  Control which of captured permission changes for files and folders will be included into the package.
- Select Compression Type, described in section 3.8.6.6
  Select the compression type for the package files.
- Use Advertising Tables, described in section 3.8.6.7
  Select which of MSI advertizing tables will be used for storing the capturing registry.
- Add Active Setup, described in section 3.8.6.8
  Intergrate the Active Setup to the MSI package automatically, if it contains per-user data.
- Set Summary Information, described in section 3.8.6.9
  Set the default Summary Information values, which will be saved to each created MSI package.

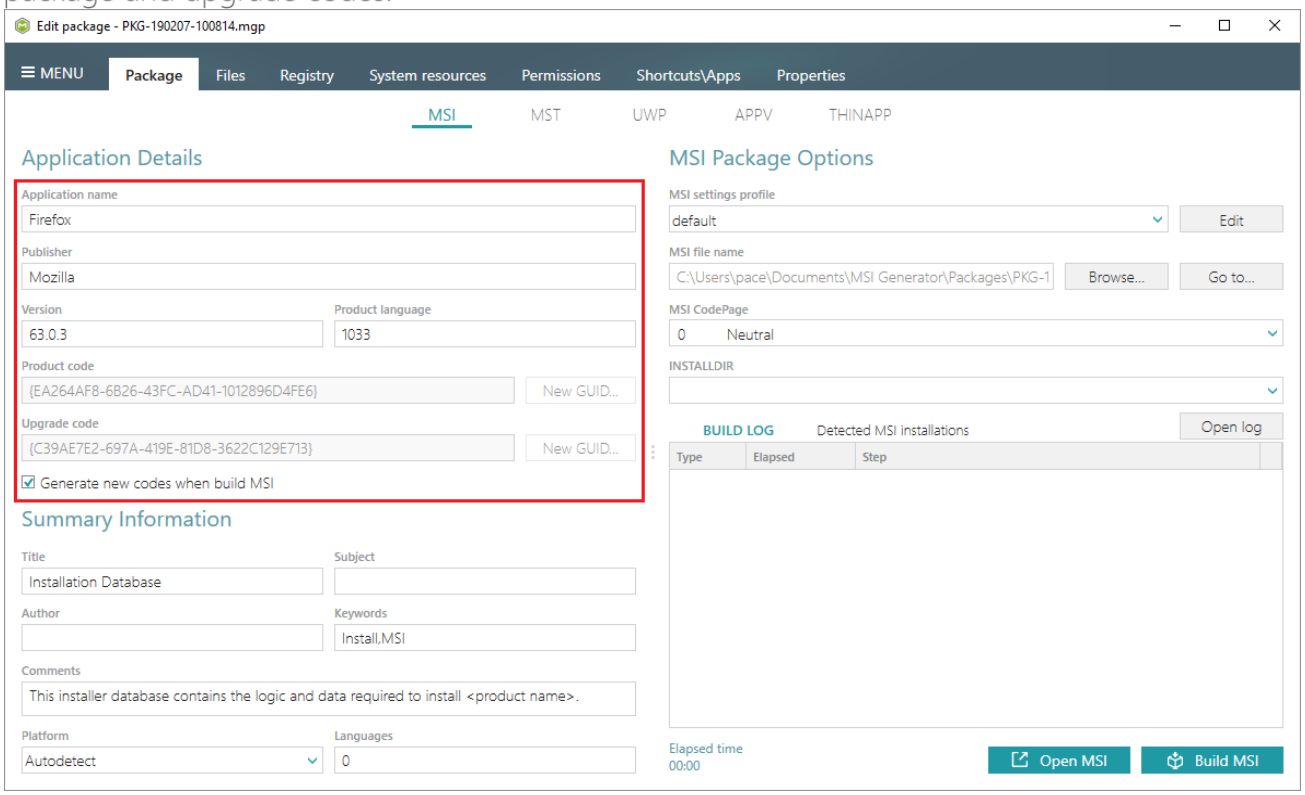
3.8.6.1 Set Application Details

Control how the package will be displayed in the App list (Add & Remove Programs).

- Go to the Package -> MSI tab.



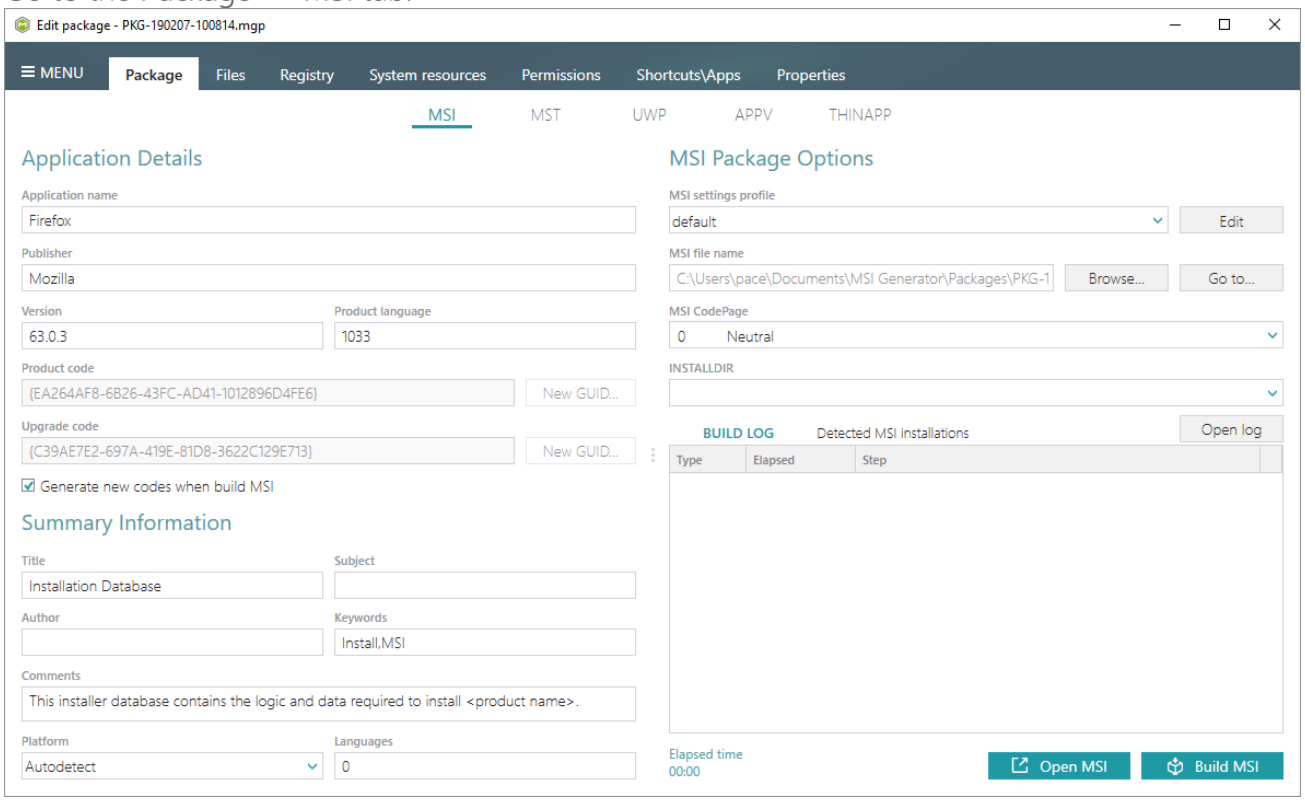
- Set application details such as application name (product name), version, publisher, language, package and upgrade codes.



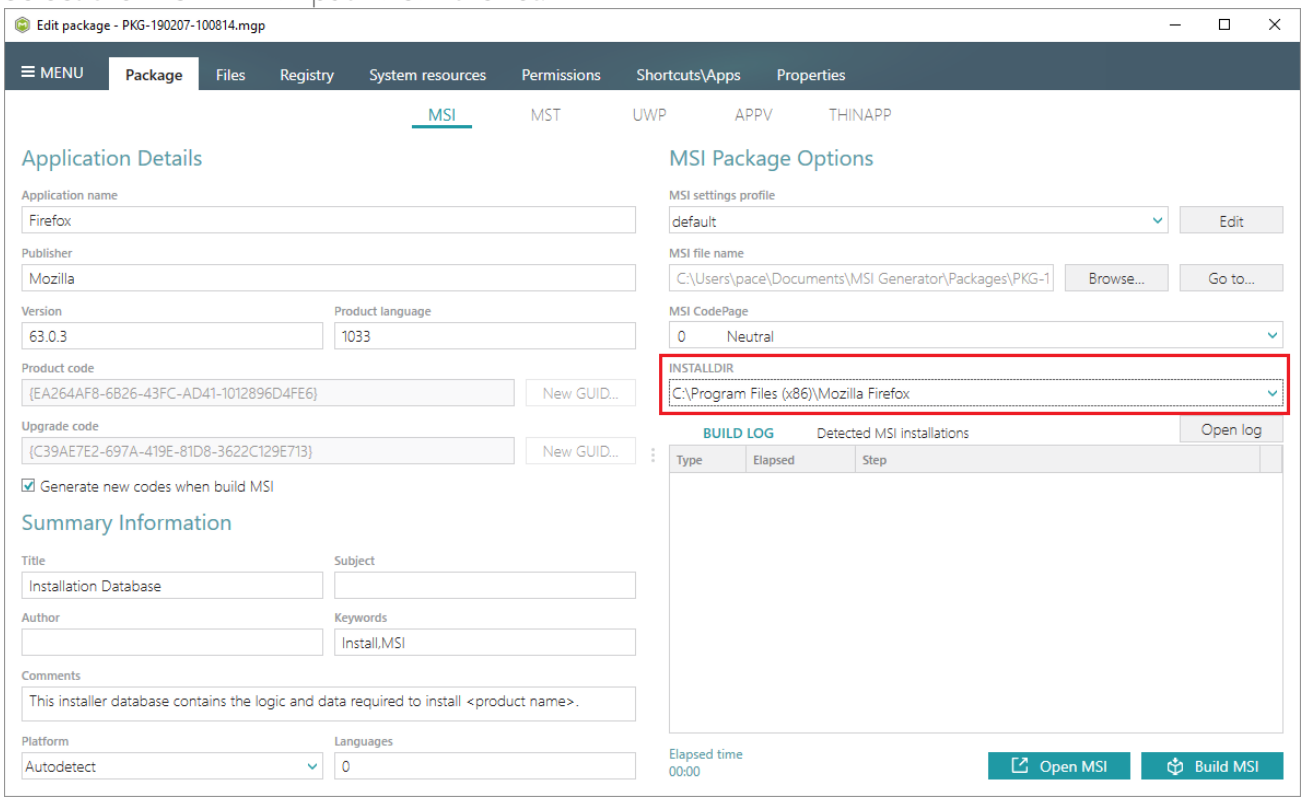
3.8.6.2 Set INSTALLDIR

Set a folder of the project as the INSTALLDIR. This means that an identifier of the selected folder will be set to the 'INSTALLDIR' value. Usually, the INSTALLDIR property represents the main product installation folder, and you can re-define this path via the command line or a Custom Action.

- Go to the Package -> MSI tab.



- Select the INSTALLDIR path from the list.



3.8.6.3 Manage Shortcuts

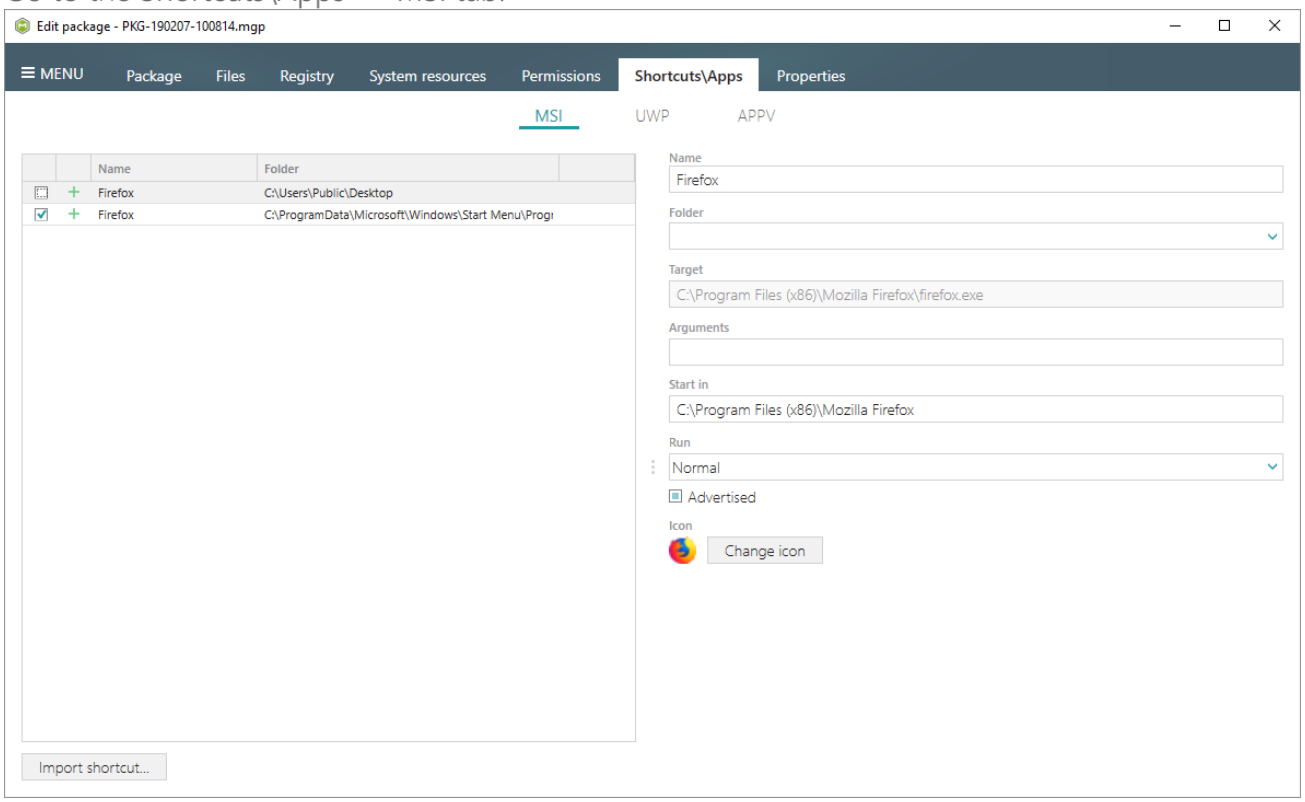
Choose a scenario that better suits your needs:

- Exclude Shortcut, described in section 3.8.6.3.1
  Exclude the unnecessary shortcuts from the project
- Include Excluded Shortcut, described in section 3.8.6.3.2
  Include the necessary shortcuts, previously excluded from the project.
- Edit Shortcut, described in section 3.8.6.3.3
  Edit the shortcut detail such as name, destination folder, shortcut type, icon, etc.
- Import Shortcut, described in section 3.8.6.3.4
  Import shortcuts from the file system to the project.

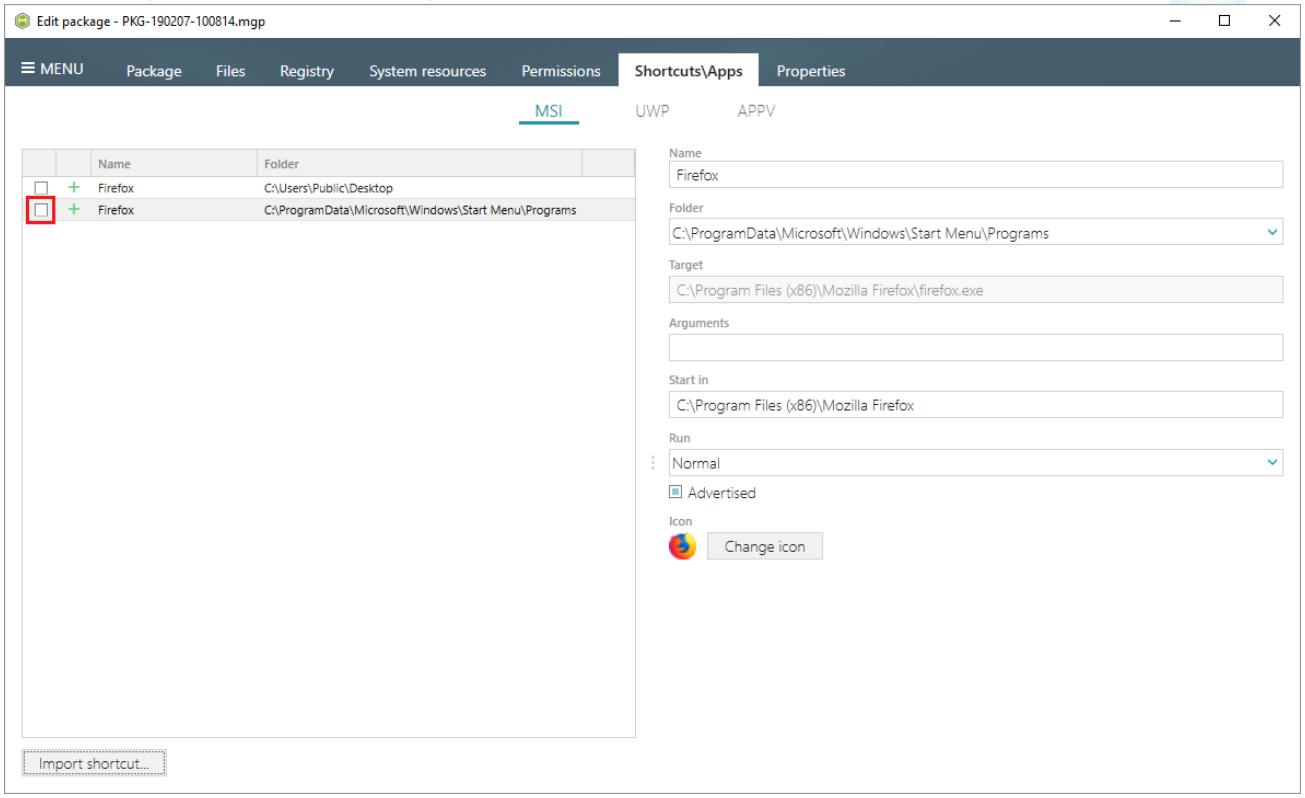
3.8.6.3.1 Exclude Shortcut

Exclude the unnecessary shortcuts from the project.

- Go to the Shortcuts\Apps -> MSI tab.



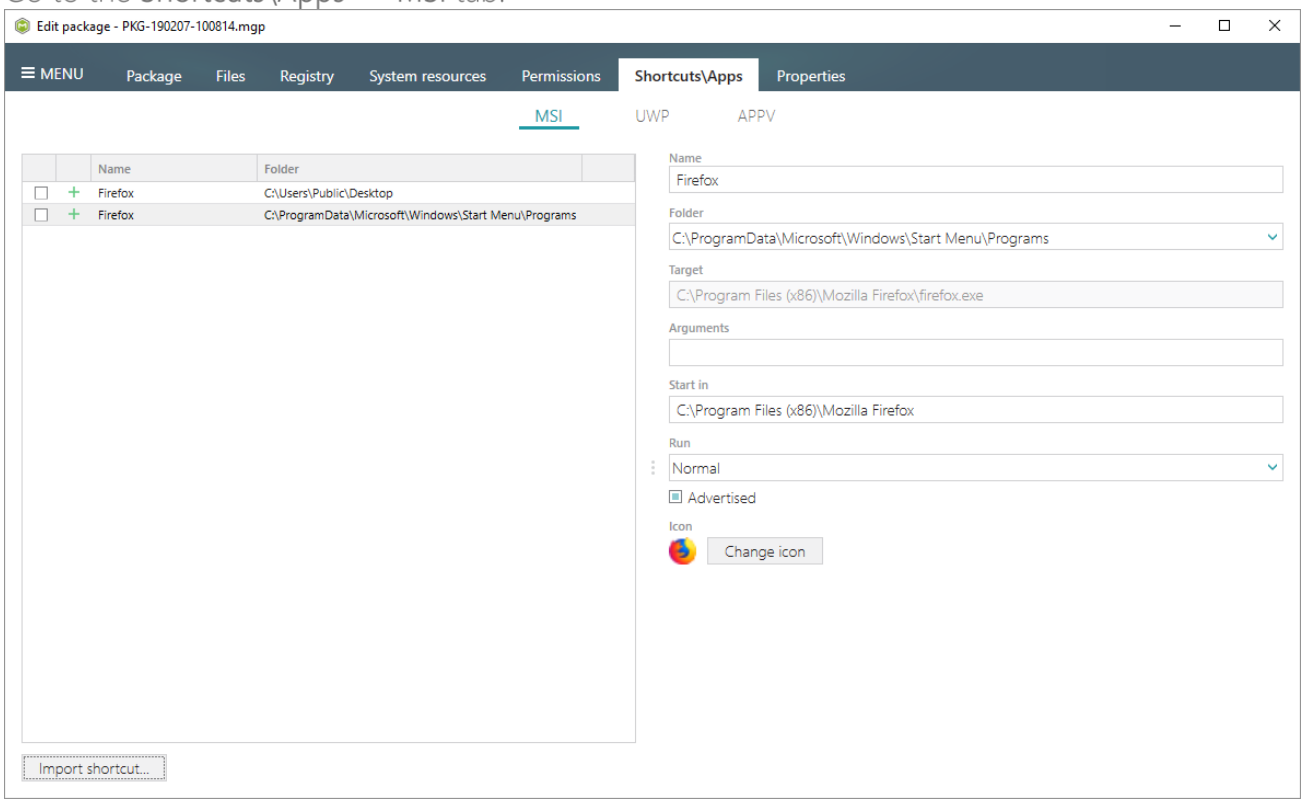
- To exclude, untick the checkbox, located before the shortcut name.



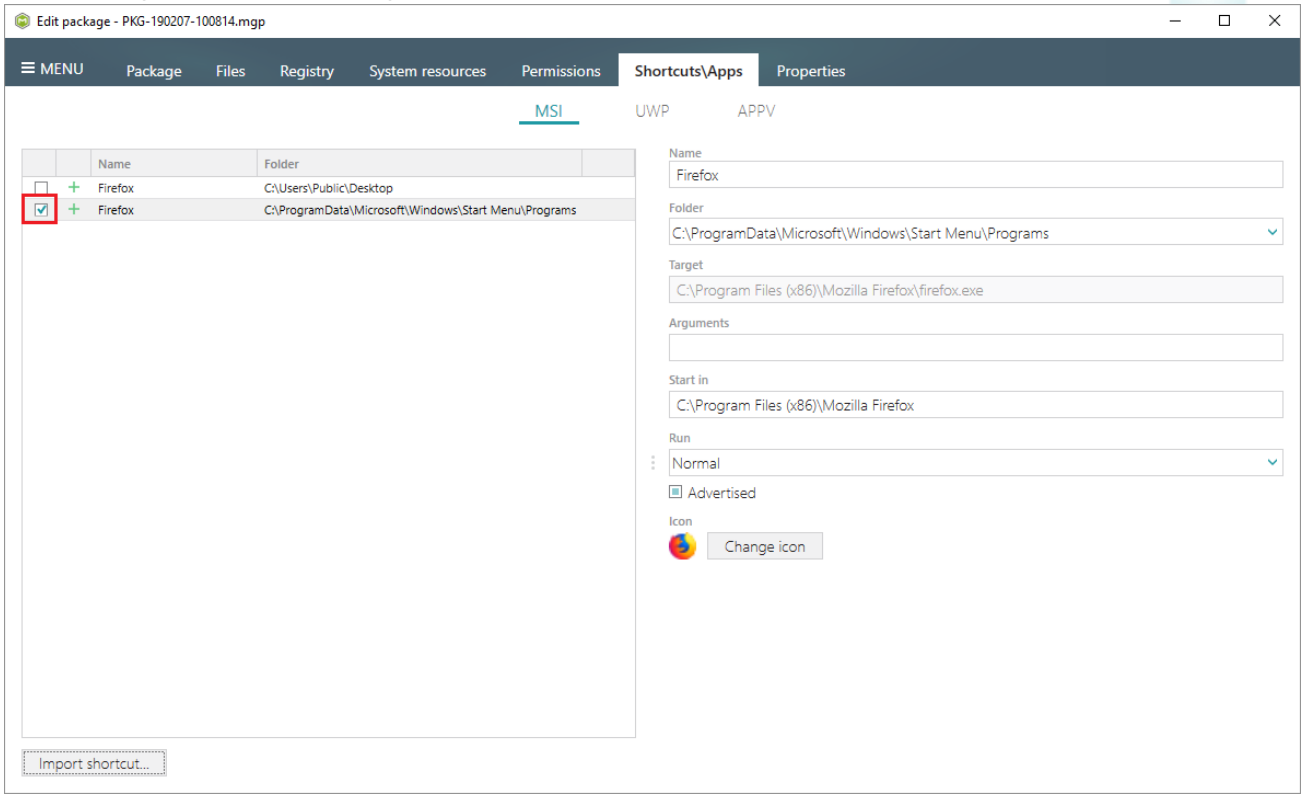
3.8.6.3.2 Include Excluded Shortcut

Include the necessary shortcuts, previously excluded from the project.

- Go to the Shortcuts\Apps -> MSI tab.



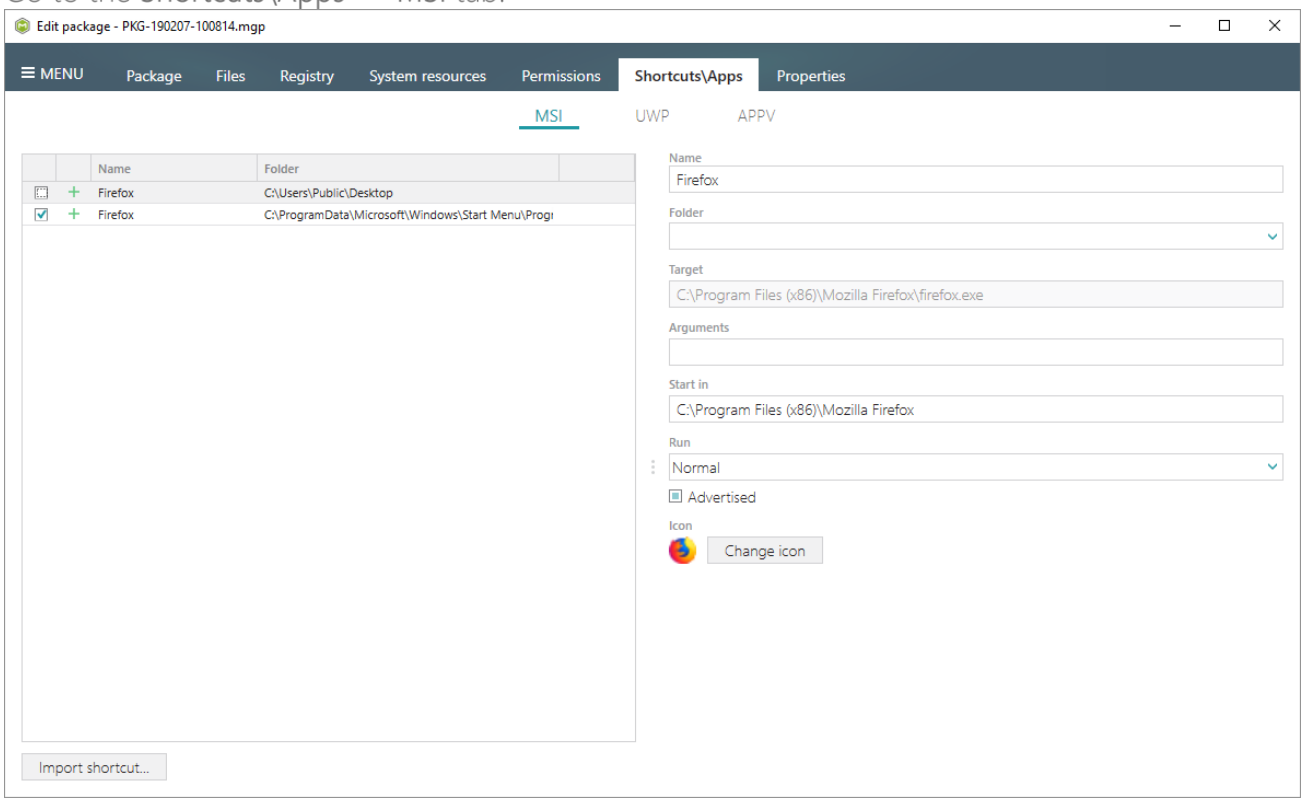
- To include, tick the checkbox, located before the shortcut name.



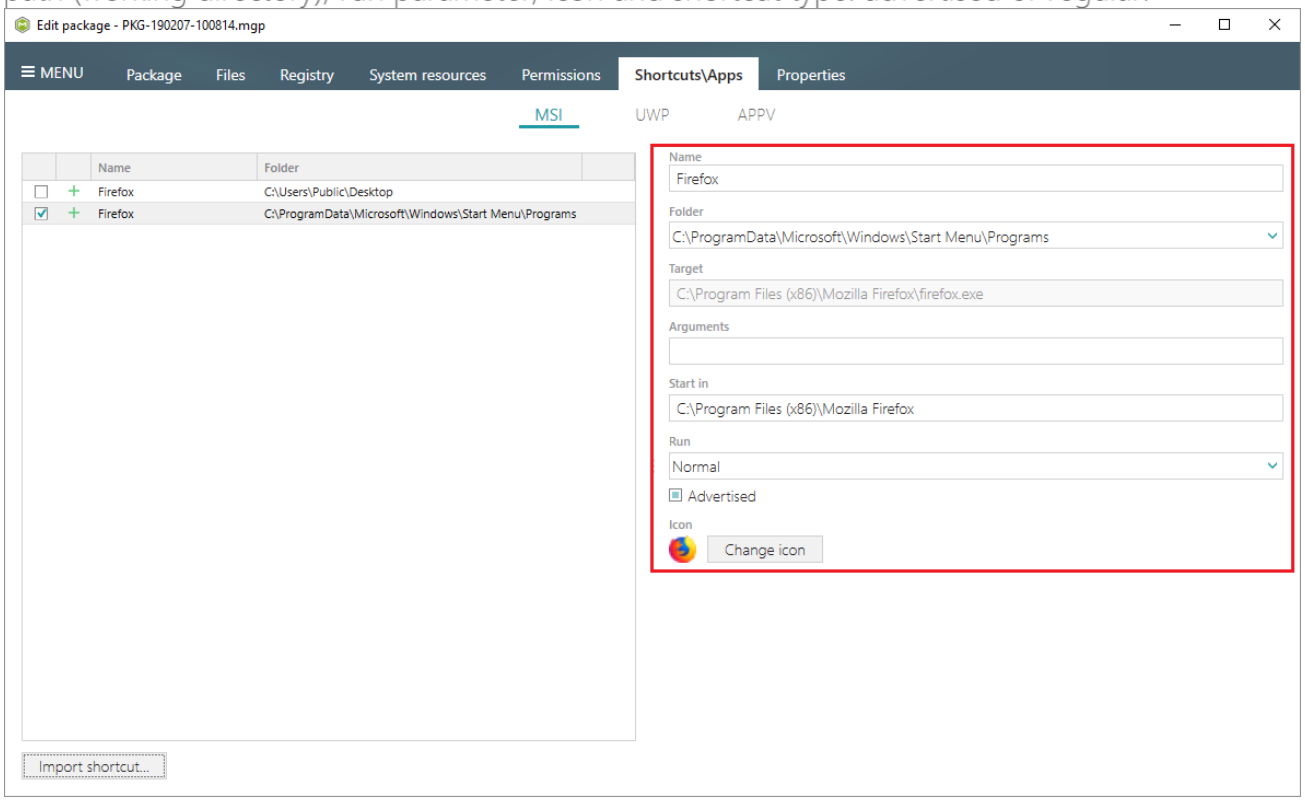
3.8.6.3.3 Edit Shortcut

Edit the shortcut detail such as name, destination folder, shortcut type, icon, etc.

- Go to the Shortcuts\Apps -> MSI tab.



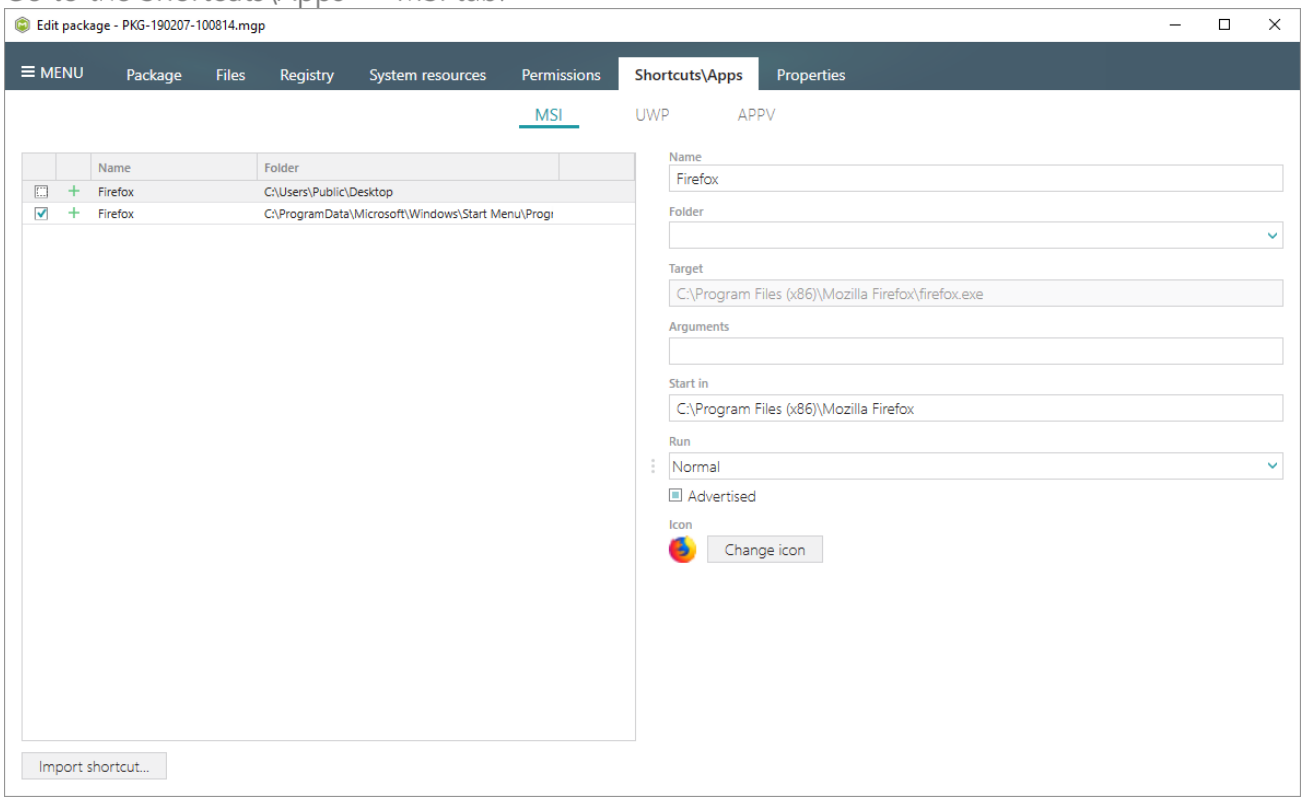
- Update the necessary values such as shortcut name, its destination folder, arguments, start in path (working directory), run parameter, icon and shortcut type: advertised or regular.



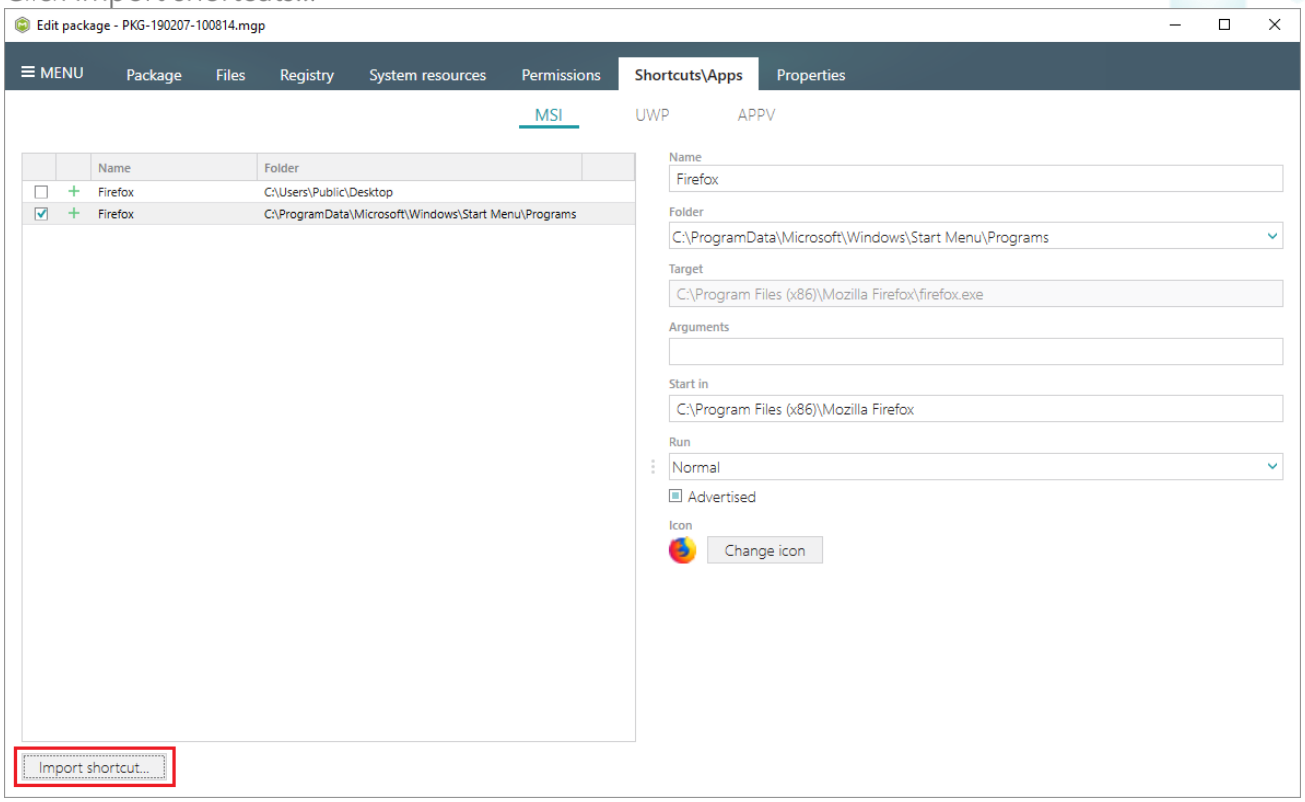
3.8.6.3.4 Import Shortcut

Import shortcuts from the file system to the project.

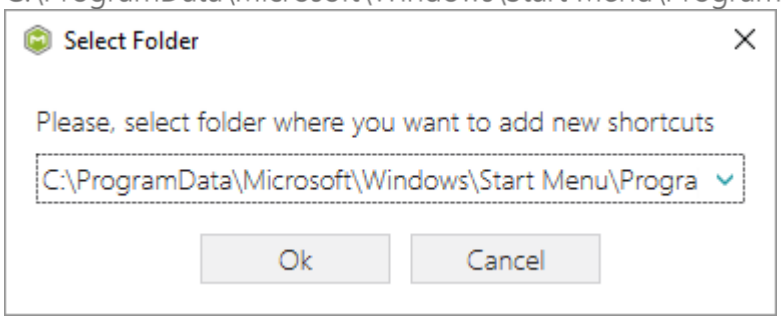
- Go to the Shortcuts\Apps -> MSI tab.



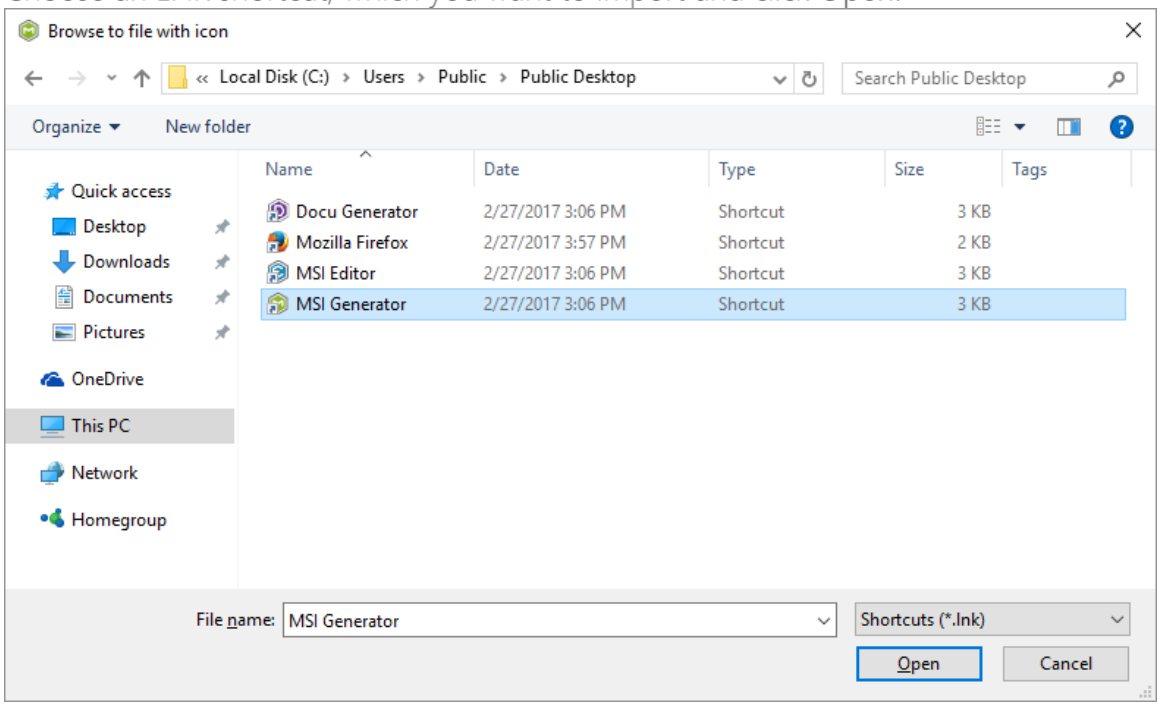
- Click Import shortcuts...



- Select a folder, to which you want to import shortcuts and click Ok.
NOTE To install a shortcut to the Start menu program, select the C:\ProgramData\Microsoft\Windows\Start Menu\Programs path.



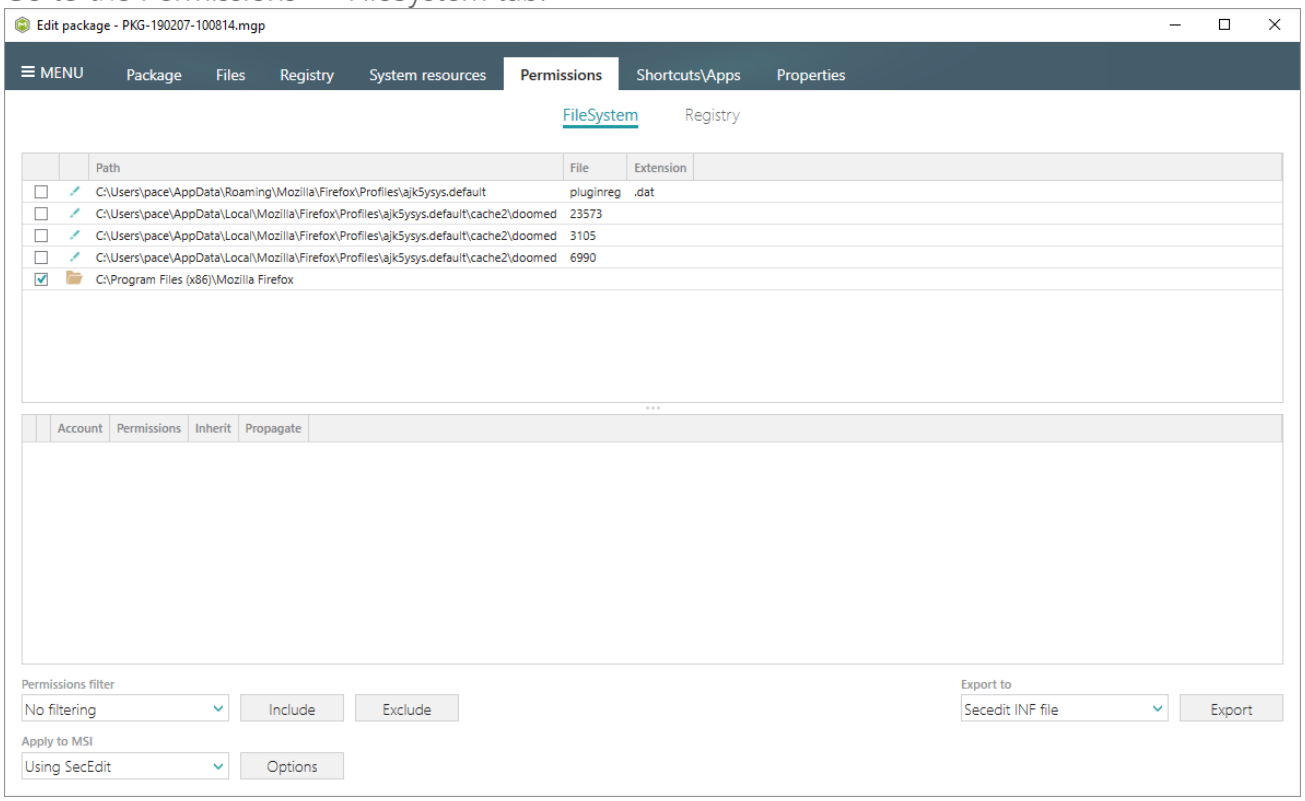
- Choose an LNK shortcut, which you want to import and click Open.



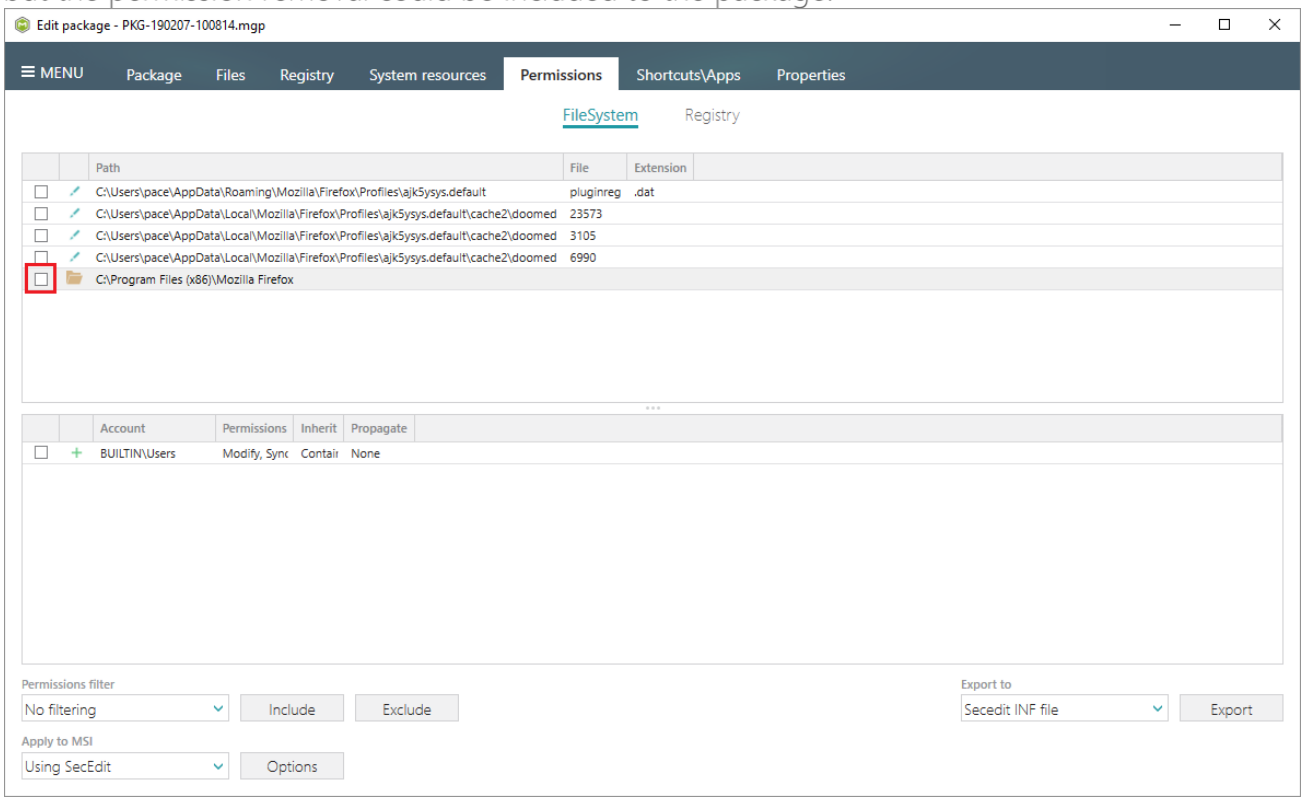
3.8.6.4 Control File/Folder Permissions

Control which of captured permission changes for files and folders will be included into the package.

- Go to the Permissions -> FileSystem tab.



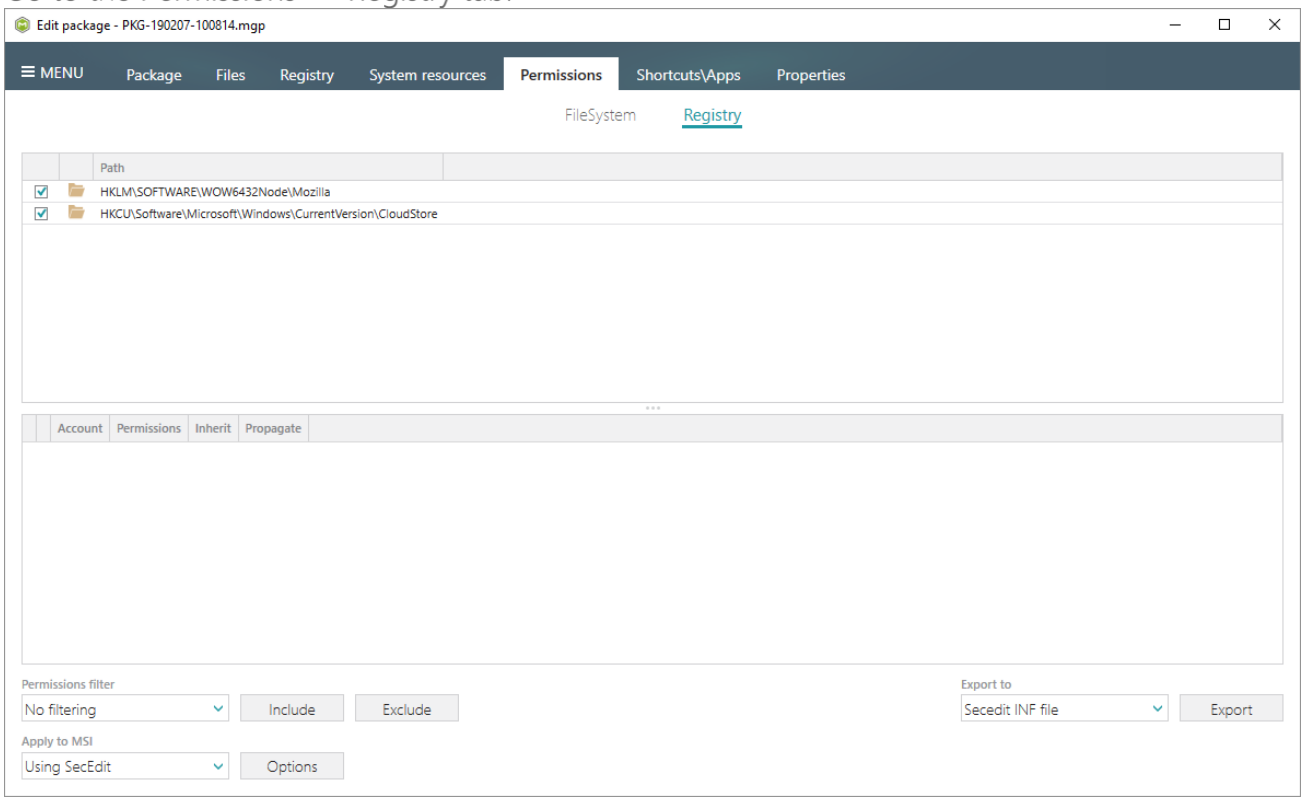
- To exclude the unnecessary permissions, untick the checkbox, located before the path. Respectively, to include the necessary permission, tick the checkbox back.
NOTE If permissions were removed during the capturing, they also will be displayed in the list, but the permission removal could be included to the package.



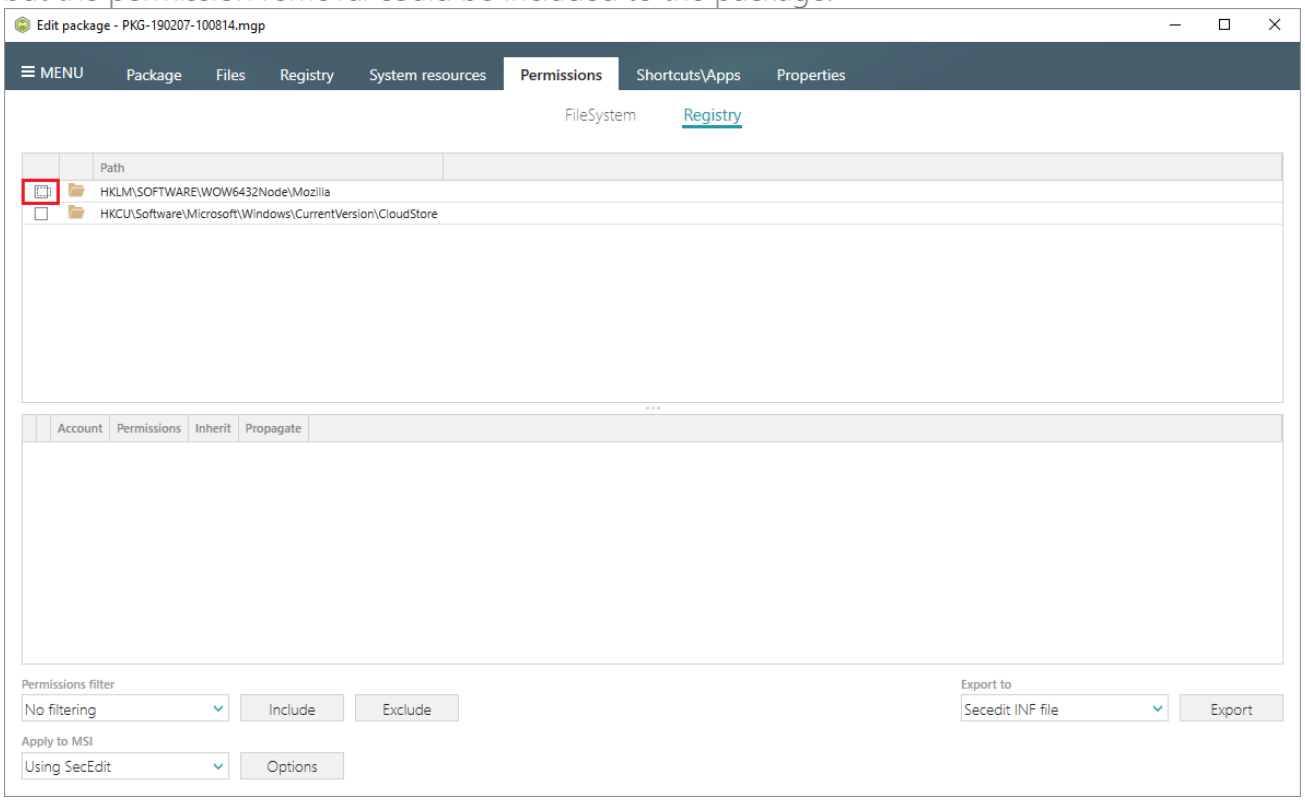
3.8.6.5 Control Registry Permissions

Control which of captured permission changes for registry keys will be included into the package.

- Go to the Permissions -> Registry tab.



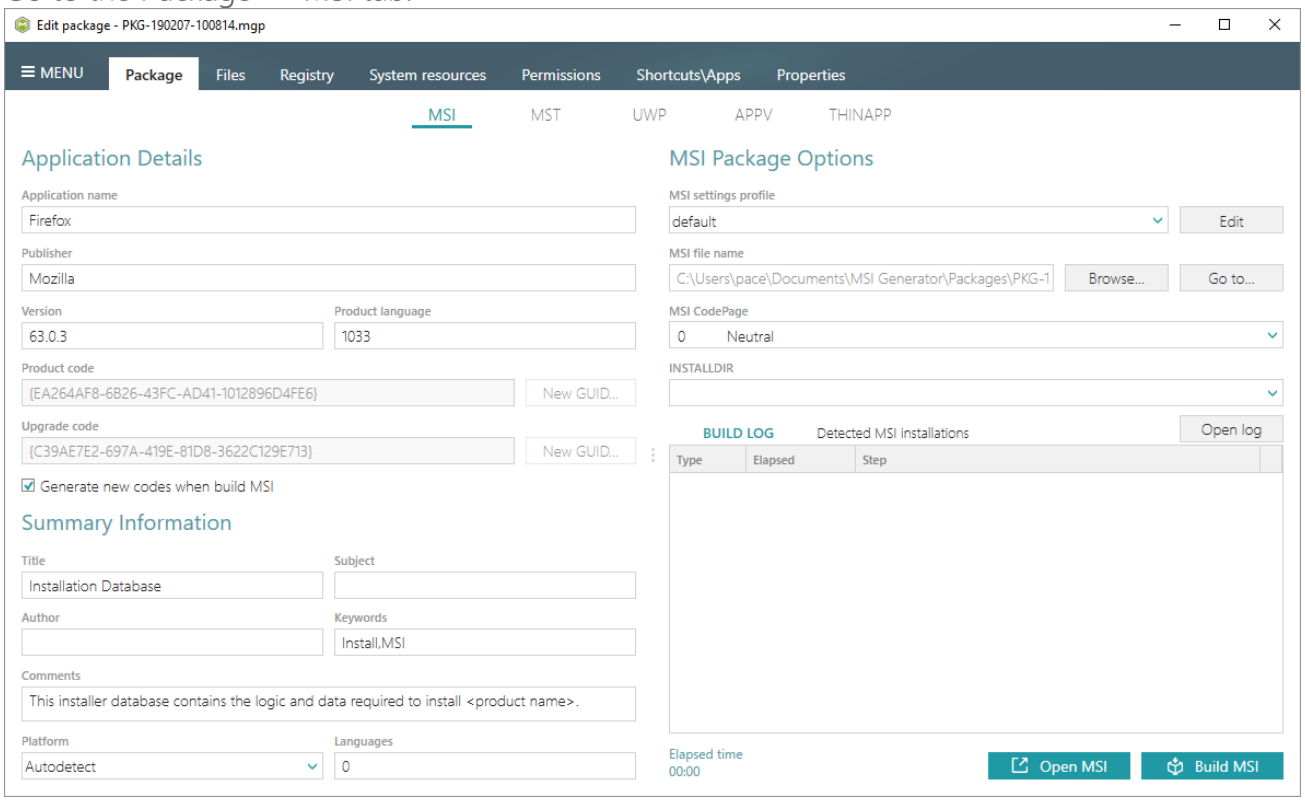
- To exclude the unnecessary permissions, untick the checkbox, located before the registry path. Respectively, to include the necessary permission, tick the checkbox back.
NOTE If permissions were removed during the capturing, they also will be displayed in the list, but the permission removal could be included to the package.



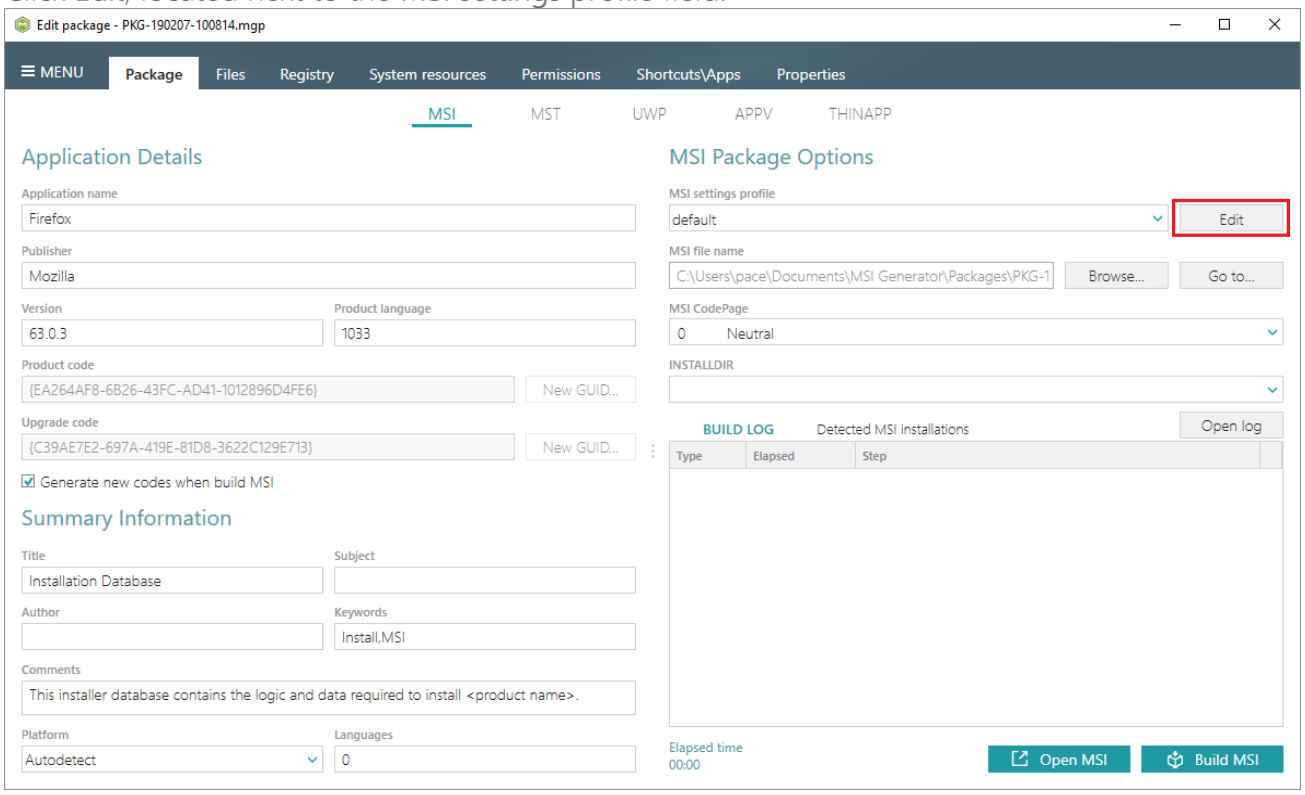
3.8.6.6 Select Compression Type

Select the compression type for the package files.

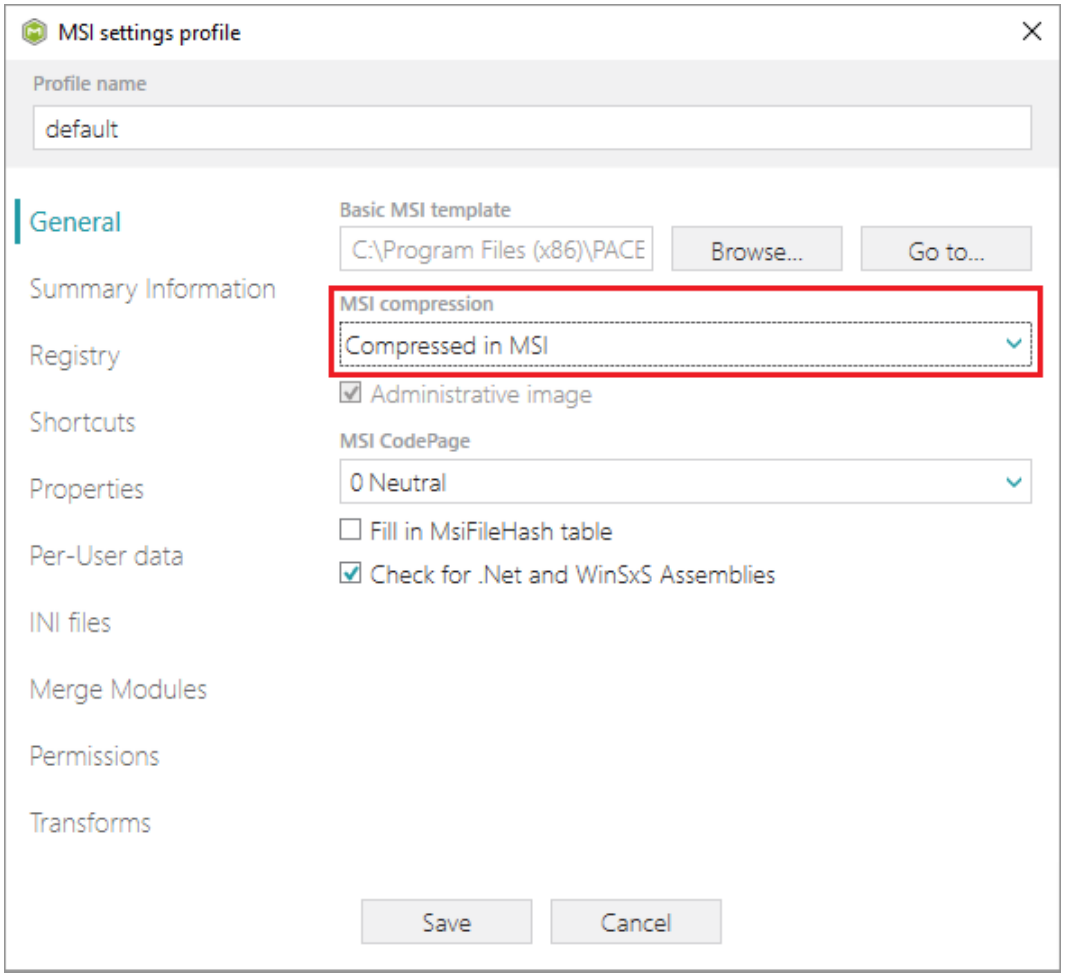
- Go to the Package -> MSI tab.



- Click Edit, located next to the MSI settings profile field.



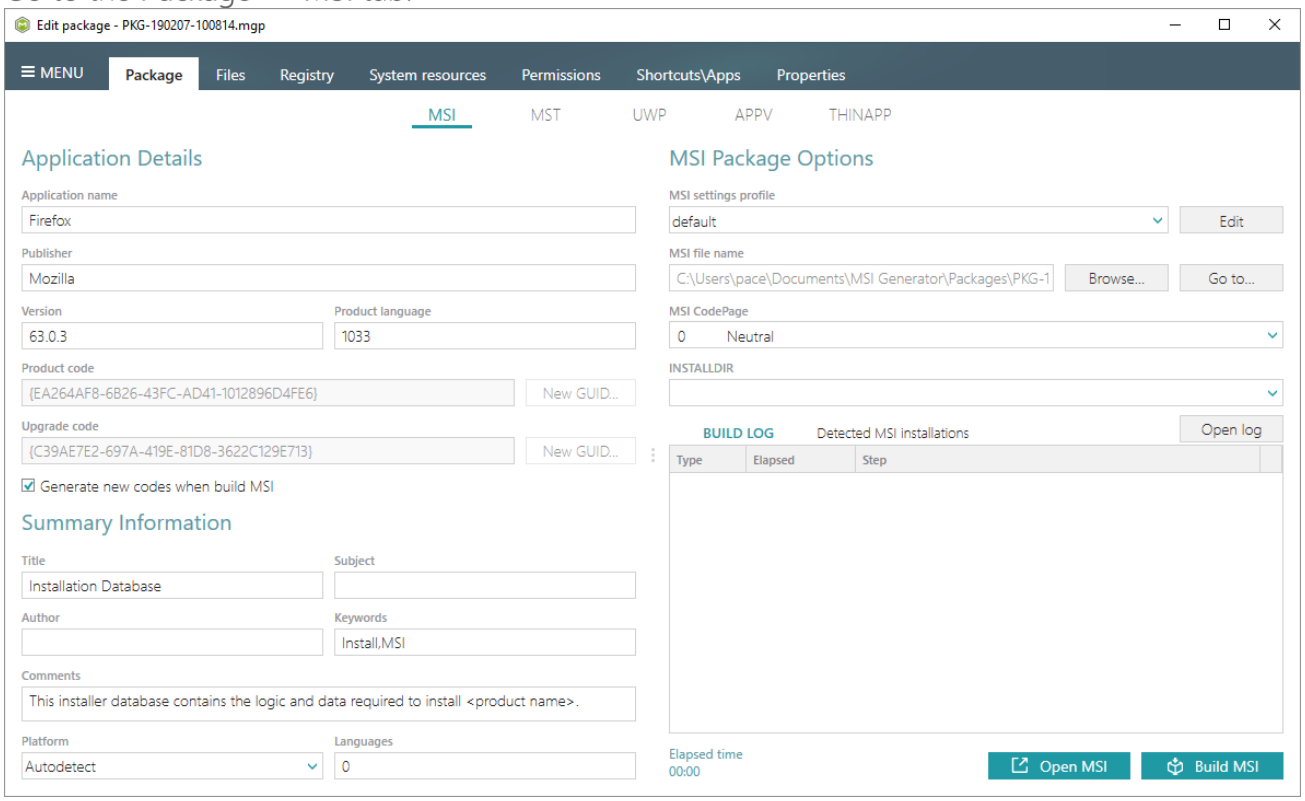
- On the General tab, select the necessary compression type from the MSI compression field and click Save.



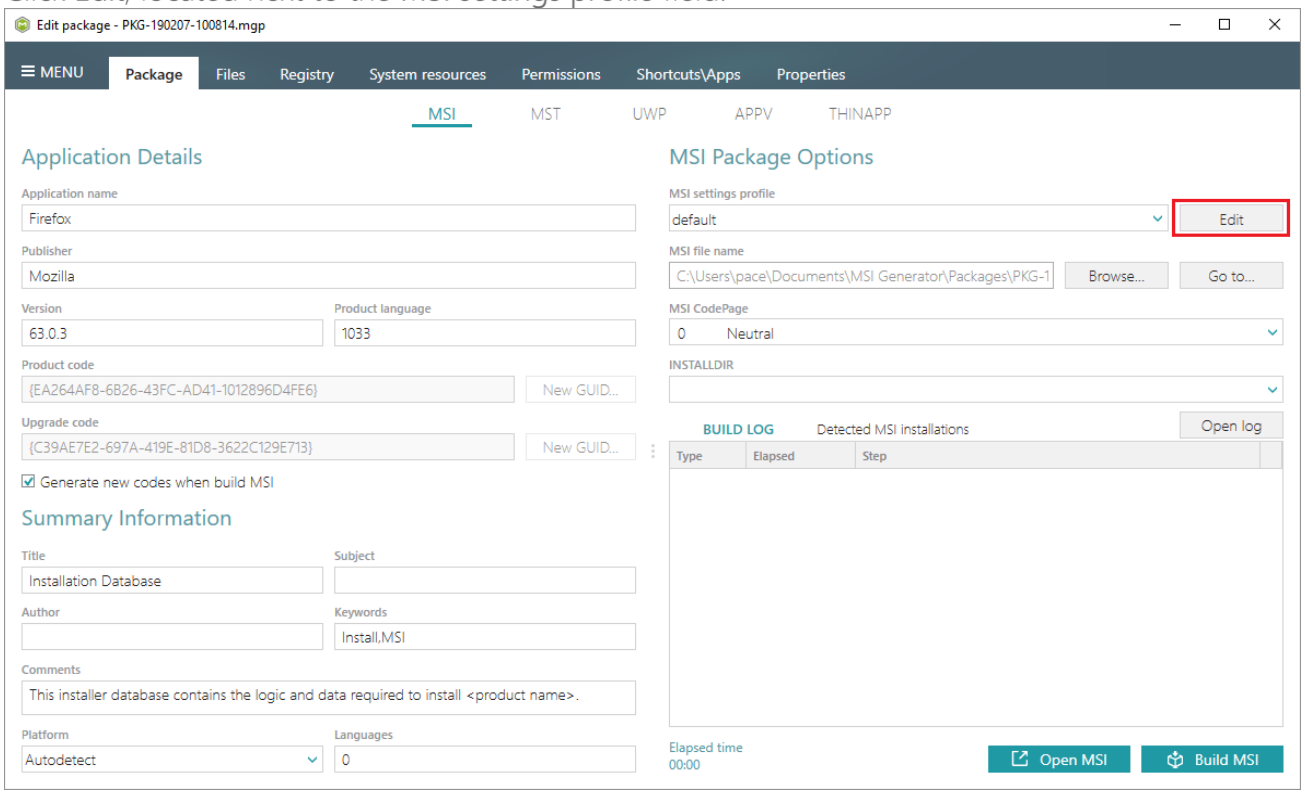
3.8.6.7 Use Advertising Tables

Select which of MSI advertizing tables will be used for storing the capturing registry.

- Go to the Package -> MSI tab.

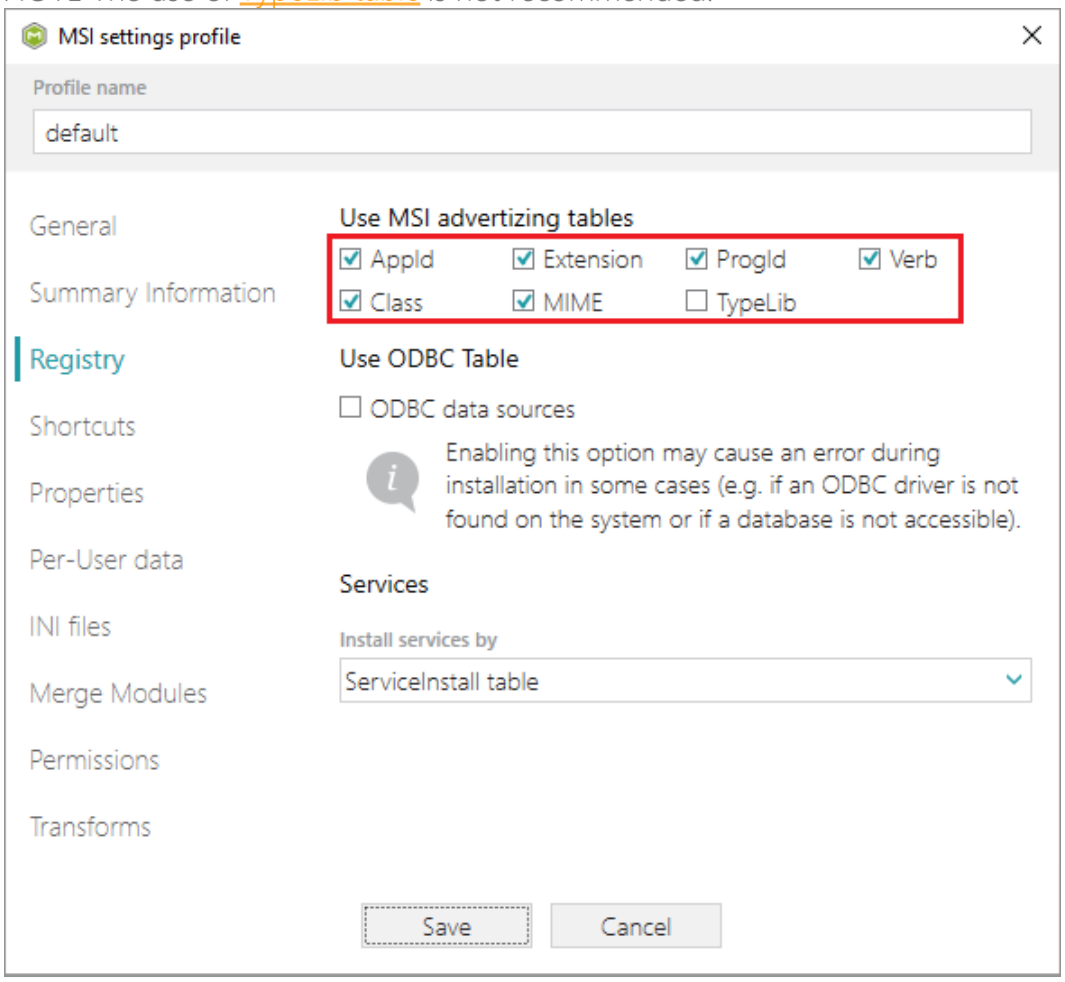


- Click Edit, located next to the MSI settings profile field.



- Go to the Registry tab and tick the advertizing tables, which you want to use for storing the captured registry. Afterwards, click Save.

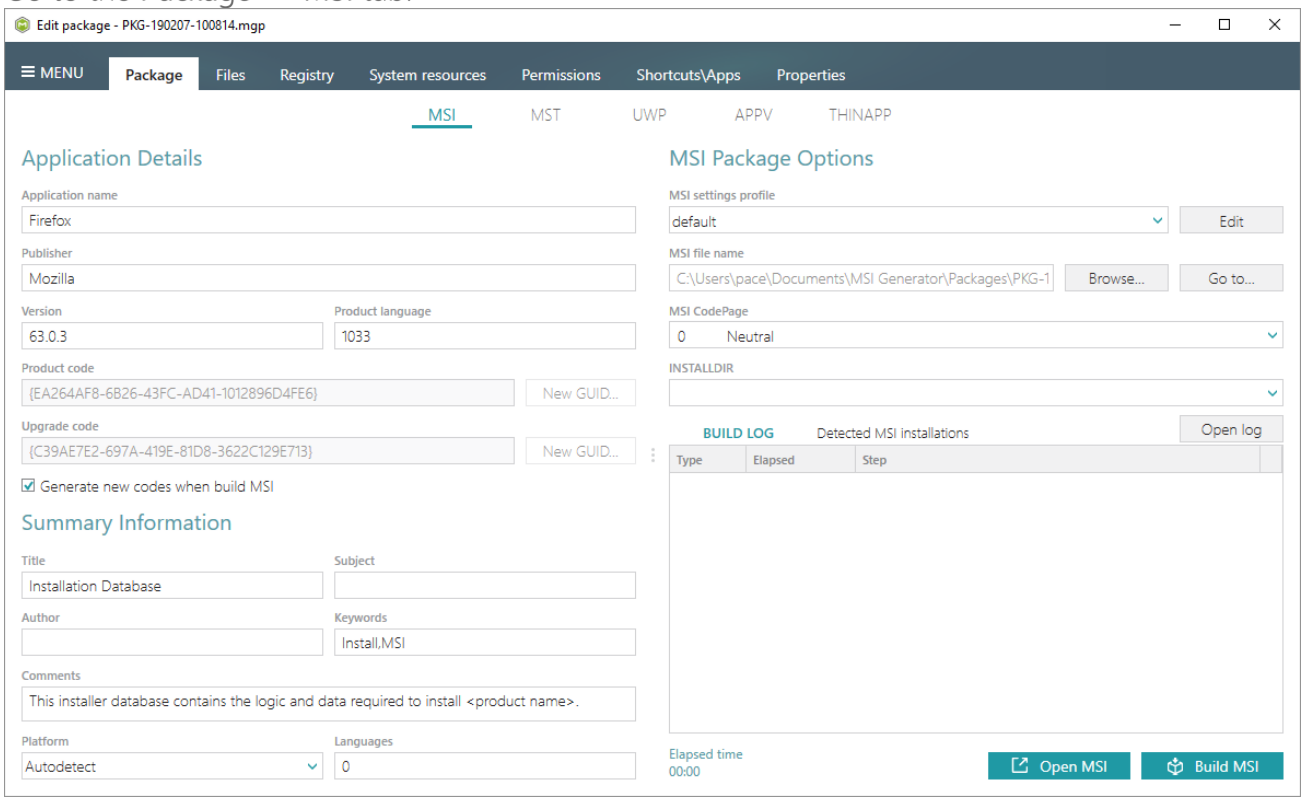
NOTE The use of TypeLib table is not recommended.



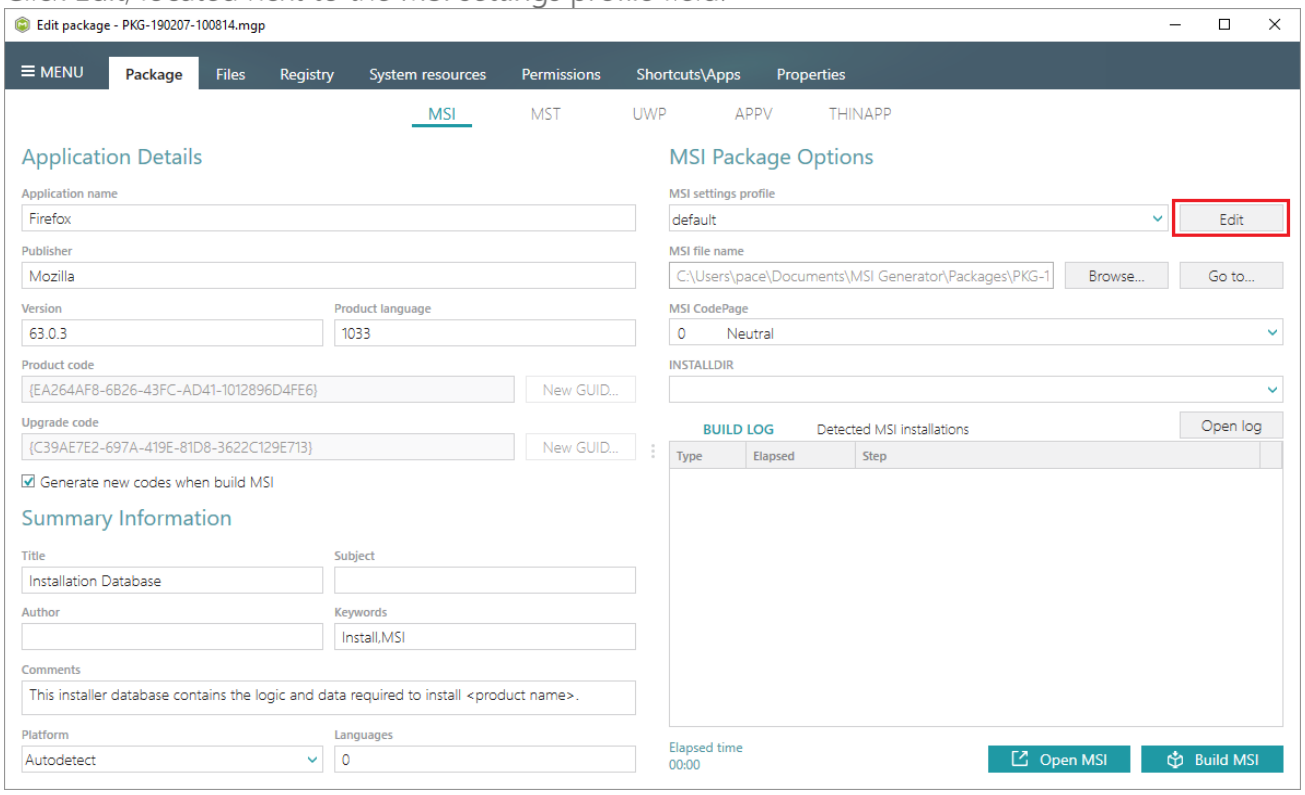
3.8.6.8 Add Active Setup

Intergrate the Active Setup to the MSI package automatically, if it contains per-user data.

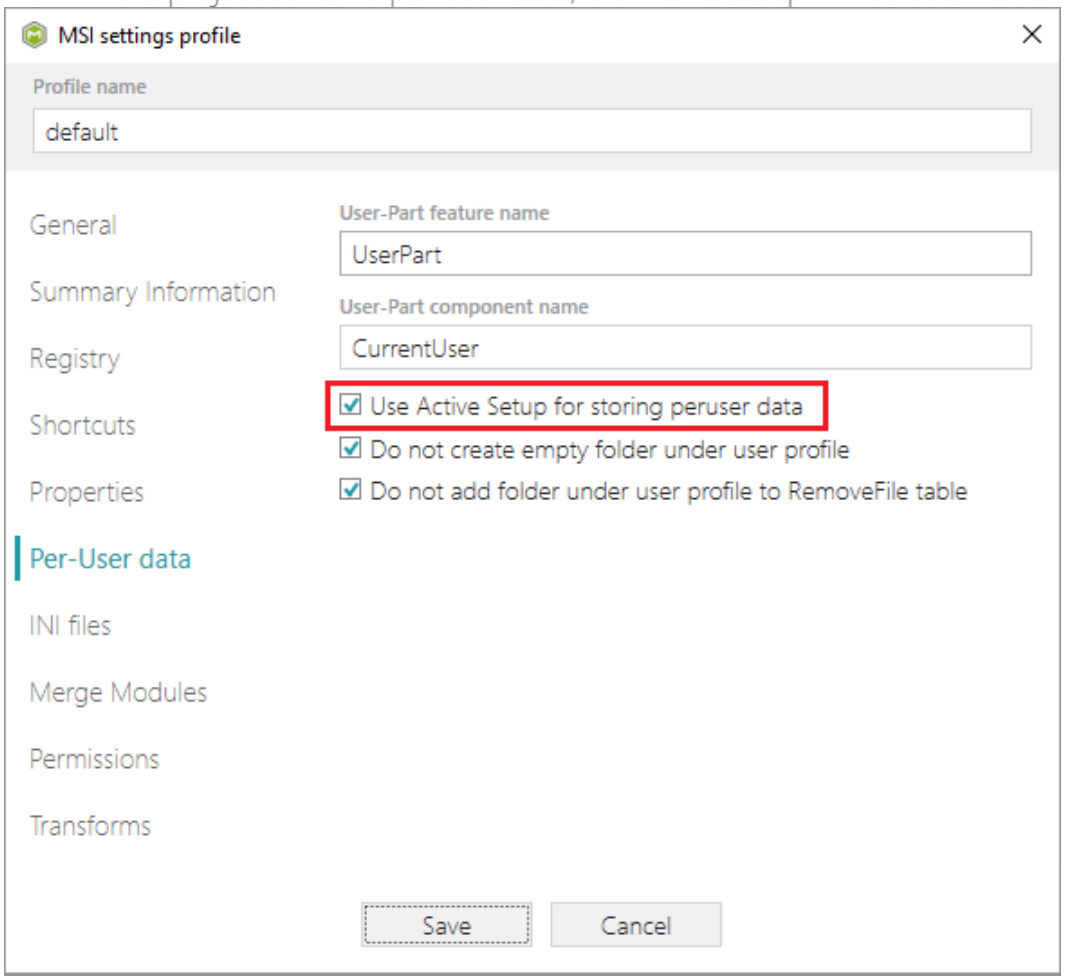
- Go to the Package -> MSI tab.



- Click Edit, located next to the MSI settings profile field.



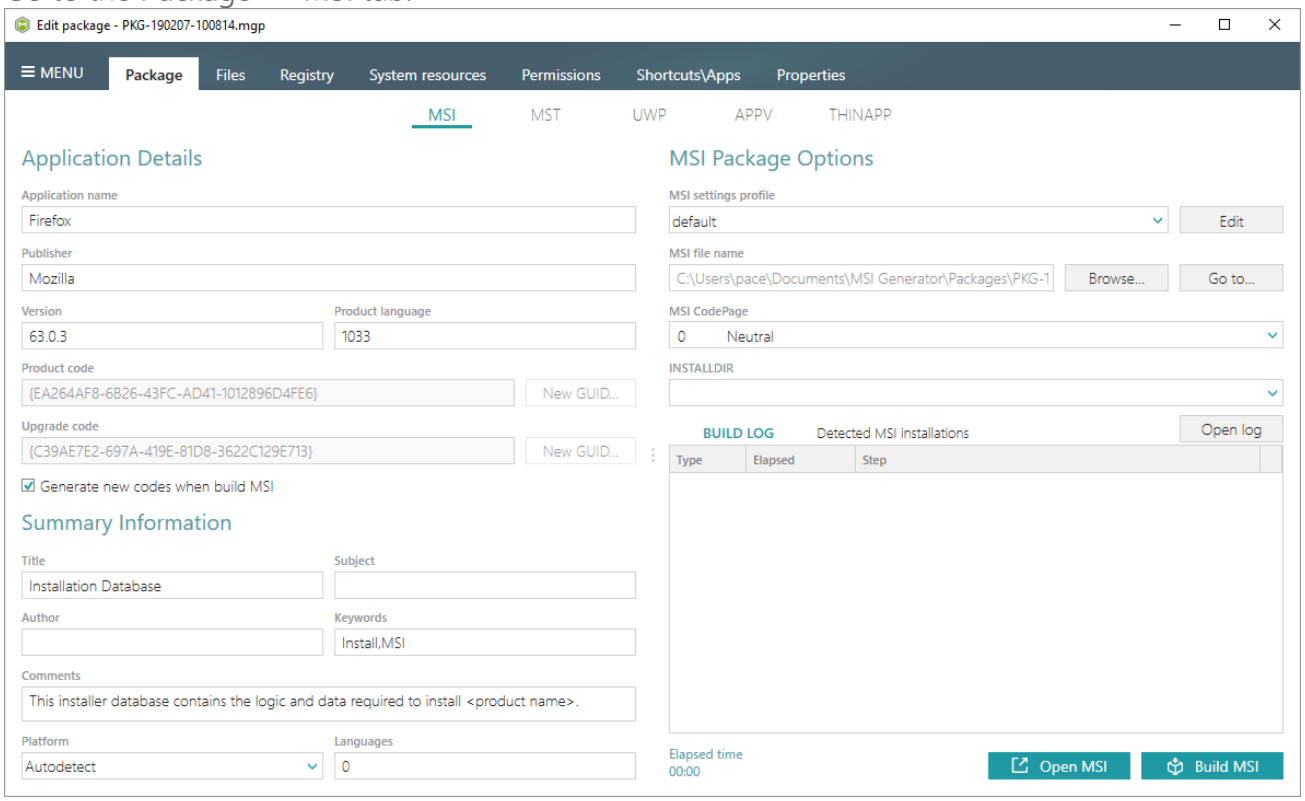
- Go to the Per-User data tab and tick the Use Active Setup for storing peruser data option. Then click Save.
  NOTE If the project contains per-user data, the Active Setup will be automatically integrated.



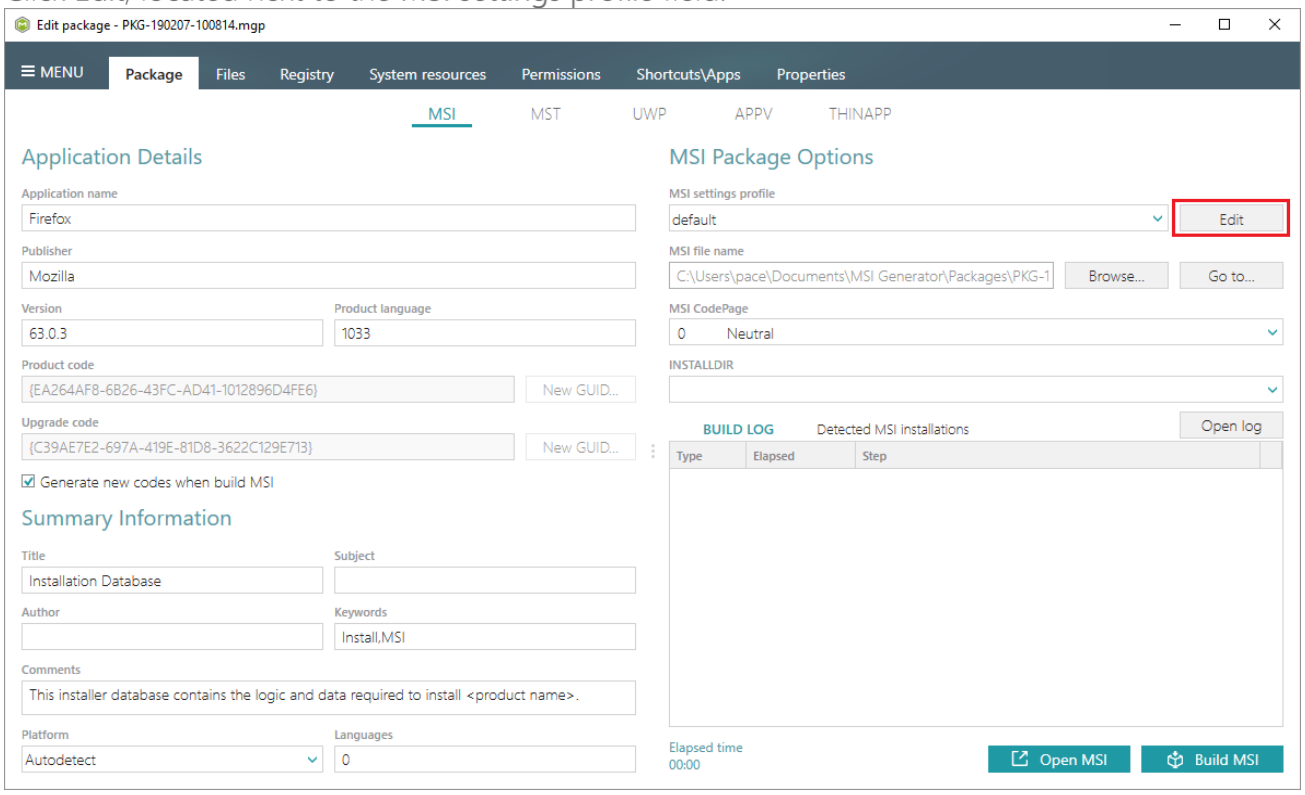
3.8.6.9 Set Summary Information

Set the default Summary Information values, which will be saved to each created MSI package.

- Go to the Package -> MSI tab.

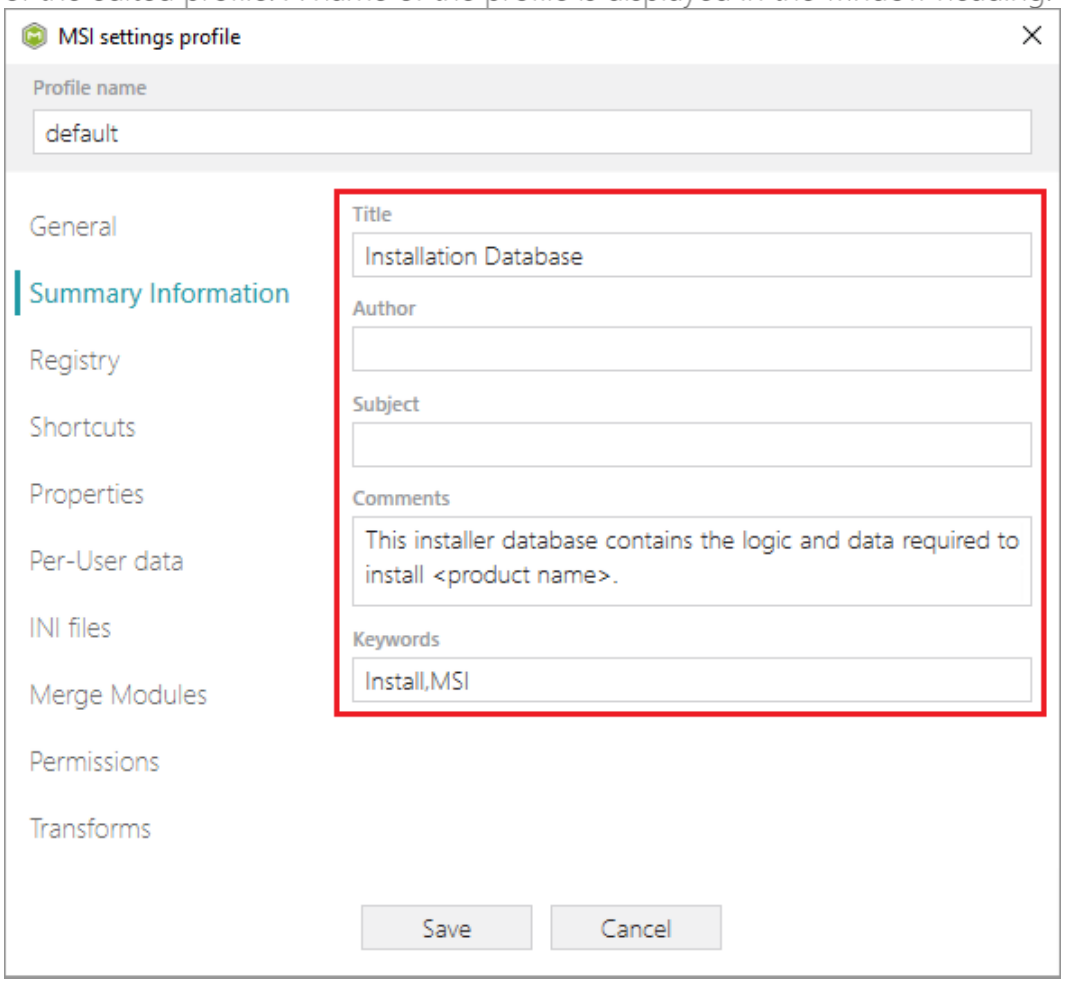


- Click Edit, located next to the MSI settings profile field.



- Go the Summary Information tab and enter the default Summary Information values. Afterwards, click Save.
  NOTE The specified Summary Information will be saved to each package, created with the use

of the edited profile. A name of the profile is displayed in the window heading.



3.8.7 MSIX/APPX Settings

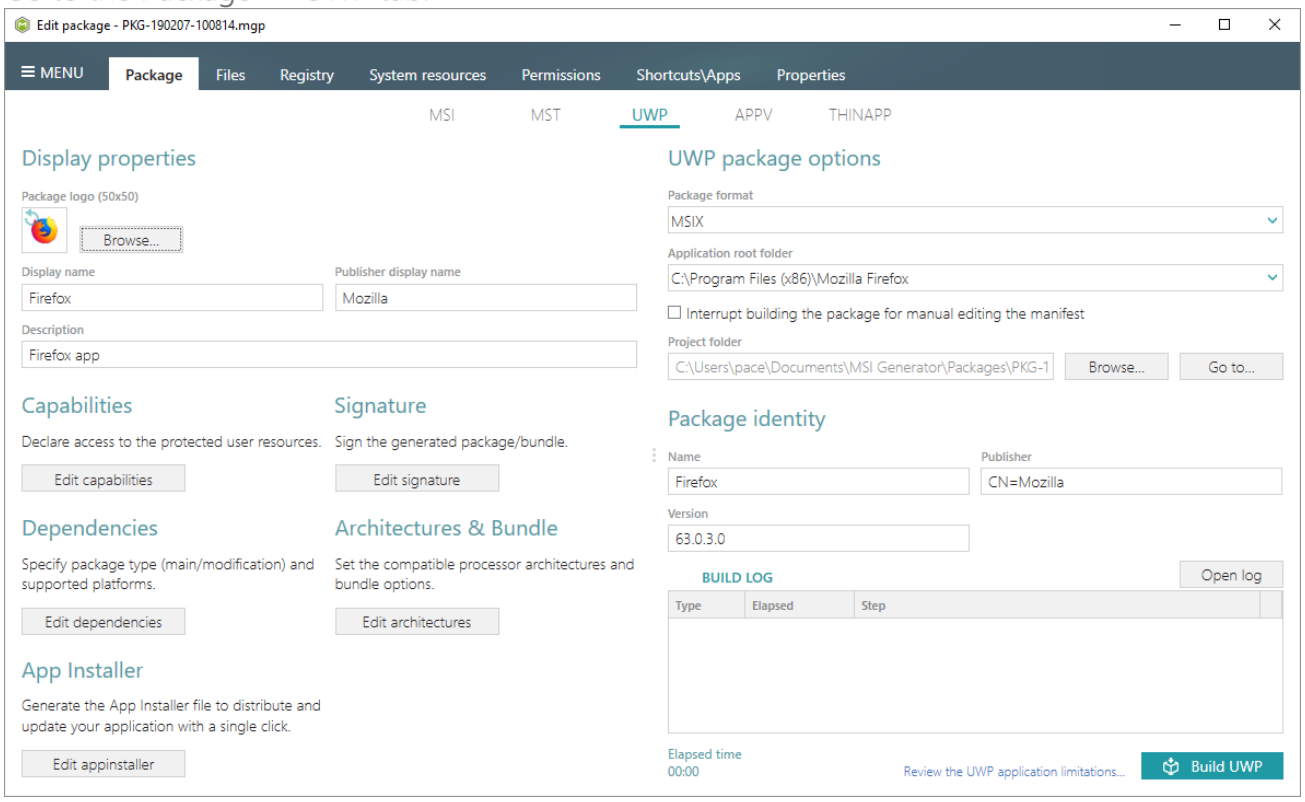
Choose a scenario that better suits your needs:

- Create Modification Package, described in section 3.8.7.1
  Create a modification package to customize the settings of the app, installed by main package.
- Set Display Properties, described in section 3.8.7.2
  Control how your UWP app package will be displayed during the installation and the list of installed apps.
- Set Identity Properties, described in section 3.8.7.3
  Define a globally unique identifier for your UWP package.
- Declare Capabilities, described in section 3.8.7.4
  Declares the access to protected user resources that the package requires.
- Set Supported Platforms, described in section 3.8.7.5
  Set the supported platforms (target device families) on which, the UWP package could be run.
- Manage Package Architectures, described in section 3.8.7.6
  Select the processor architectures (x86, x64, ARM) on which you want to run your UWP package.
- Create Bundle, described in section 3.8.7.7
  Combine the generated UWP packages into one single bundle.
- Sign Package/Bundle, described in section 3.8.7.8
  Sign the generated package or bundle with a certificate for the testing purposes or distribution.
- Edit Manifest Manually, described in section 3.8.7.9
  Edit the package manifest file(s) manually to modify existing or add new specific package settings.
- Manage Apps and Visual Assets, described in section 3.8.7.10
  Include/exclude captured apps, add new ones and control their appearance settings such as logos, name, background color, etc.
- Generate AppInstaller, described in section 3.8.7.11
  Generate the AppInstaller file for the non-Store distribution and automatic update of your UWP package.

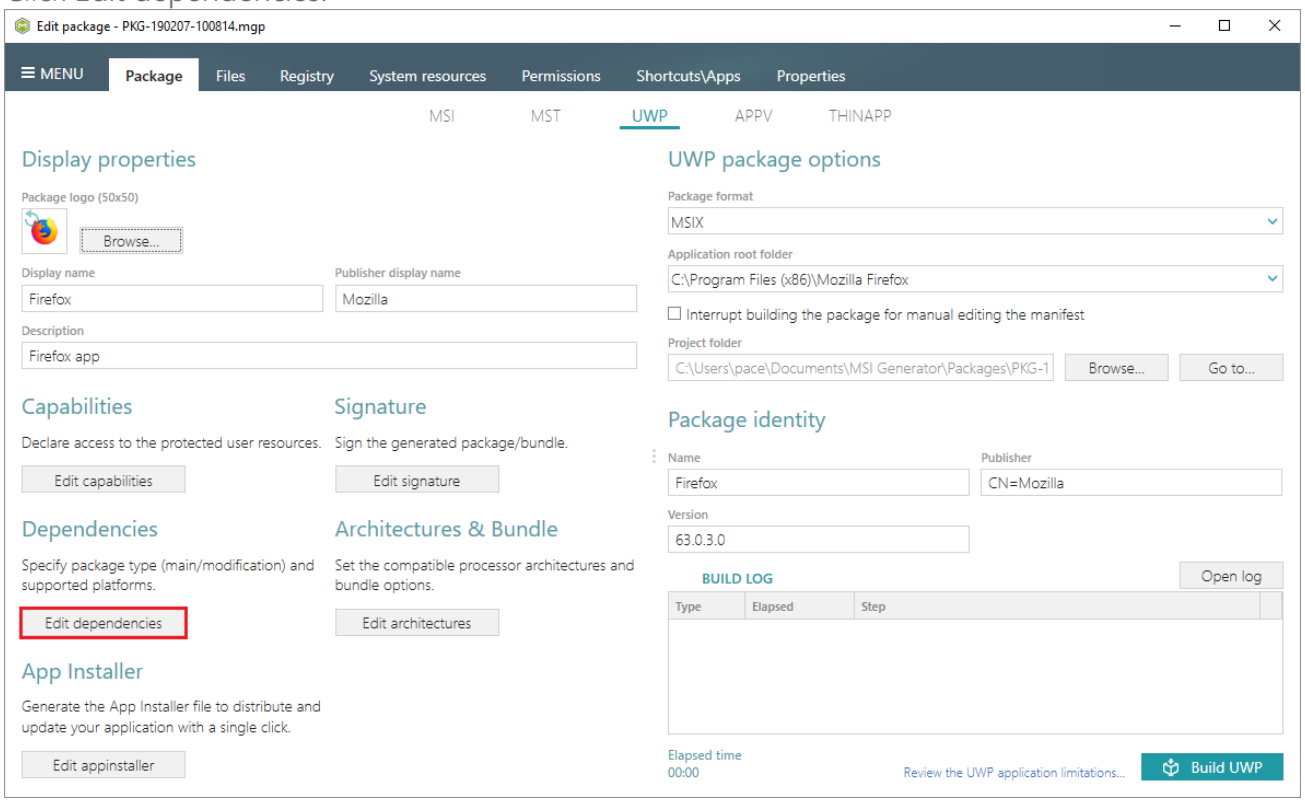
3.8.7.1 Create Modification Package

Create a modification package to customize the settings of the app, installed by main package. At runtime the application will view resources of both the main package and the modification package. For information, see MSIX modification packages on Windows 10 version 1809 and MSIX modification packages on Windows 10 Insider Preview Build 18312 and later.

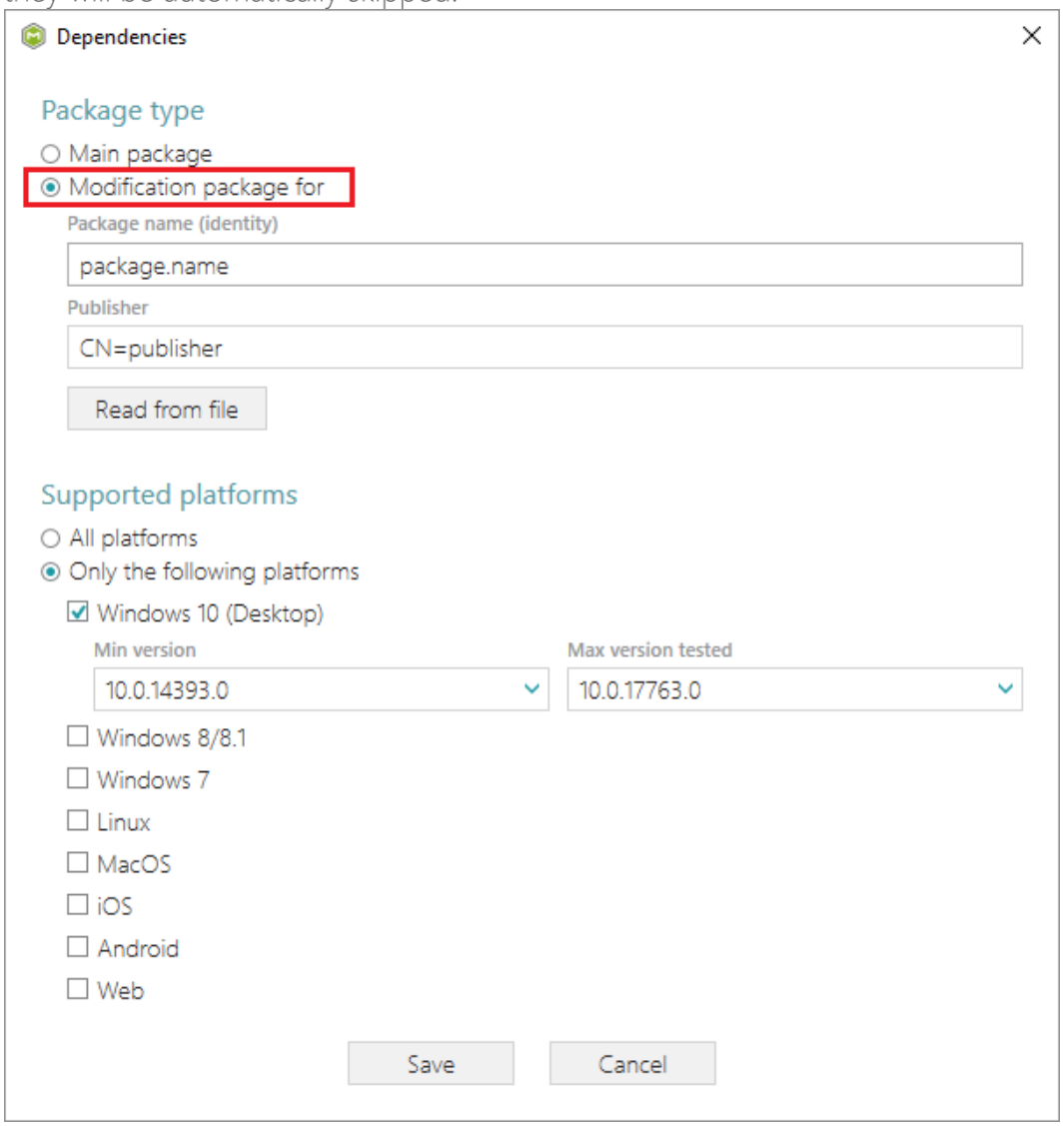
- Go to the Package -> UWP tab.



- Click Edit dependencies.



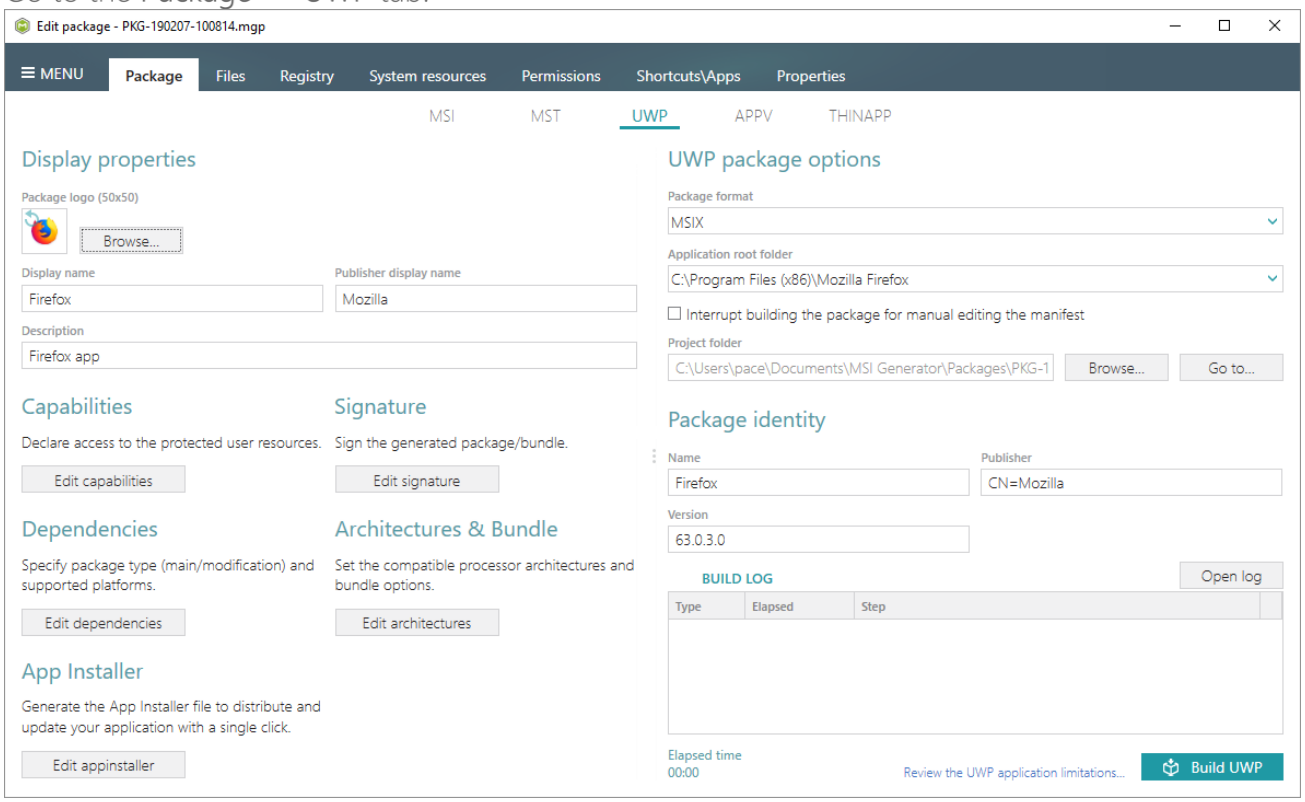
- Select the Modification package for option and either enter a package identity name and publisher of the main package manually or click Read from file to get the necessary identity information from the main package automatically. Afterwards, click Save.
  NOTE The modification package could not declare capabilities and contain apps, therefore they will be automatically skipped.



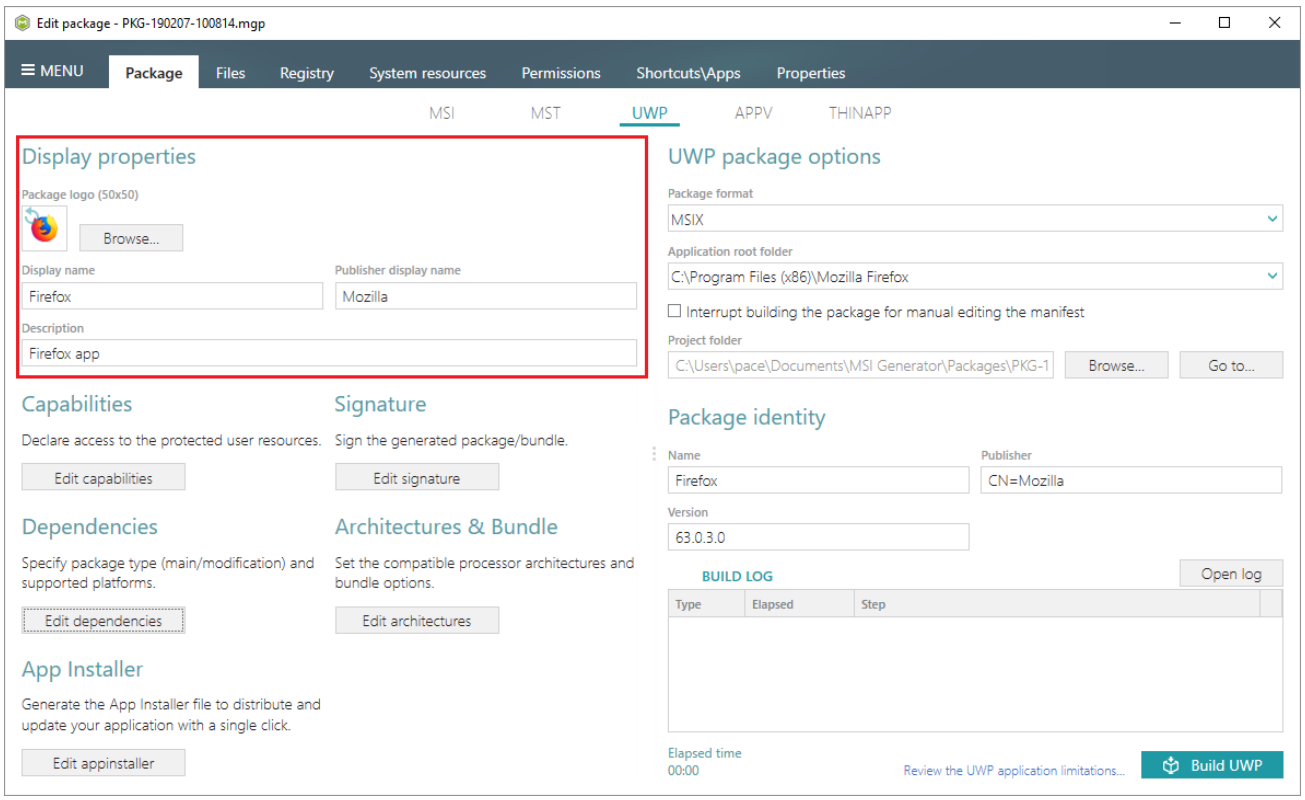
3.8.7.2 Set Display Properties

Control how your UWP app package will be displayed during the installation and the list of installed apps.

- Go to the Package -> UWP tab.



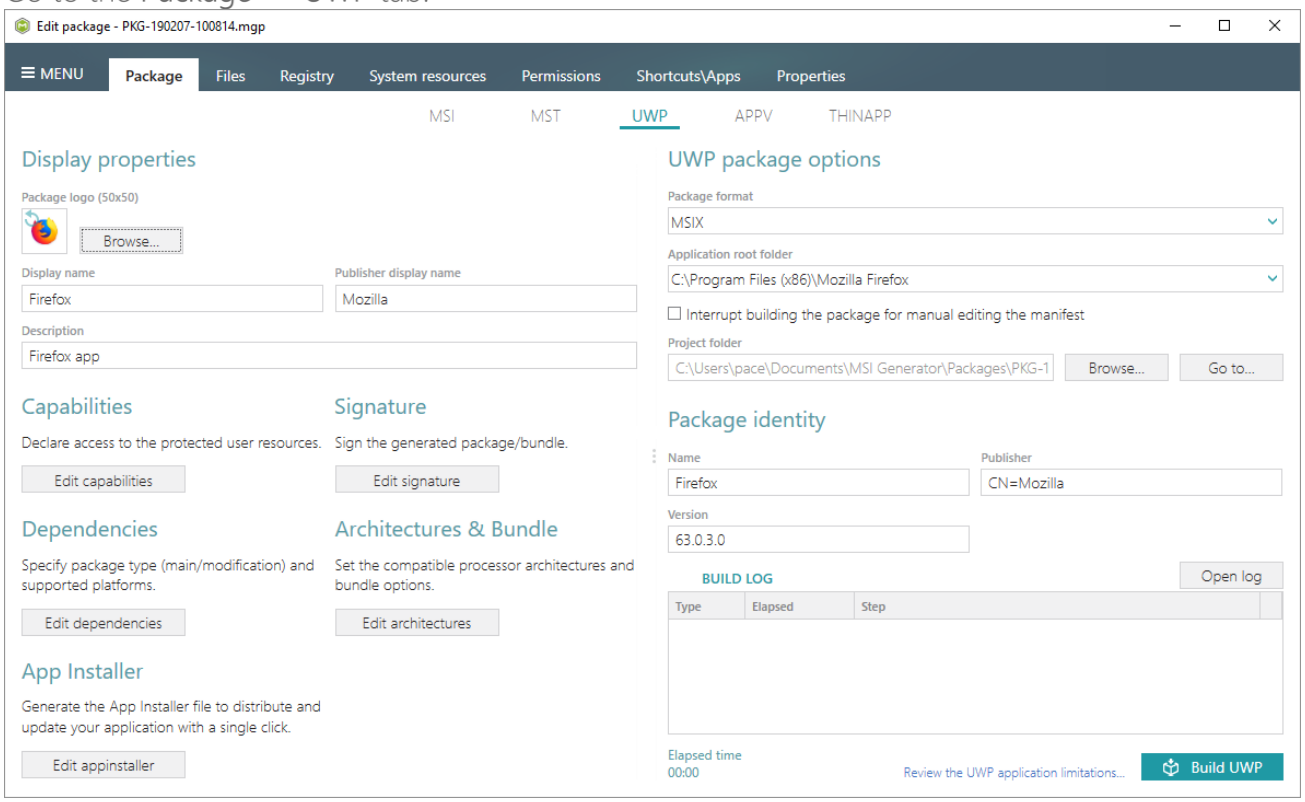
- Set a new icon, diplay name, publisher dispay name and description in the Display properties section.



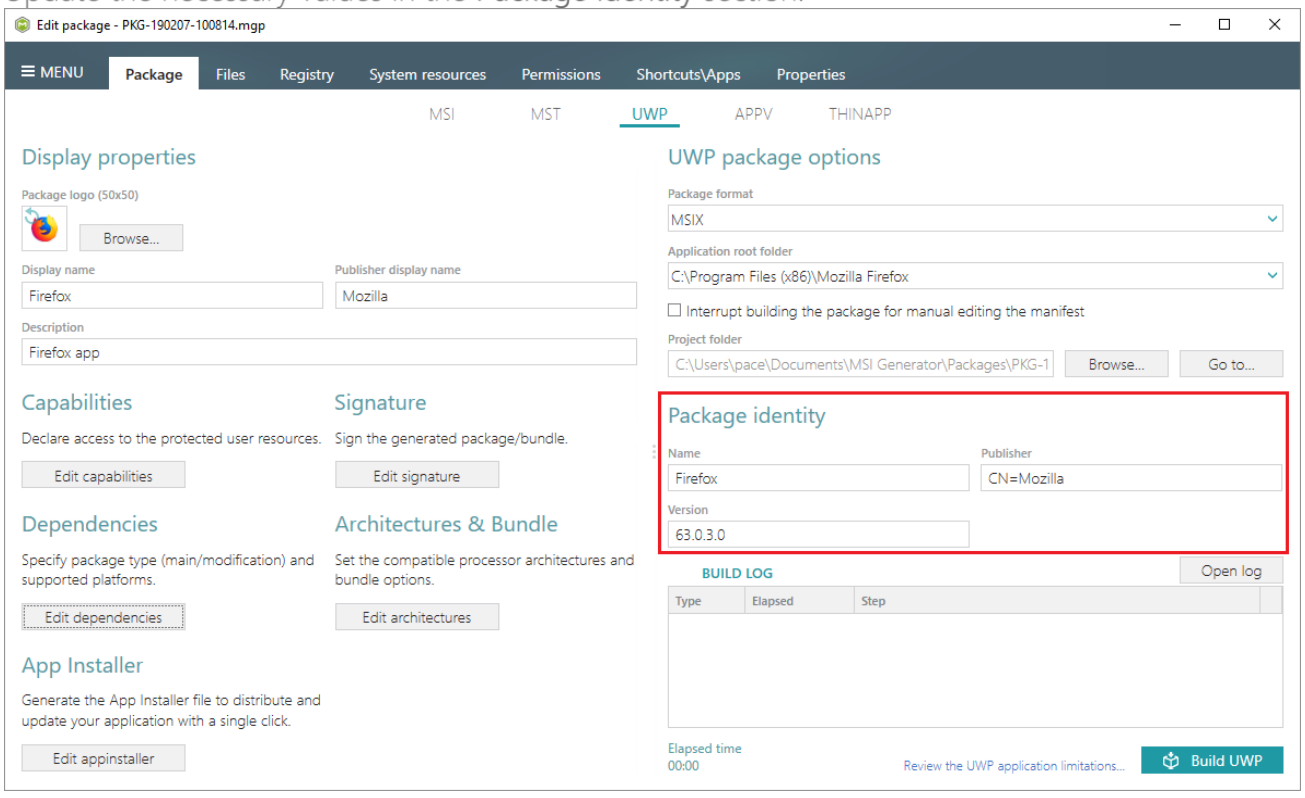
3.8.7.3 Set Identity Properties

Define a globally unique identifier for your UWP package. A package identity is represented as a tuple of attributes of the package.

- Go to the Package -> UWP tab.



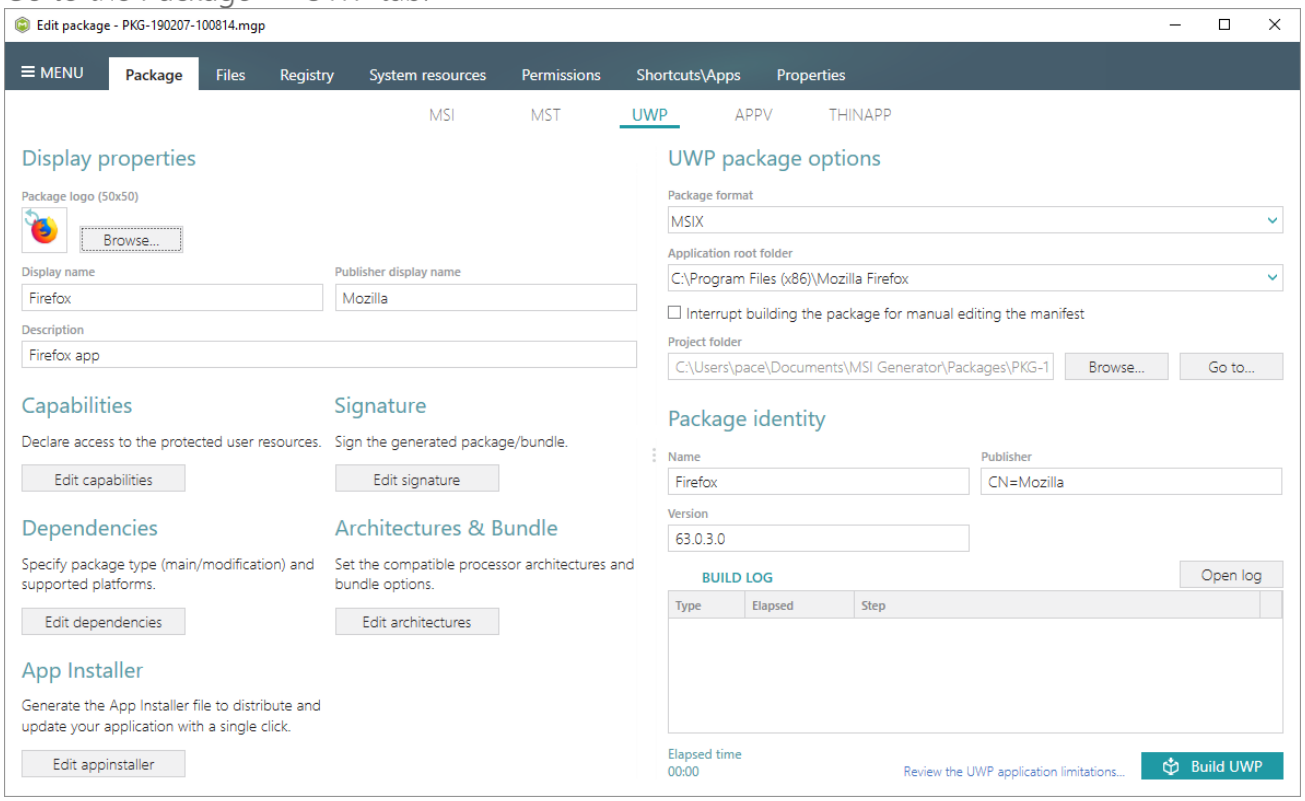
- Update the necessary values in the Package identity section.



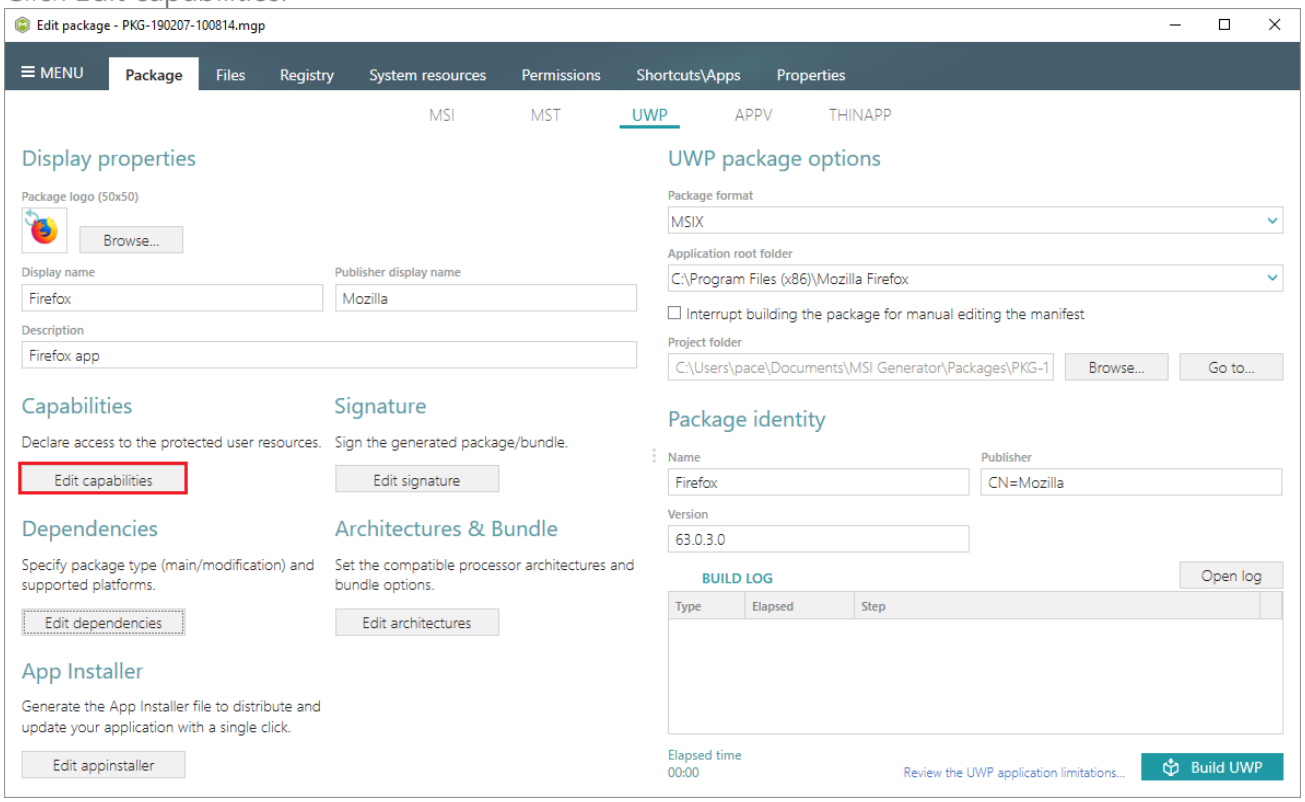
3.8.7.4 Declare Capabilities

Declares the access to protected user resources that the package requires.

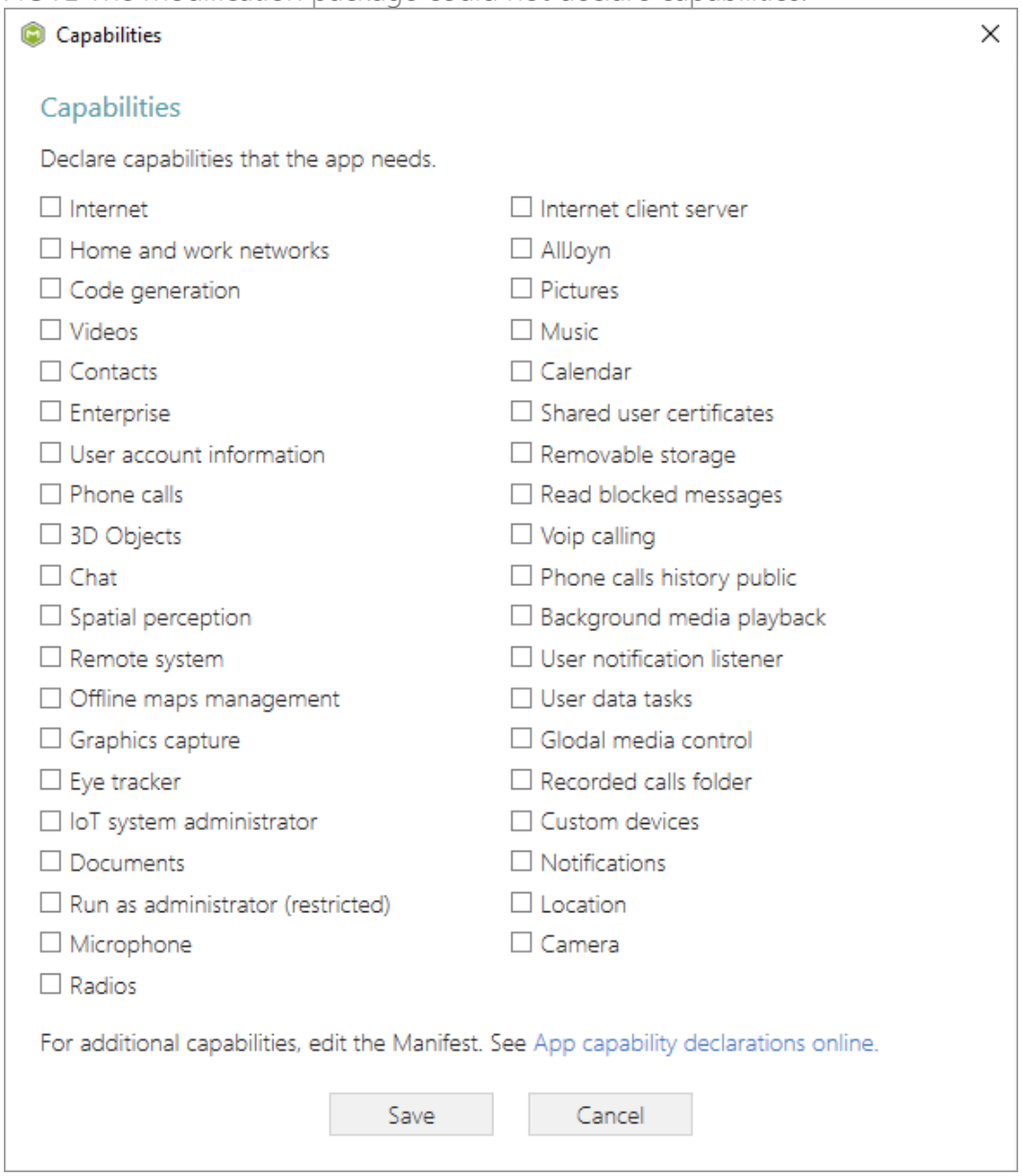
- Go to the Package -> UWP tab.



- Click Edit capabilities.



- Select the necessary capabilities and click Save.
  NOTE The modification package could not declare capabilities.

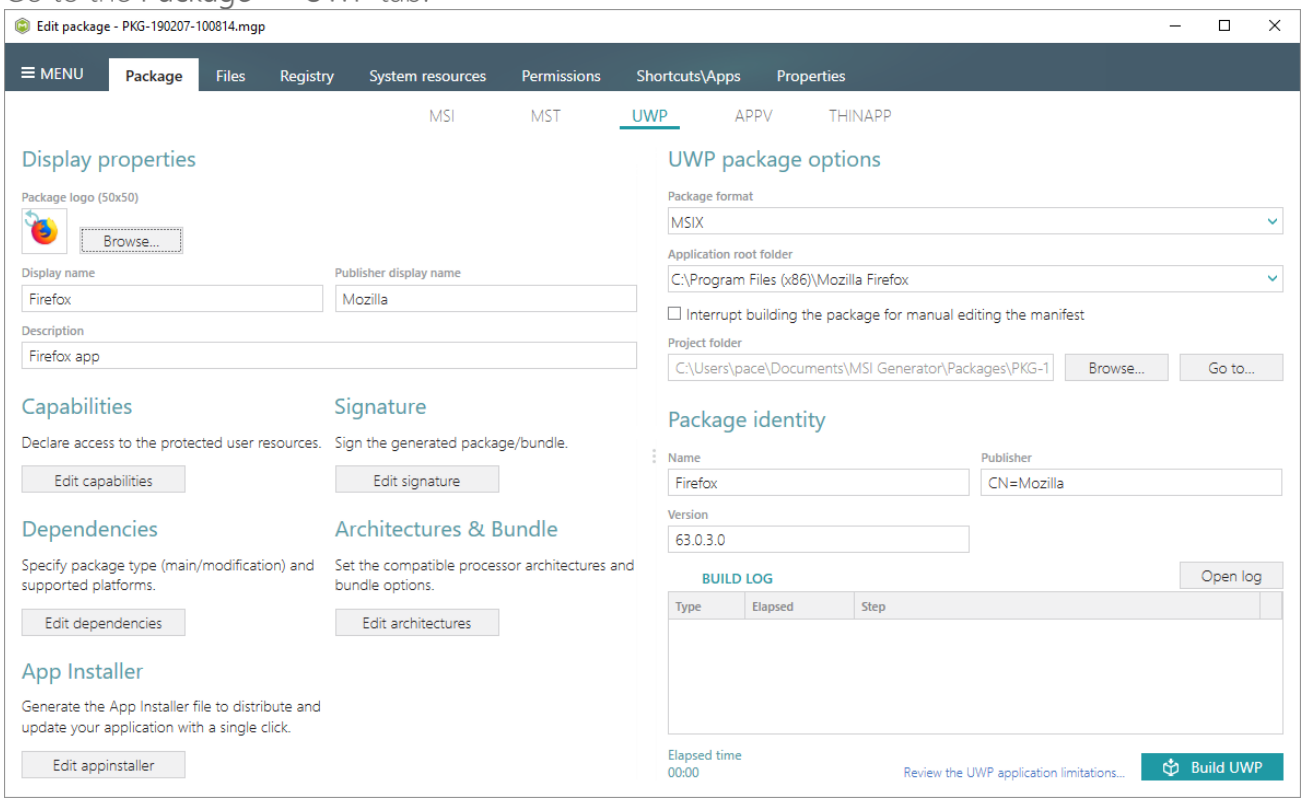


- If the list does not contain the necessary capability, edit the package manifest to specify this capability manually. For information, see App capability declarations. For editing the manifest, see section 3.8.7.9

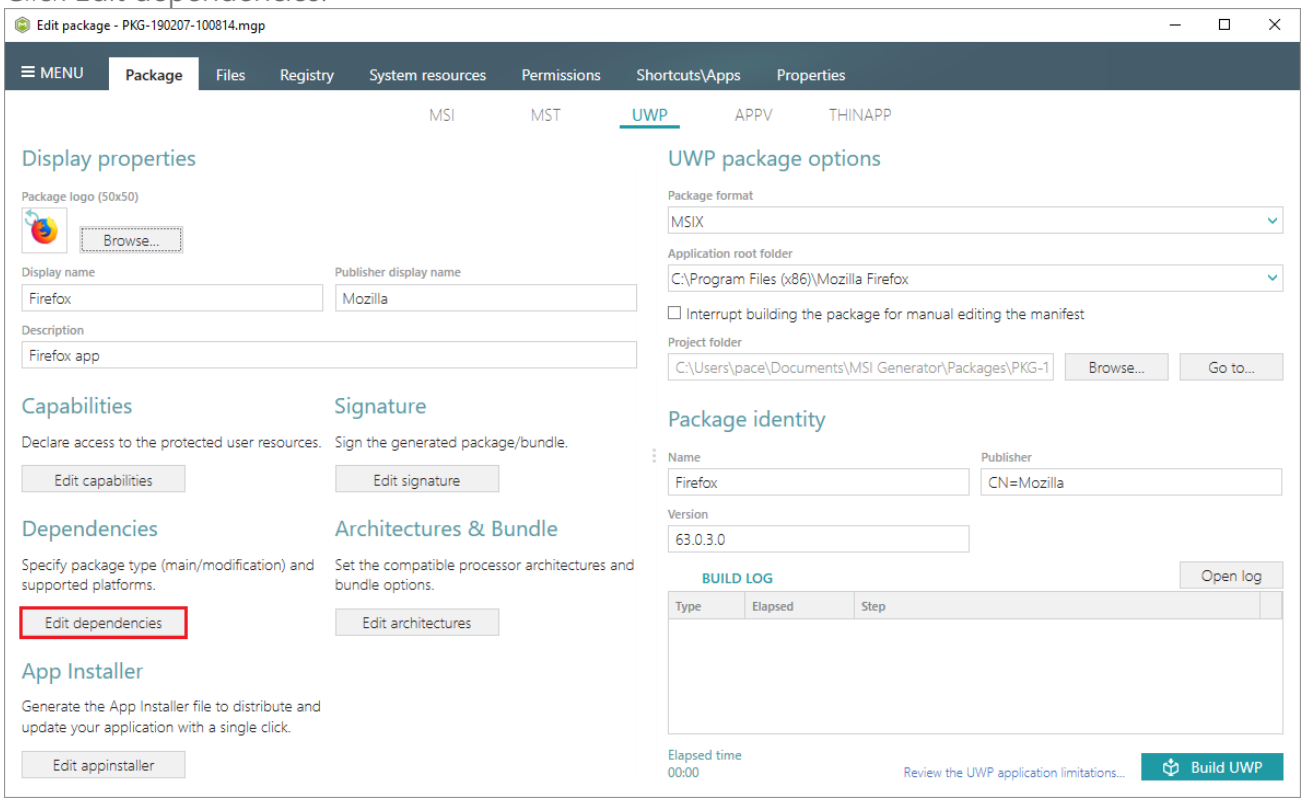
3.8.7.5 Set Supported Platforms

Set the supported platforms (target device families) on which, the UWP package could be run. For information, see Use the MSIX SDK to build a package for cross-platform use.

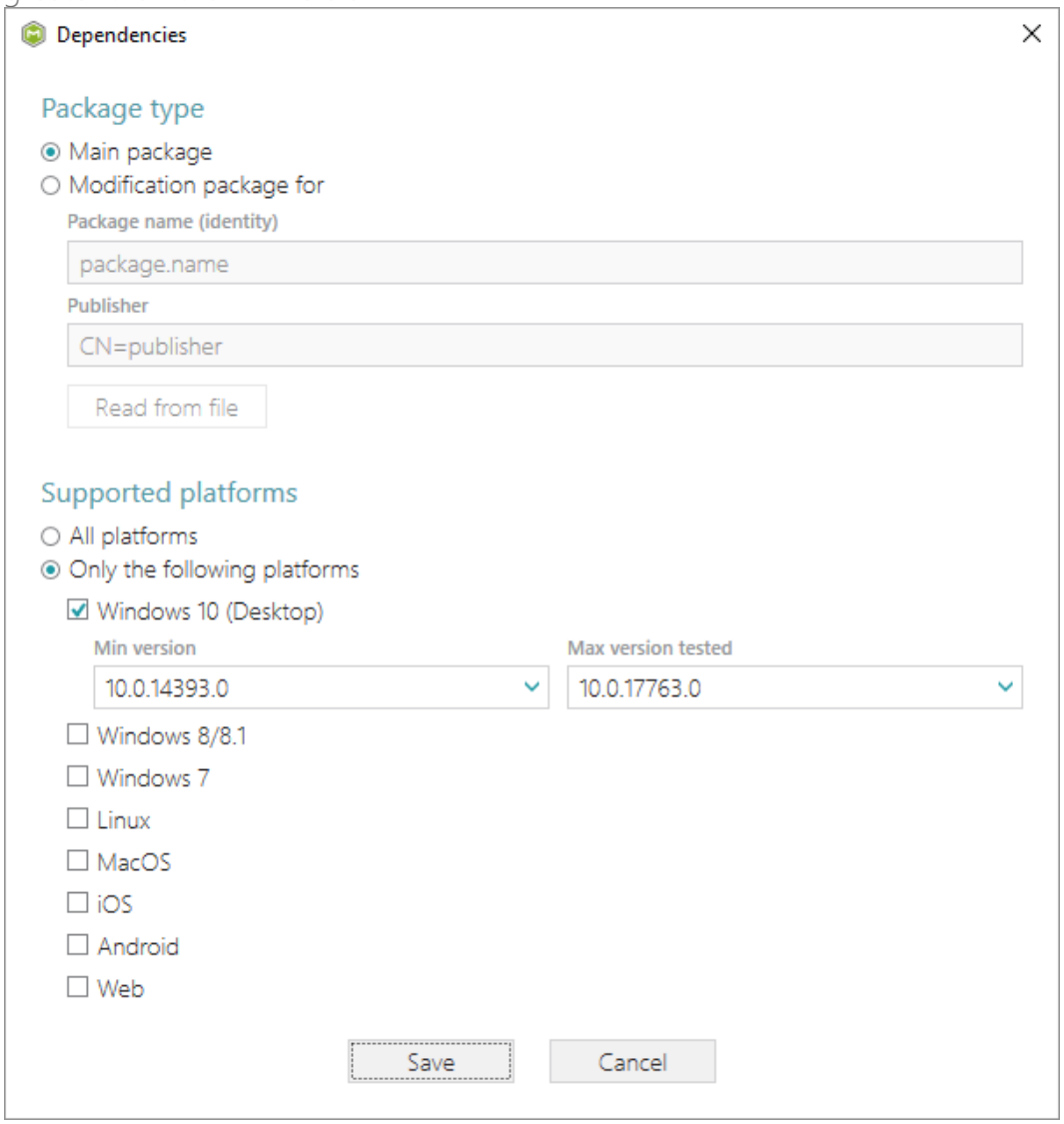
- Go to the Package -> UWP tab.



- Click Edit dependencies.



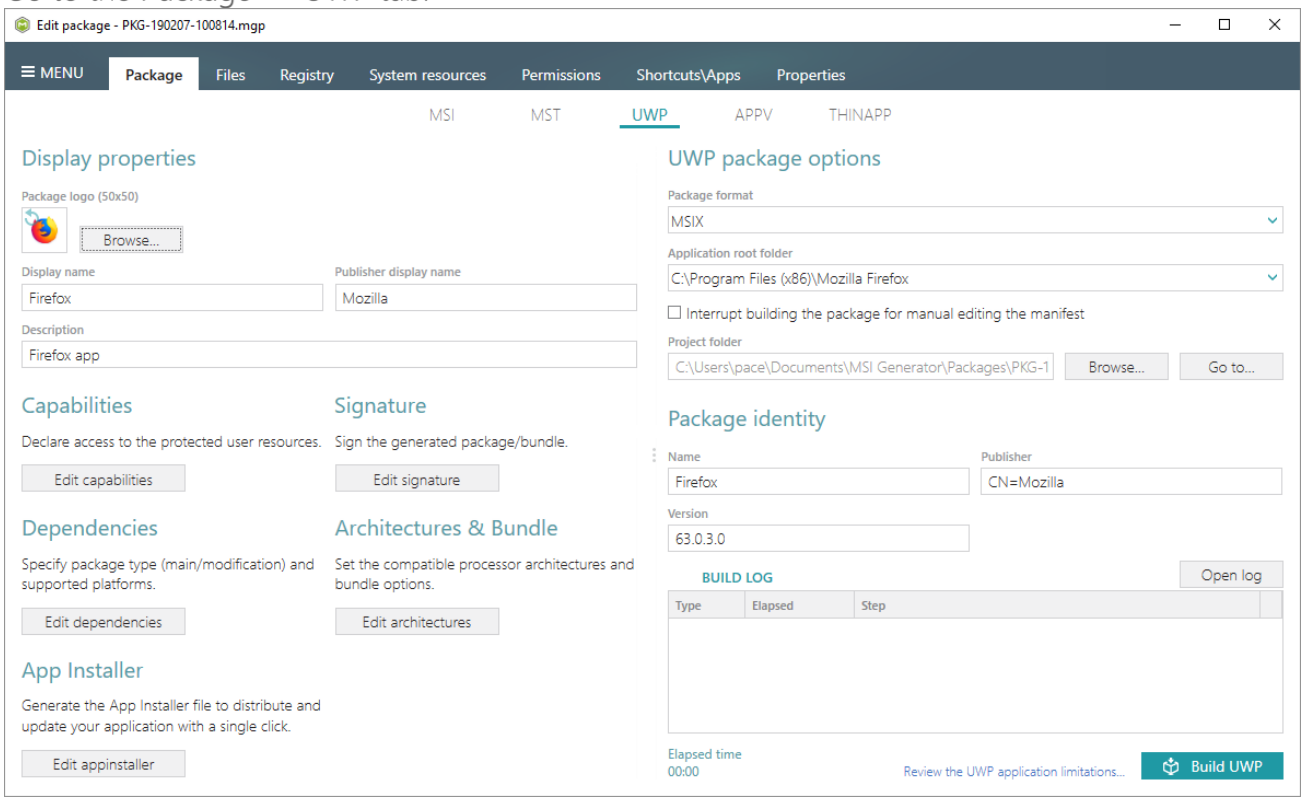
- Select the necessary supported platforms and click Save.
  NOTE Only on Windows 10 platforms, the package will only be deployed on OS versions greater than the MinVersion.



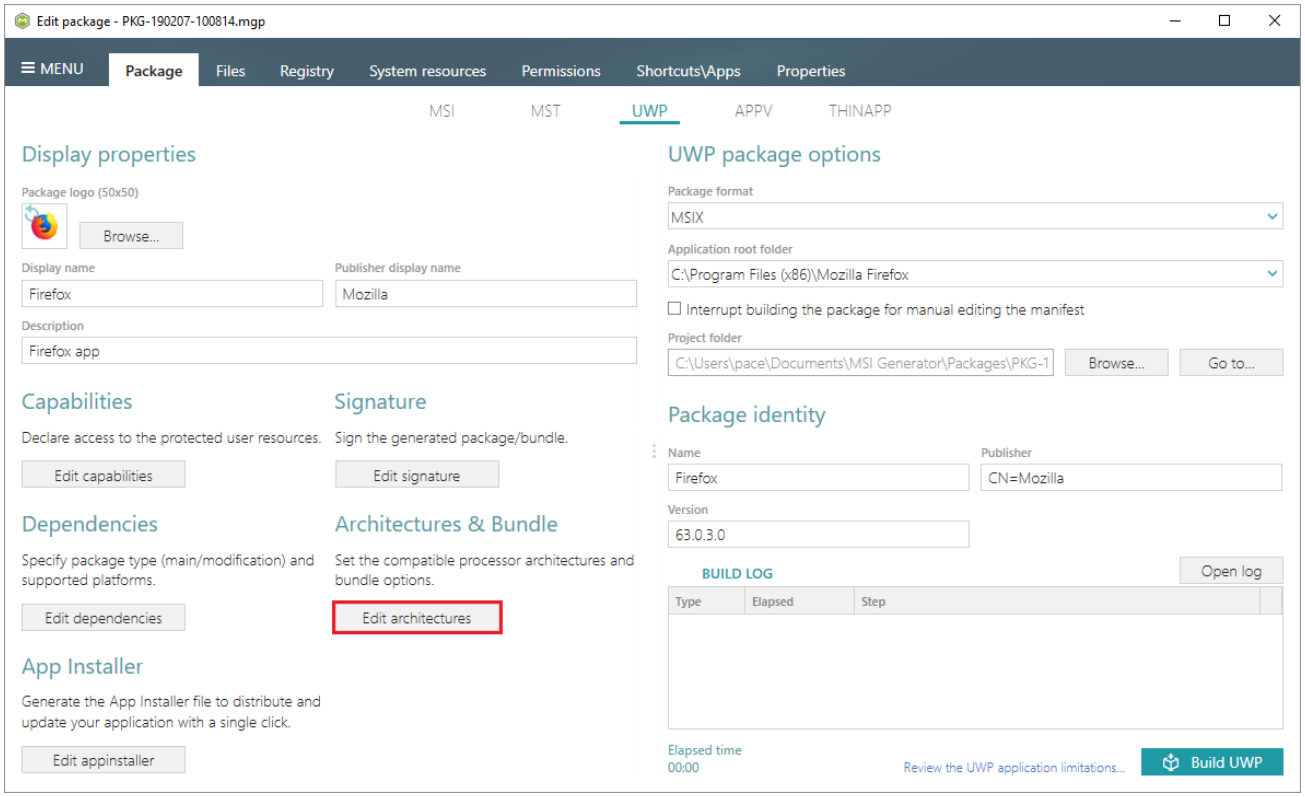
3.8.7.6 Manage Package Architectures

Select the processor architectures (x86, x64, ARM) on which you want to run your UWP package. For each selected processor architecture, a separate package will be created. For information, see App package architectures.

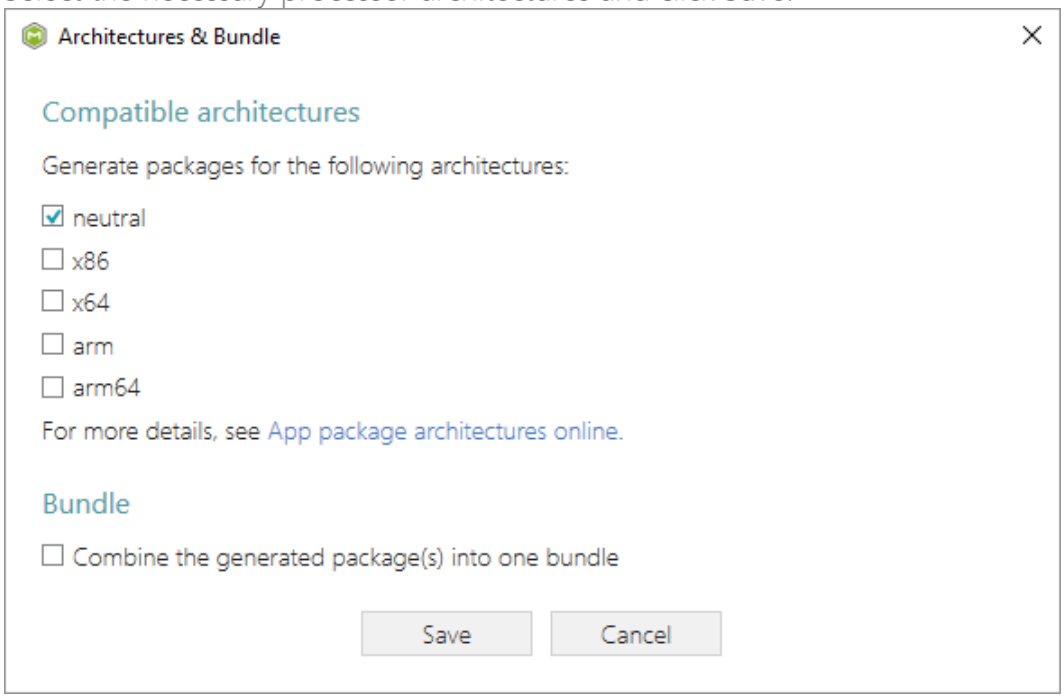
- Go to the Package -> UWP tab.



- Select Edit architectures.



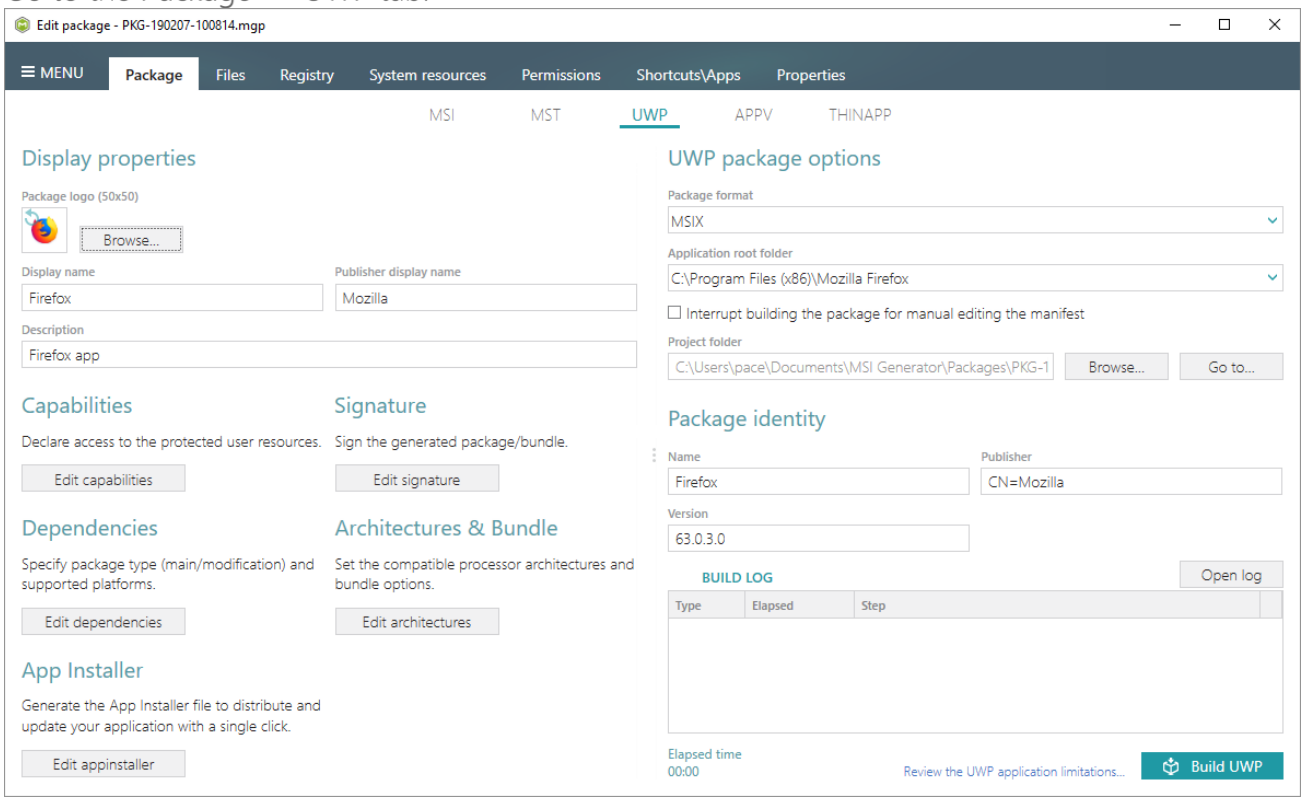
- Select the necessary processor architectures and click Save.



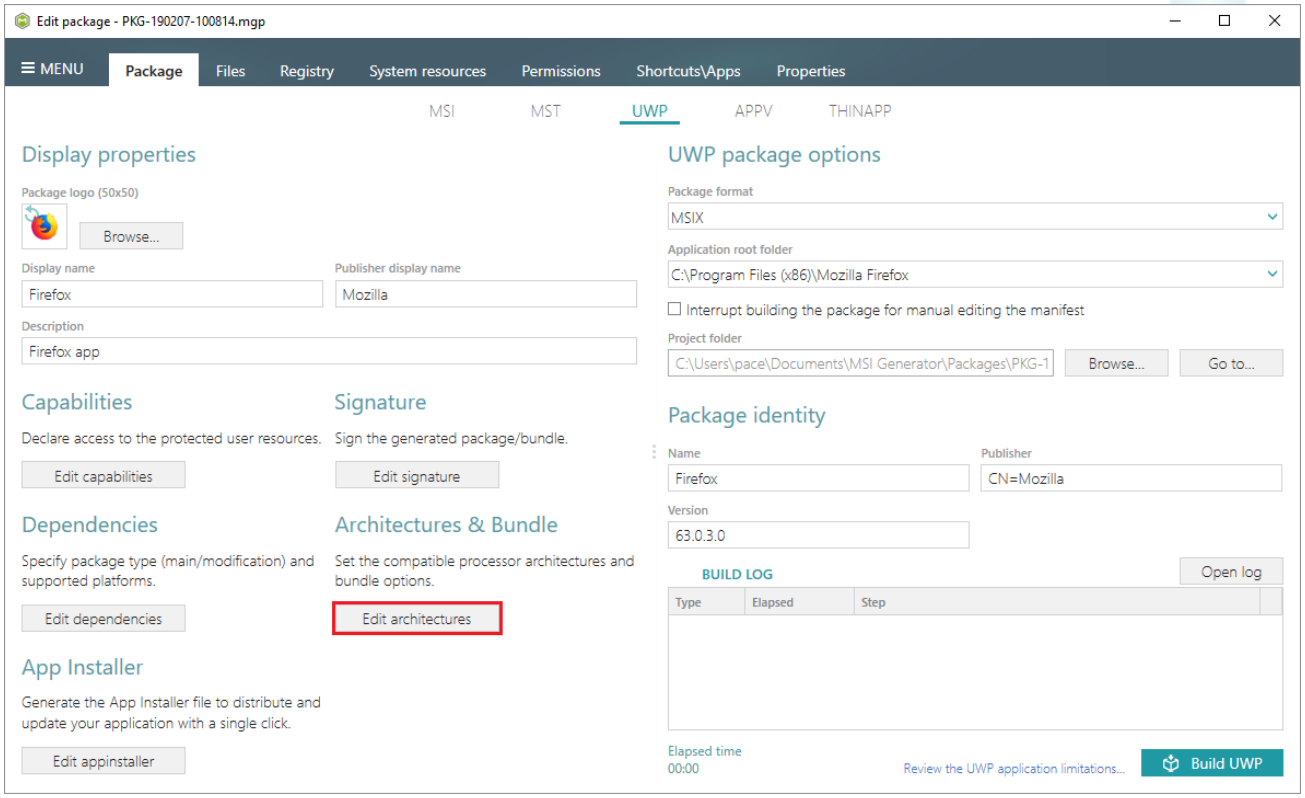
3.8.7.7 Create Bundle

Combine the generated UWP packages into one single bundle. By bundling packages of different processor architecutres into one entity, only the bundle needs to be uploaded to the Store or another distribution location. The Windows 10 deployment platform is aware of the bundle package type and will only download the files that are applicable for your device's architecture. Keep in mind that if you decide to distribute a bundle for a particular app, you cannot revert back to distributing just an UWP package.

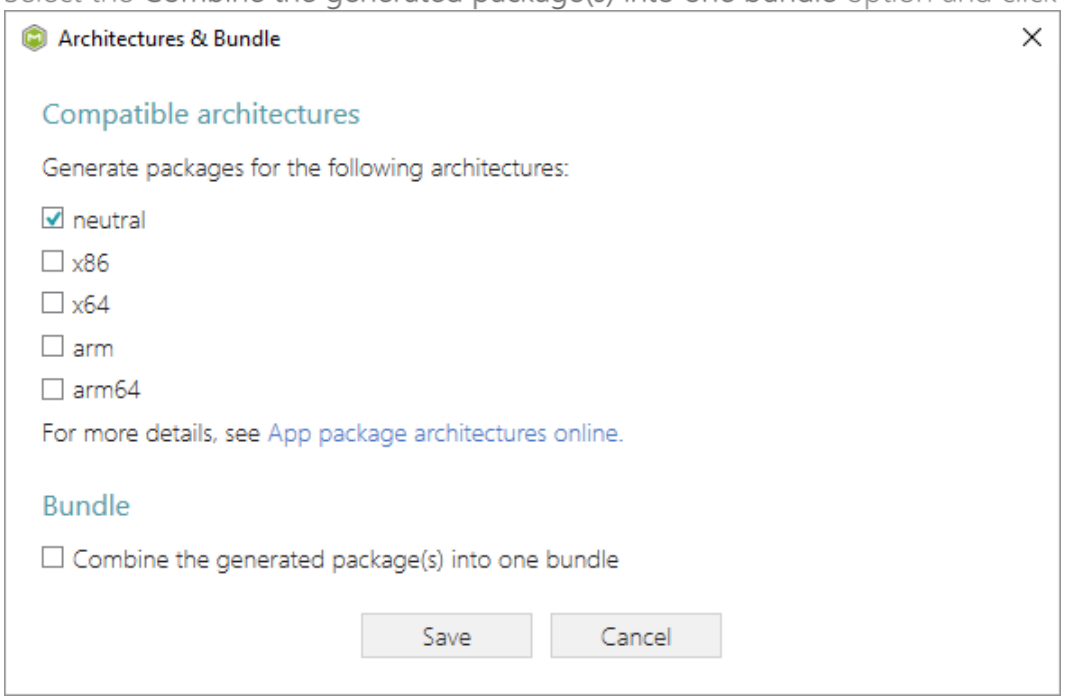
- Go to the Package -> UWP tab.



- Select Edit architectures.



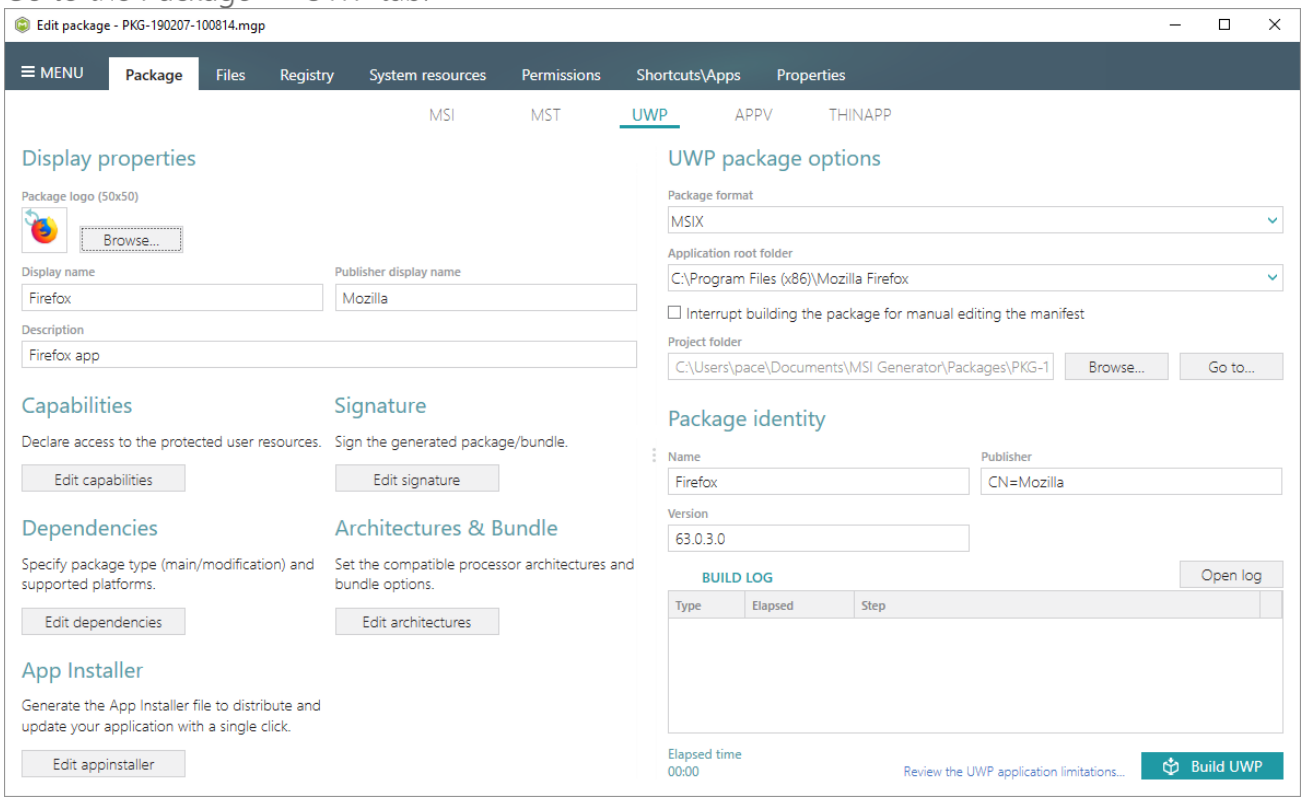
- Select the Combine the generated package(s) into one bundle option and click Save.



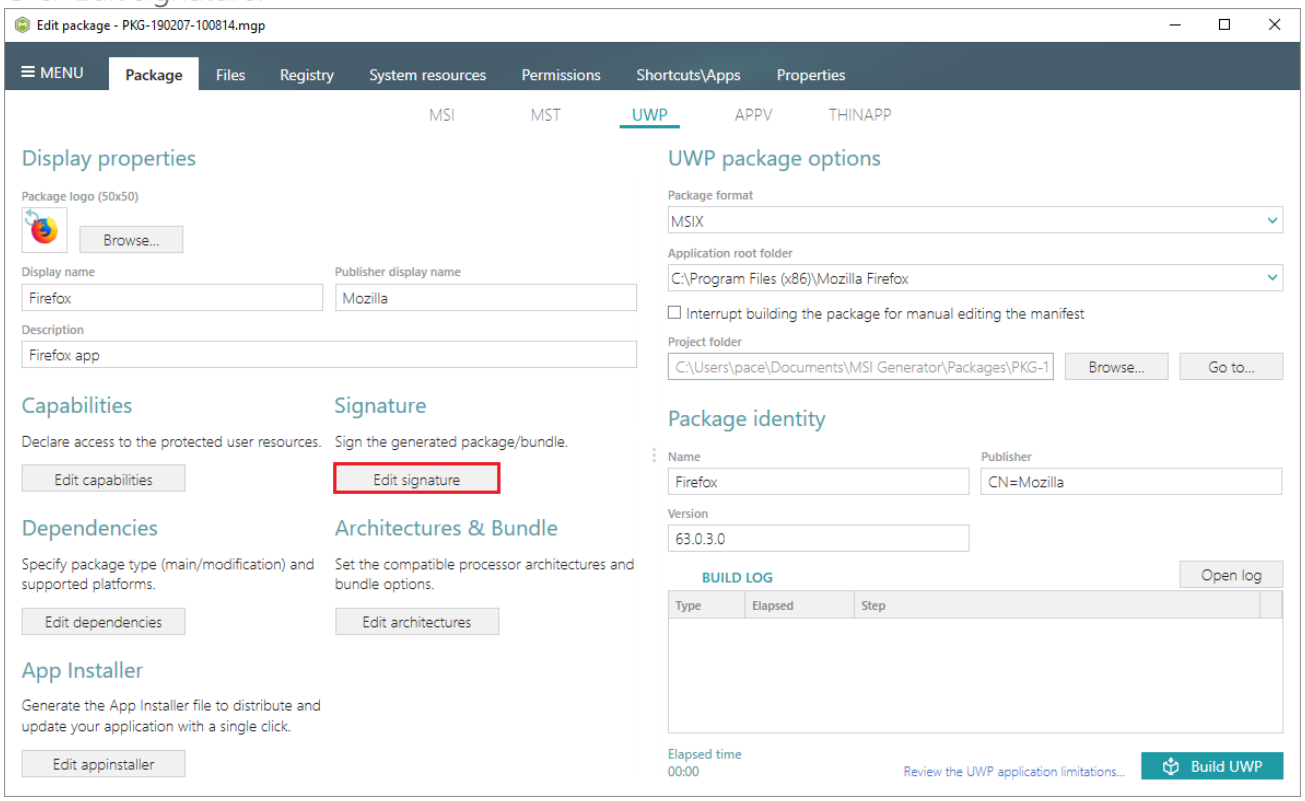
3.8.7.8 Sign Package/Bundle

Sign the generated package or bundle with a certificate for the testing purposes or distribution. Signing an app package provides the user with verification that the app's data has not been modified after it was signed while also confirming the identity of the user or company that signed it.

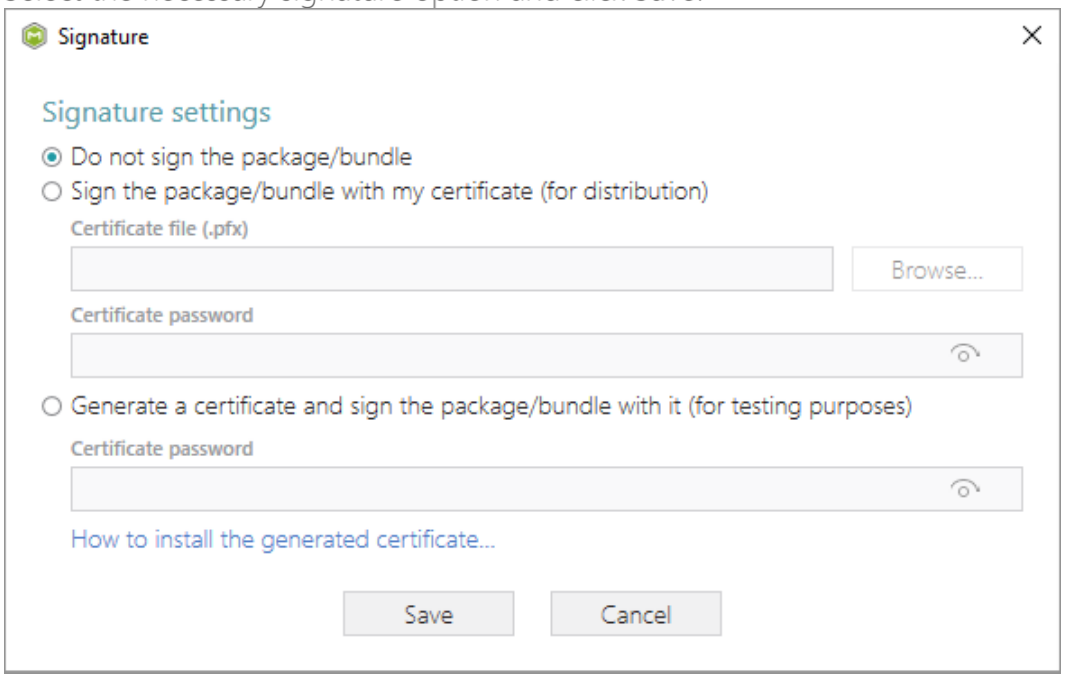
- Go to the Package -> UWP tab.



- Click Edit signature.



- Select the necessary signature option and click Save.

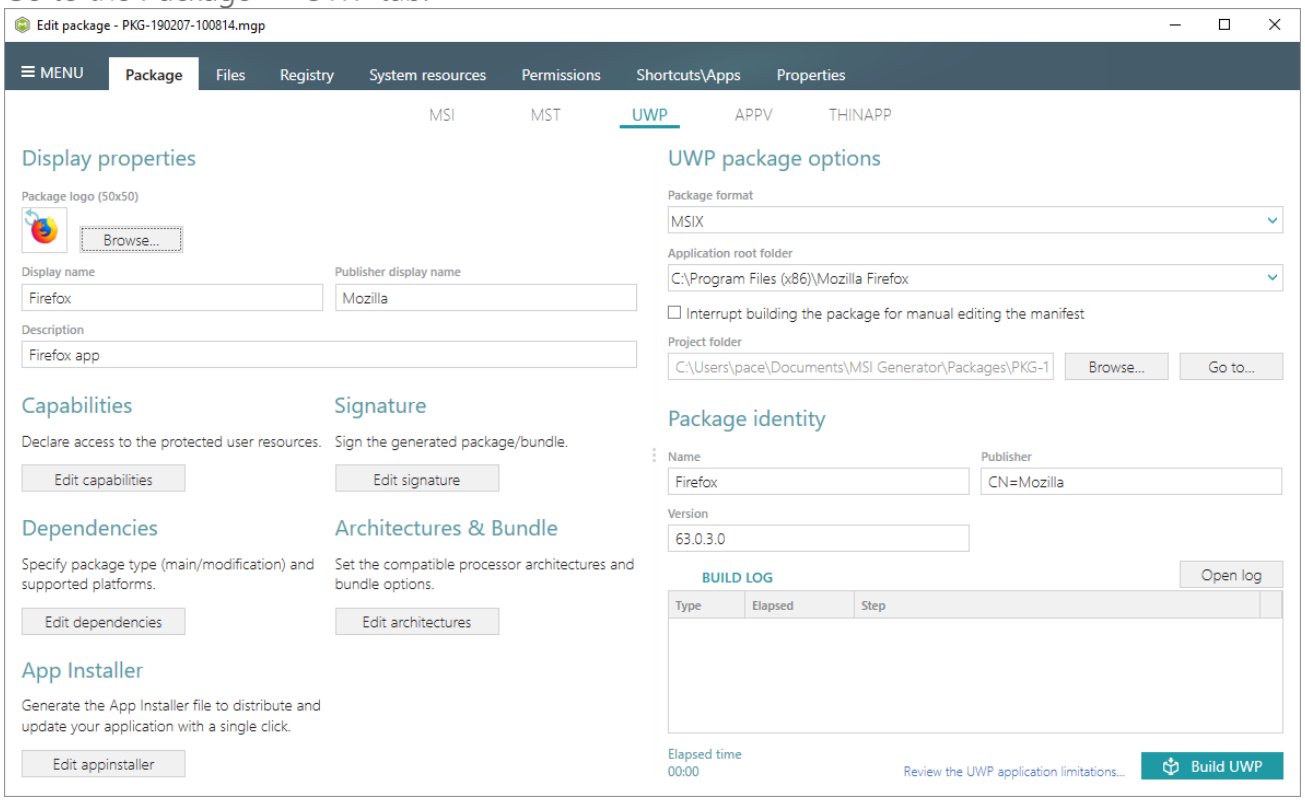


- To install the generated UWP package, (1) enable the Developer Mode or Sideloading apps on the target system and (2) ensure that the certificate, used for signing, is trusted on that system. If you are using the self signed certificate, install it manually to the Trusted People or Trusted Publishers local machine certificate store. By adding a certificate to local machine certificate stores, you affect the certificate trust of all users on the computer. It is recommended that you remove those certificates when they are no longer necessary to prevent them from being used to compromise system trust.

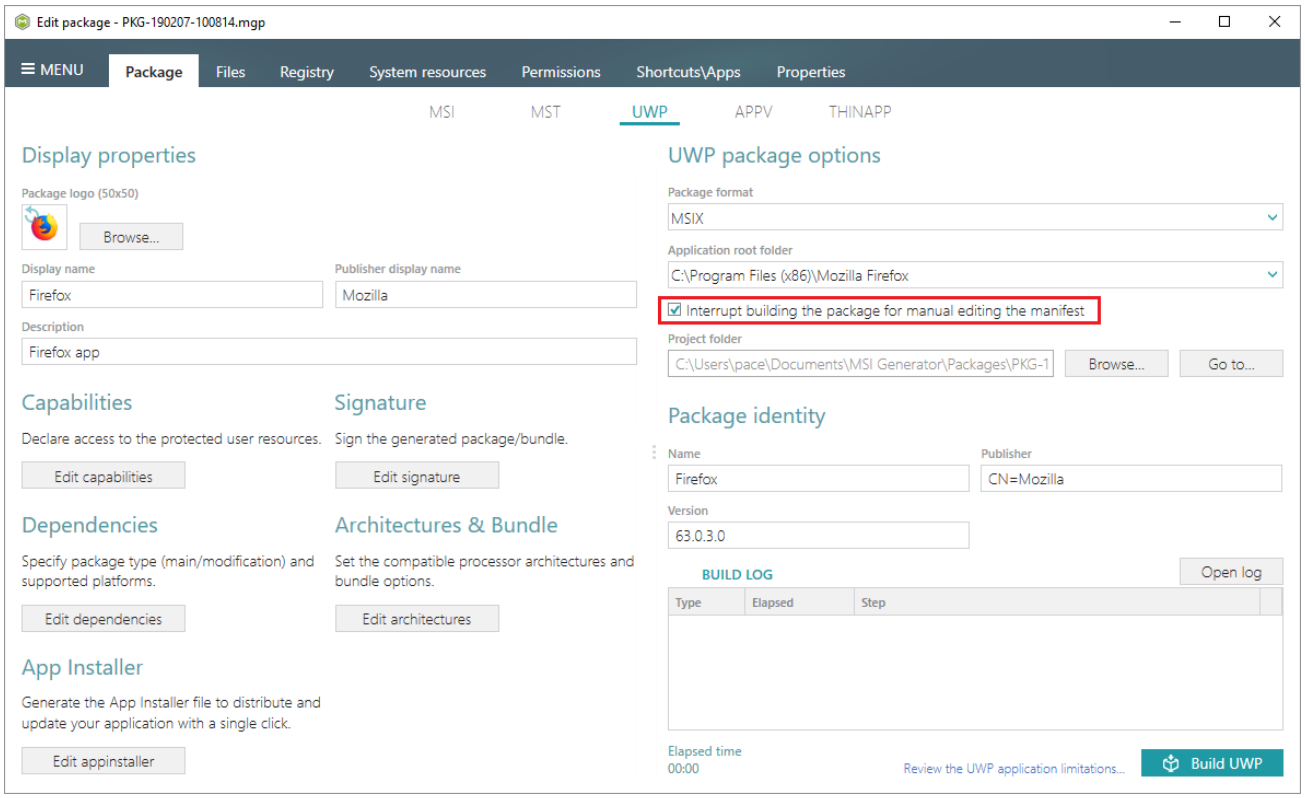
3.8.7.9 Edit Manifest Manually

Edit the package manifest file(s) manually to modify existing or add new specific package settings.

- Go to the Package -> UWP tab.

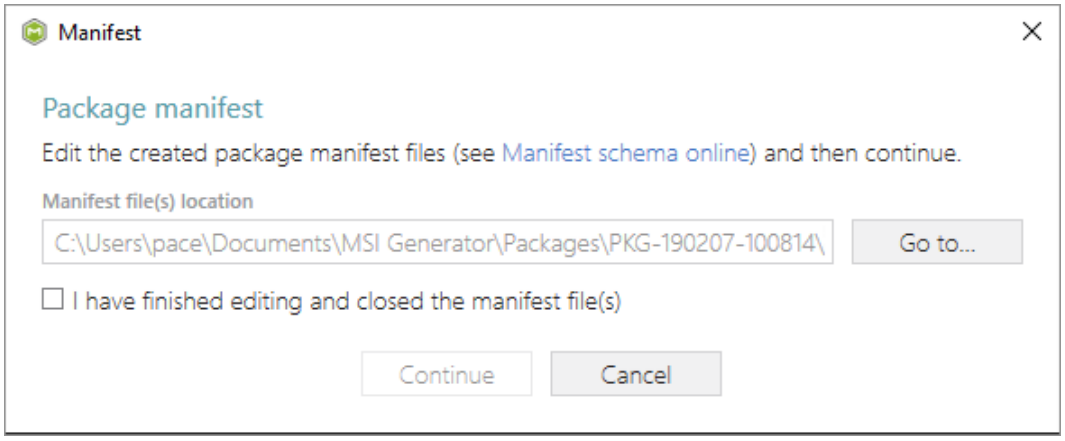


- Tick the Interrupt building the package for manual editing the manifest option and click Build UWP.

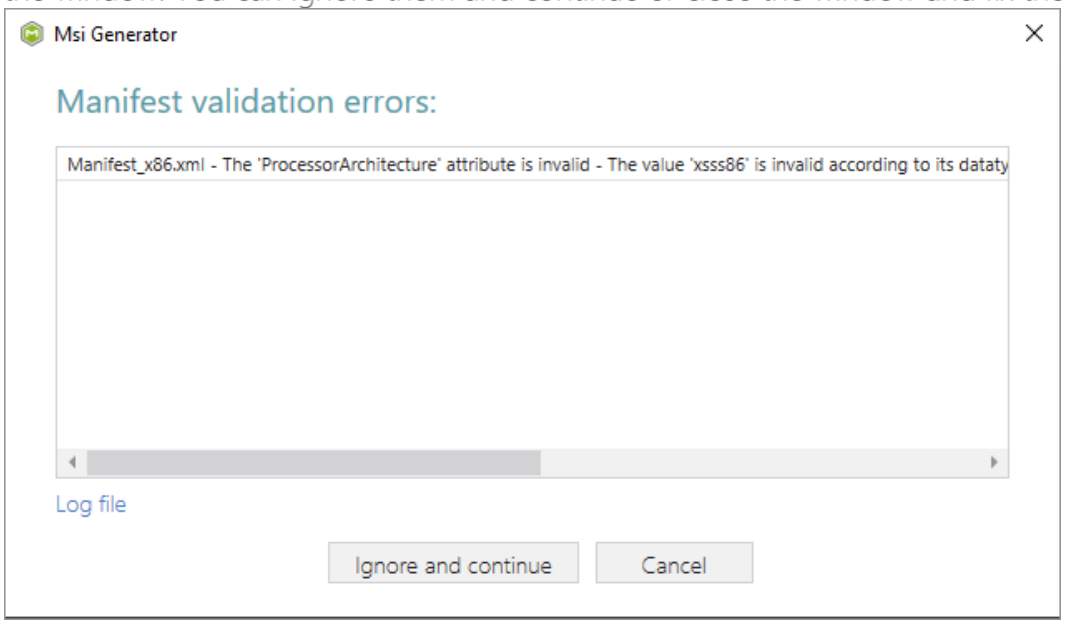


- In the opened window, click Go to... to locate the prepared manifest files on the file system. Edit them according to your needs using any text editor, save and close their folder. Afterwards, tick the I have finished editing and closed the manifest file(s) option and click

Continue.



- MSI Generator validates the edited manifest files against the schemas and displays all errors in the window. You can ignore them and continue or close the window and fix the manifest files.



3.8.7.10 Manage Apps and Visual Assets

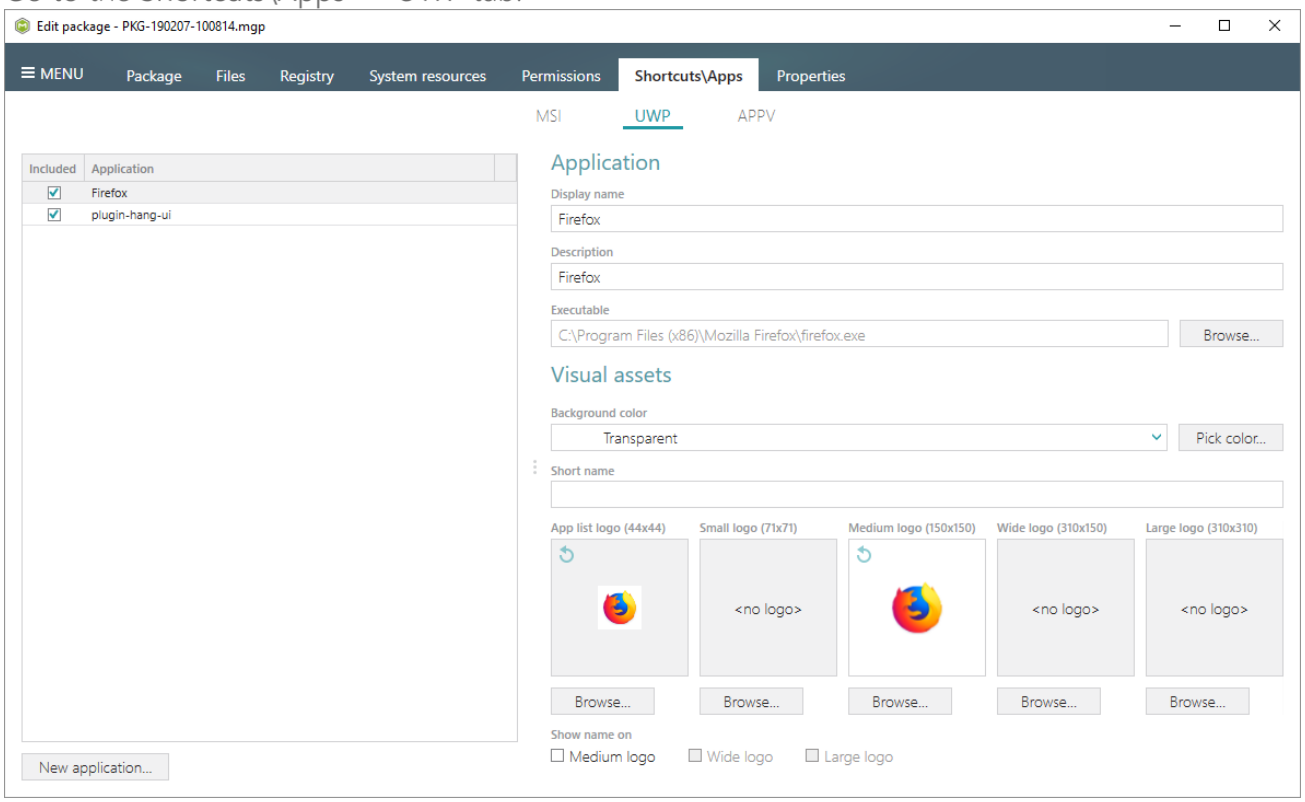
Choose a scenario that better suits your needs:

- Exclude App, described in section 3.8.7.10.1
  Exclude the unnecessary app from the package.
- Include Excluded App, described in section 3.8.7.10.2
  Include, the previously excluded app, back to the package.
- Add App, described in section 3.8.7.10.3
  Add a new app to the package.
- Set App Logos, described in section 3.8.7.10.4
  Control appearance settings of the app such as logos, name, background color.

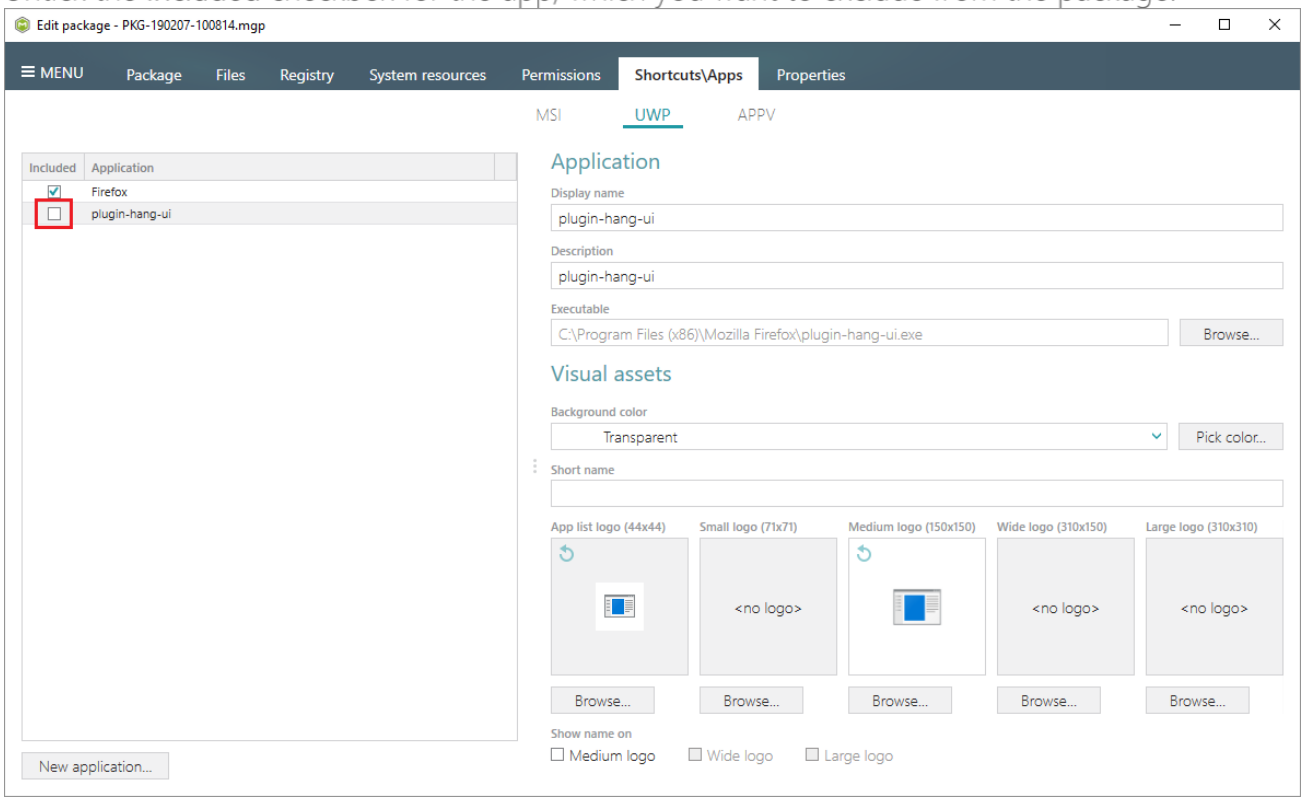
3.8.7.10.1 Exclude App

Exclude the unnecessary app from the package.

- Go to the **Shortcuts\Apps** -> **UWP** tab.



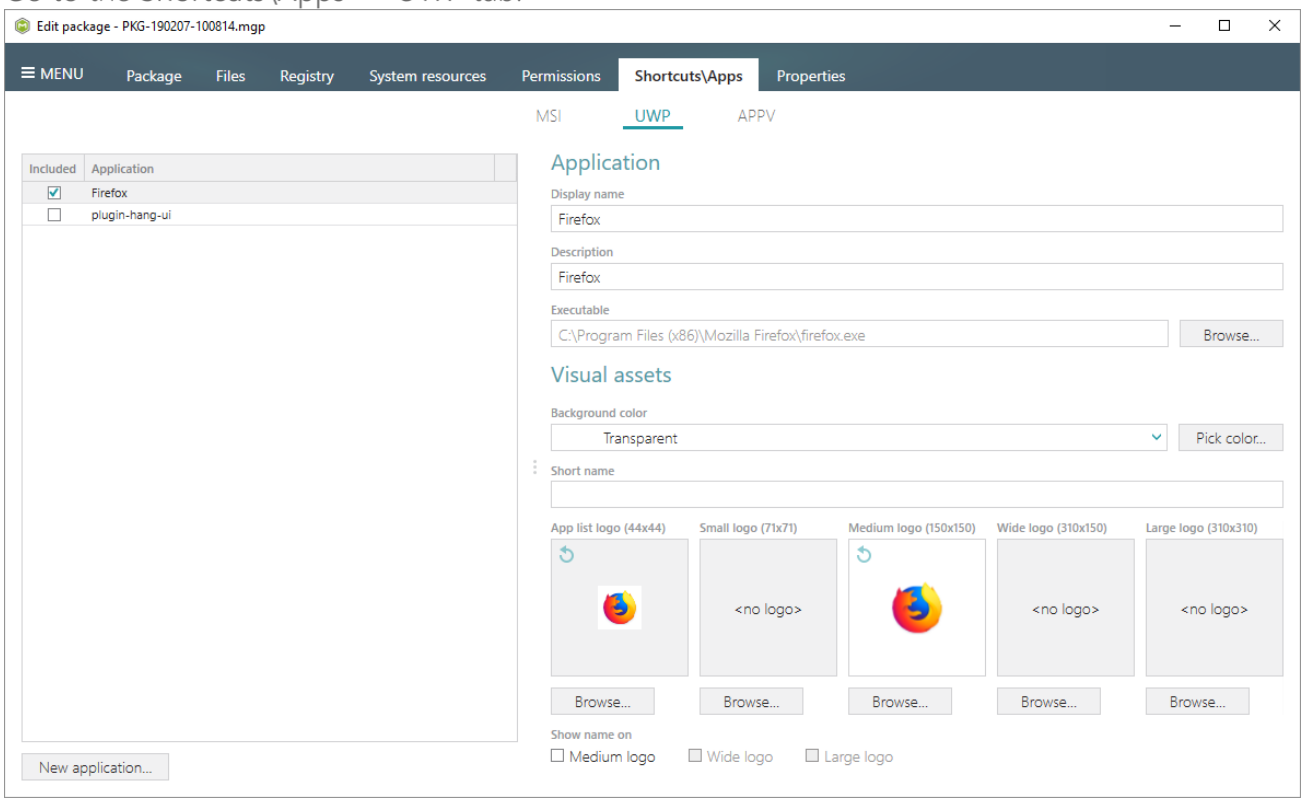
- Untick the **Included** checkbox for the app, which you want to exclude from the package.



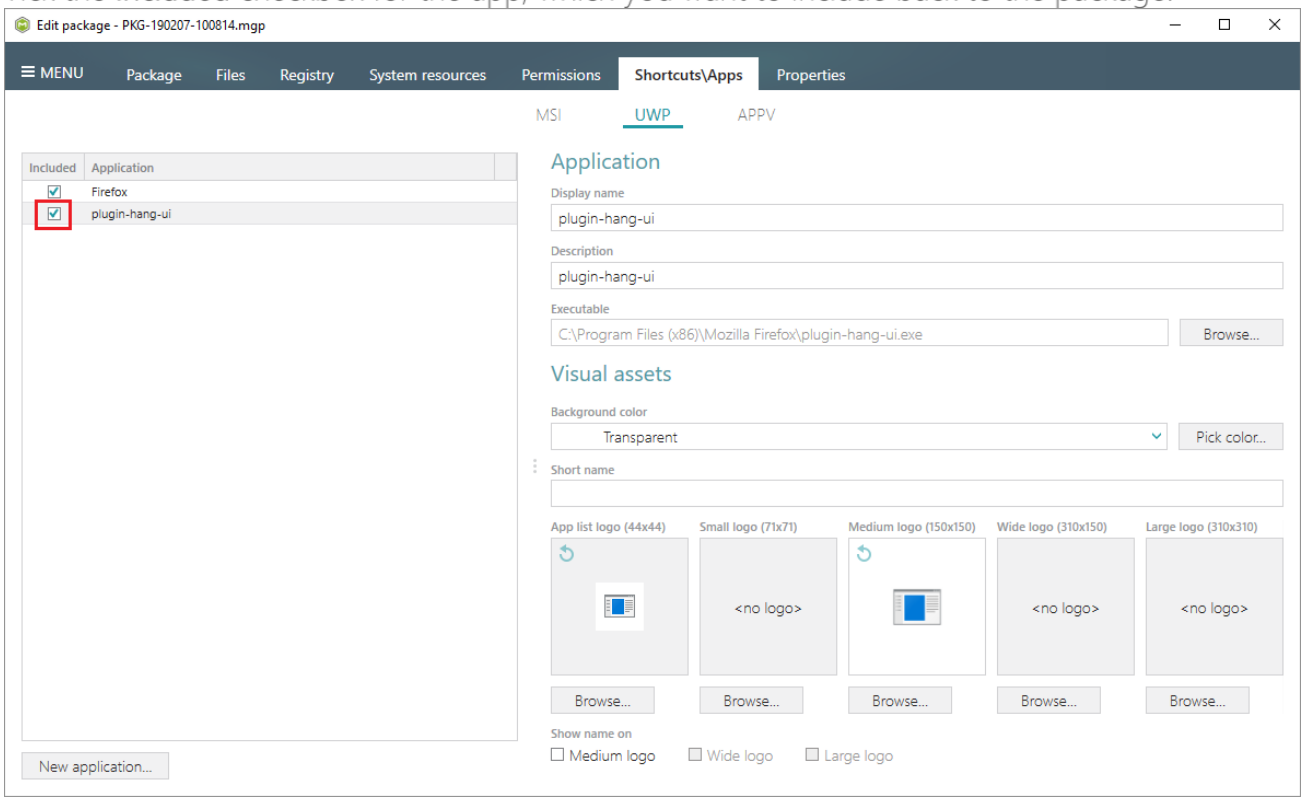
3.8.7.10.2 Include Excluded App

Include, the previously excluded app, back to the package.

- Go to the **Shortcuts\Apps** -> **UWP** tab.



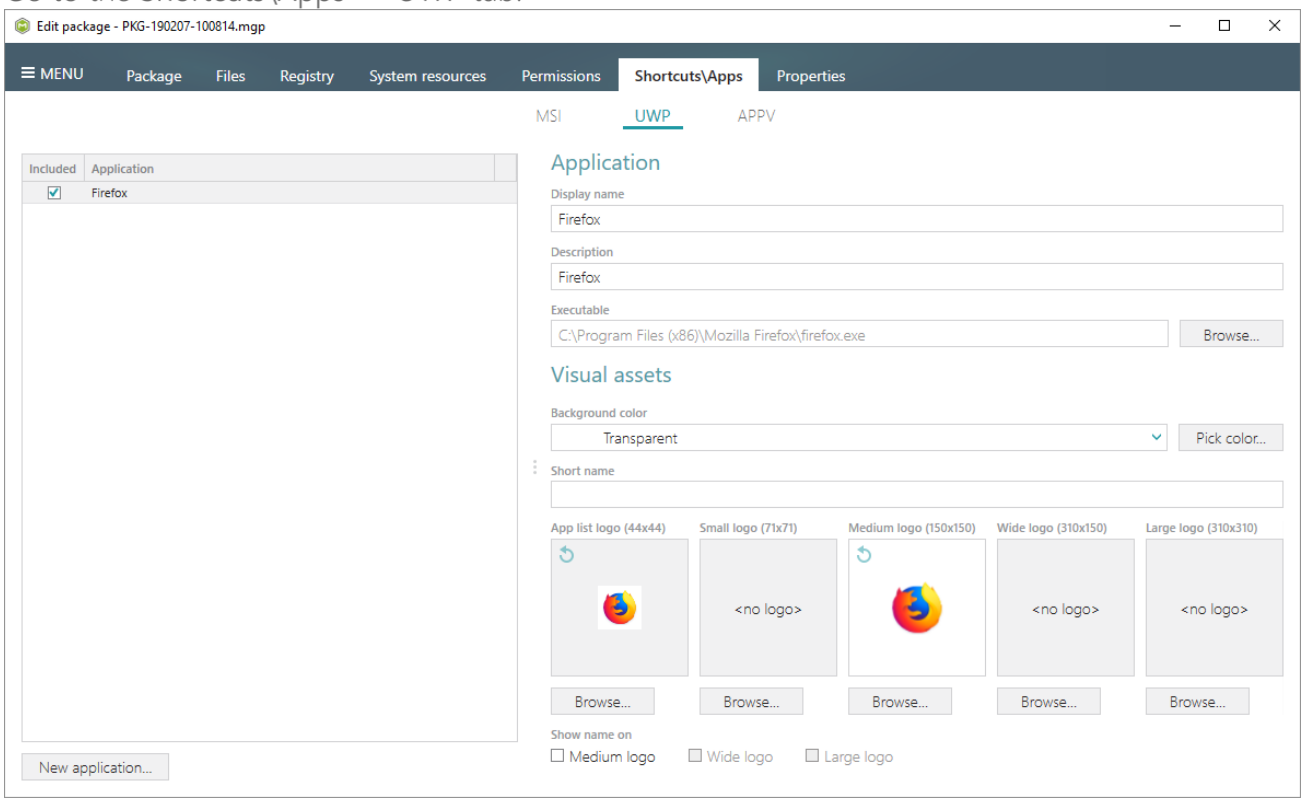
- Tick the **Included** checkbox for the app, which you want to include back to the package.



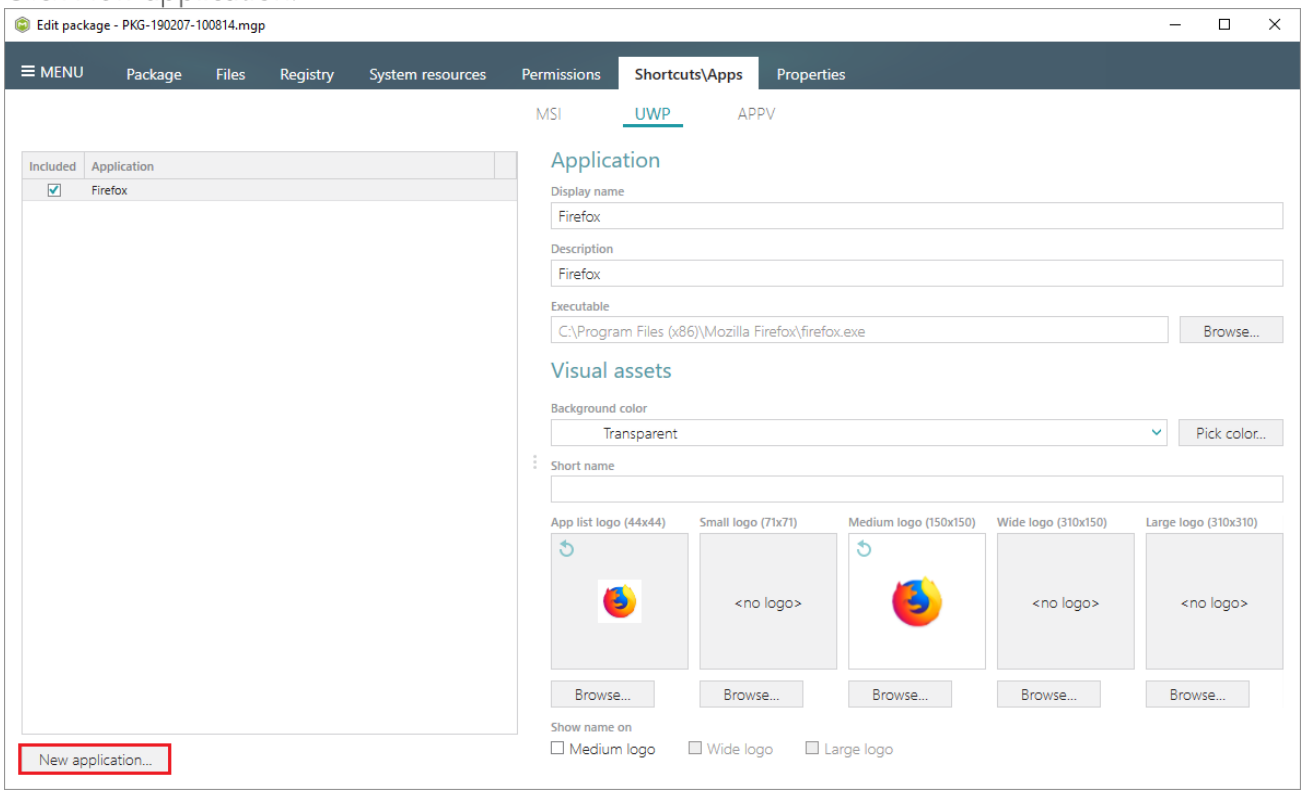
3.8.7.10.3 Add App

Add a new app to the package.

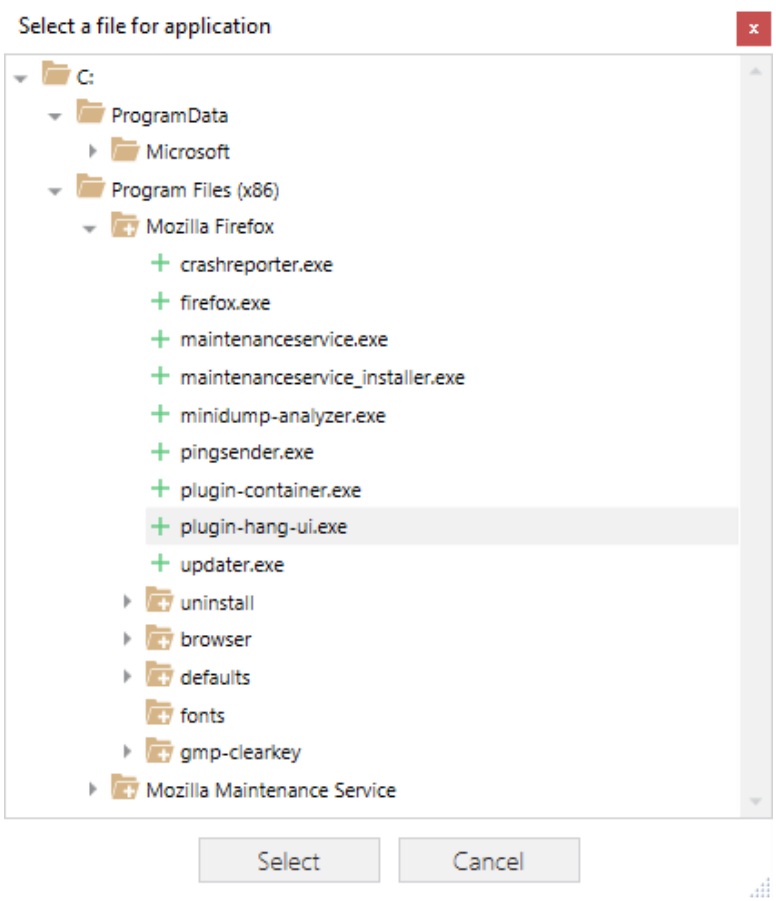
- Go to the Shortcuts\Apps -> UWP tab.



- Click New application.



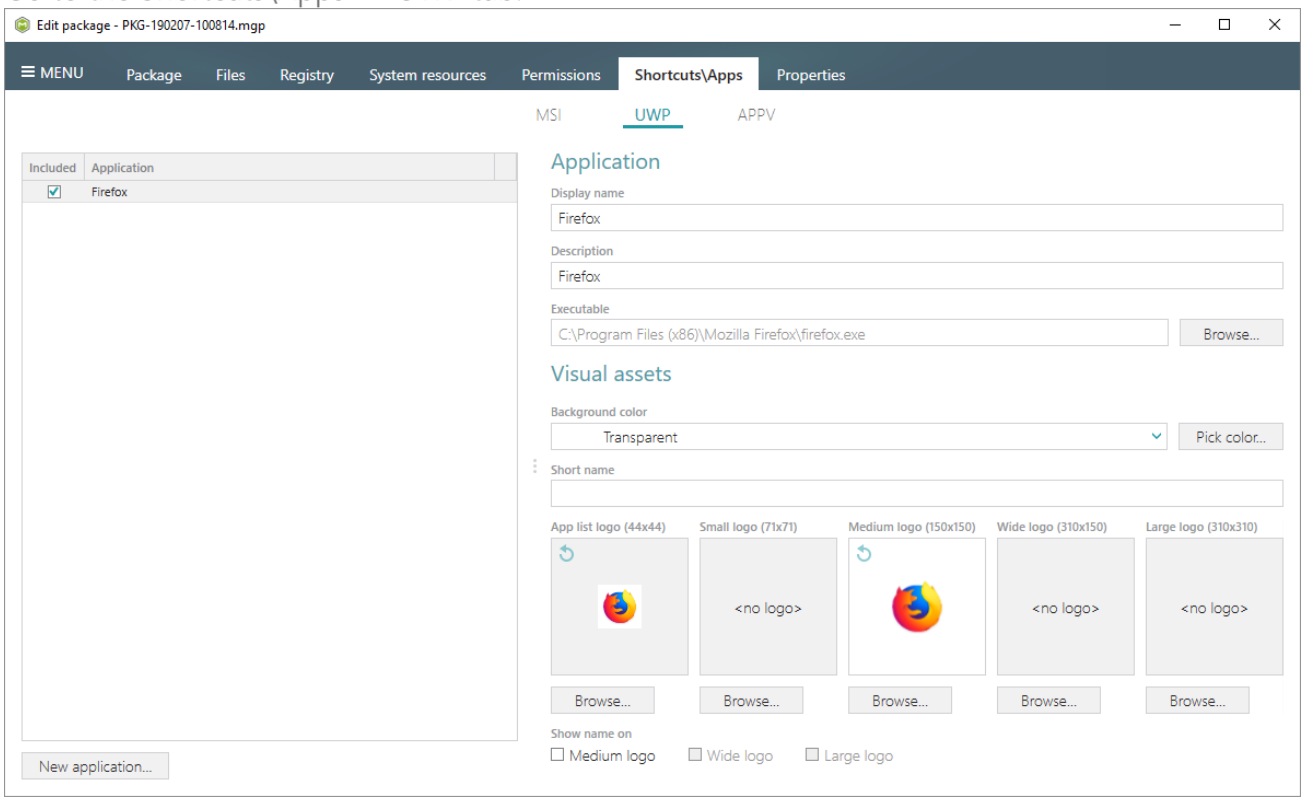
- Select an executable from the project resources for your new application and click Select.



3.8.7.10.4 Set App Logos

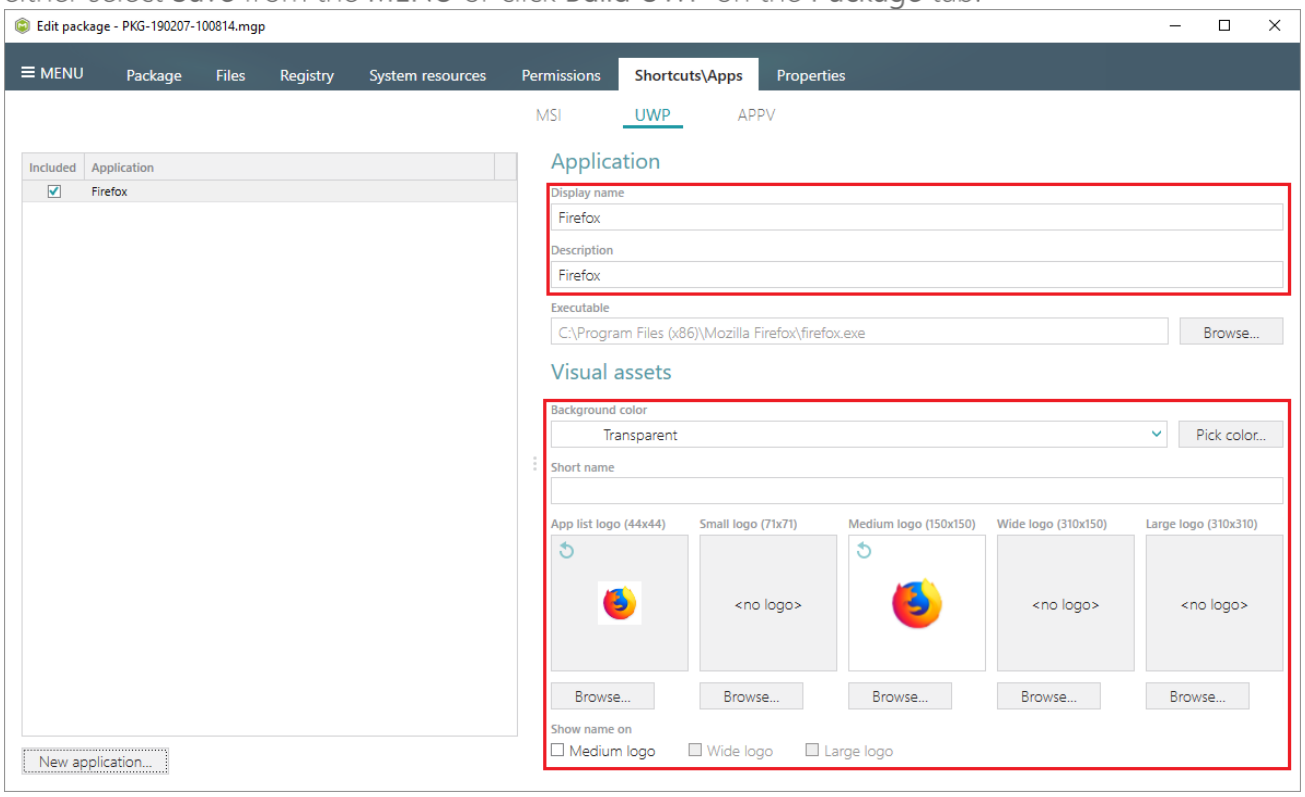
Control appearance settings of the app such as logos, name, background color.

- Go to the Shortcuts\Apps -> UWP tab.



- Update the necessary settings for the selected application such as display name, description, background color, logos, which will be displayed in the Start menu and so on.

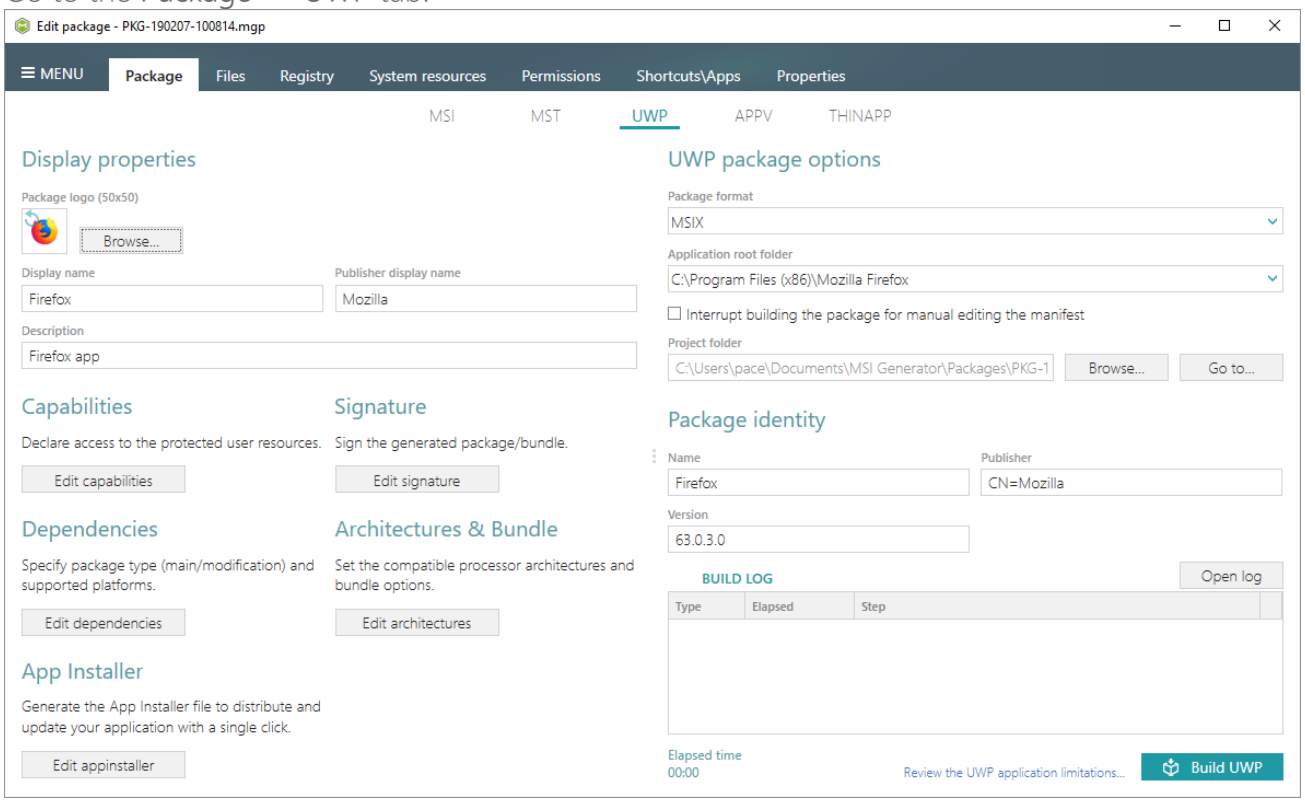
  NOTE The settings are applied once they are changed. To save these changes to the project either select Save from the MENU or click Build UWP on the Package tab.



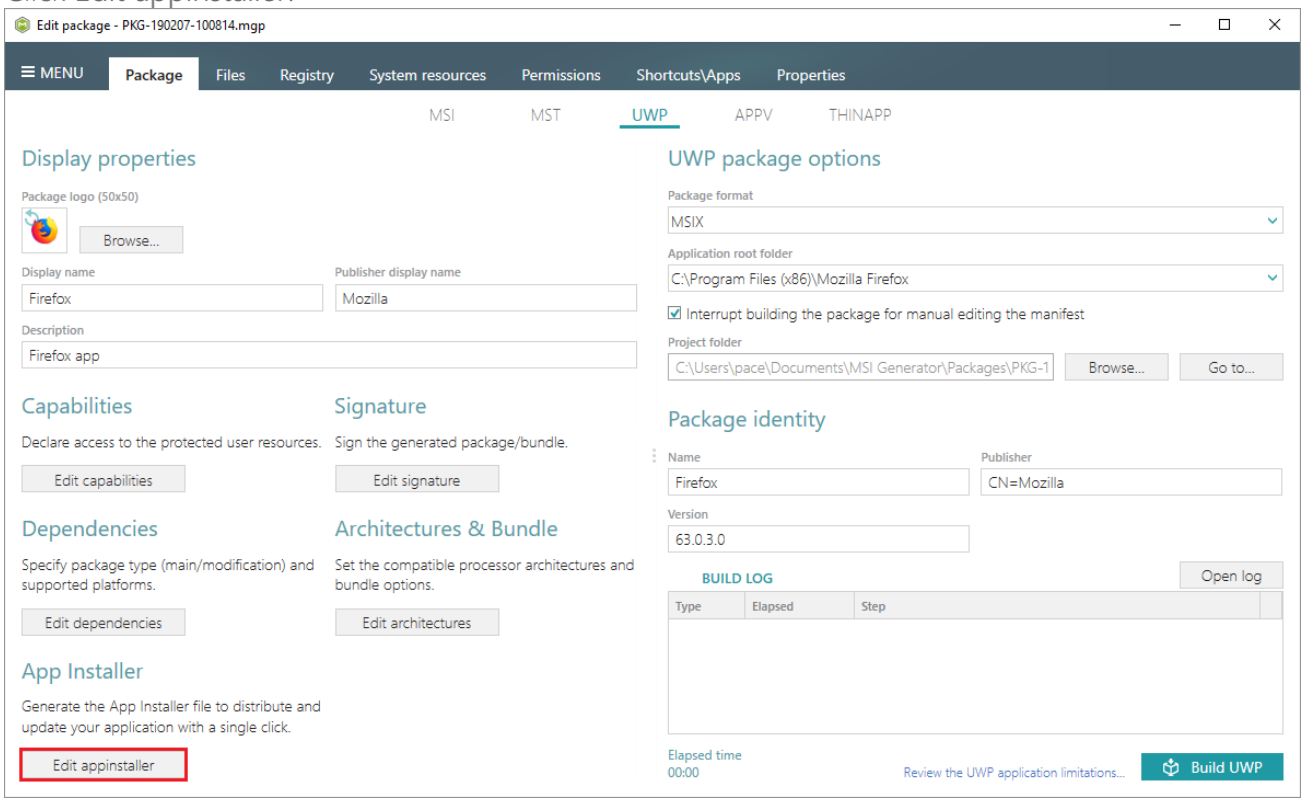
3.8.7.11 Generate AppInstaller

Generate the AppInstaller file for the non-Store distribution and automatic update of your UWP package. For information, see Install UWP apps with App Installer.

- Go to the Package -> UWP tab.

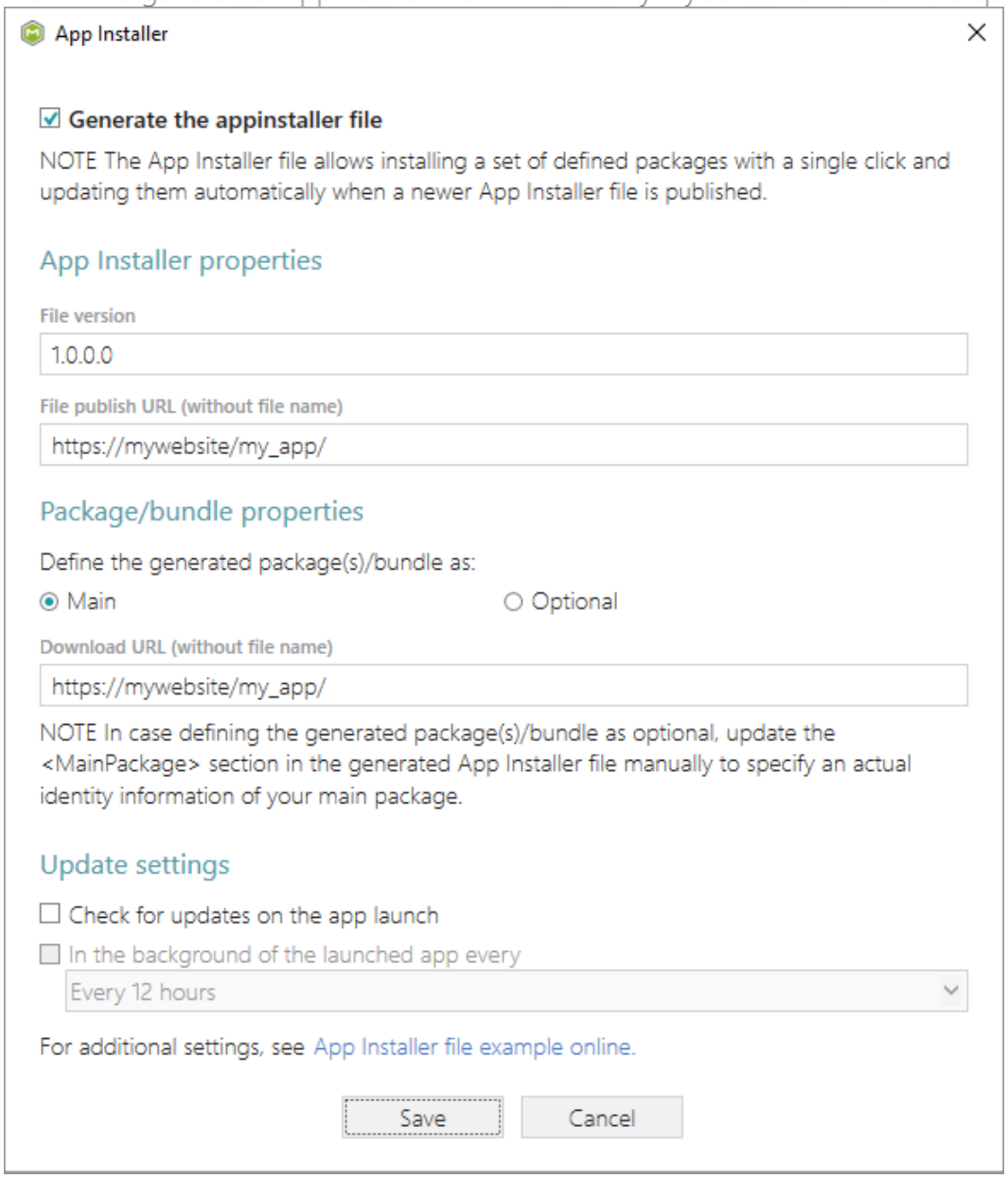


- Click Edit appinstaller.



- Tick the Generate the appinstaller file option, specify a file version (increase version for each new file), URL where this AppInstaller file will be published. Also, specify the generated package/bundle as main or as optional, and the URL, where this package/bundle will be

available for download. Select the desired package update settings and click Save.
For manual editing, see App Installer file (.appinstaller) schema reference.
NOTE The generated AppInstaller file will work only if you launch it from the specified URL.



- To install the generated UWP package with the AppInstaller file, (1) enable the Developer Mode or Sideloading apps on the target system and (2) ensure that the certificate, used for signing, is trusted on that system. If you are using the self signed certificate, install it manually to the Trusted People or Trusted Publishers local machine certificate store. By adding a certificate to local machine certificate stores, you affect the certificate trust of all users on the computer. It is recommended that you remove those certificates when they are no longer necessary to prevent them from being used to compromise system trust.
- For troubleshooting, see Troubleshoot installation issues with the App Installer file.

3.8.8 APPV Settings

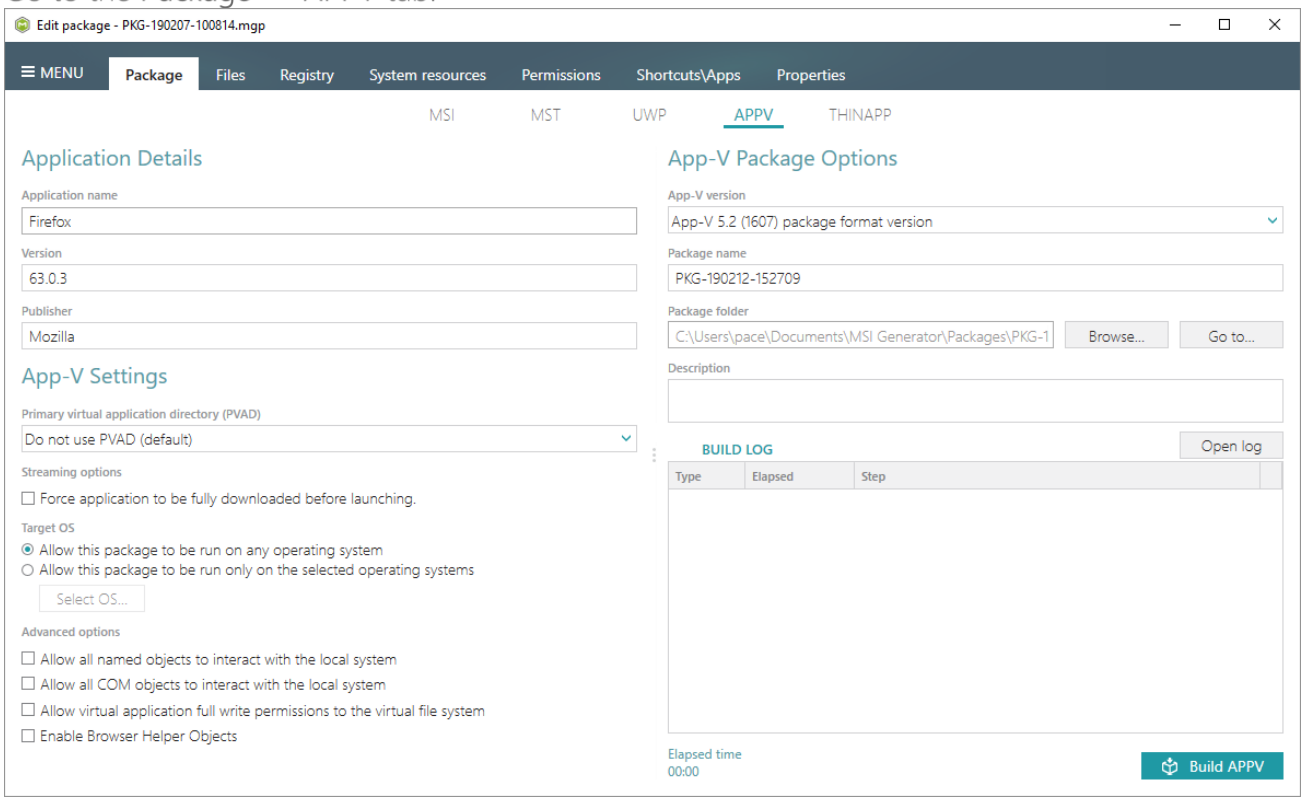
Choose a scenario that better suits your needs:

- Set Application Details, described in section 3.8.8.1
  Set the application details, such as app name, version and publisher of the APPV package.
- Manage Package Settings, described in section 3.8.8.2
  Select the package settings such as APPV version, Primary virtual application directory (PVAD), target OS, streaming and some advanced options.
- Set Folder Virtualization Level, described in section 3.8.8.3
  Change the default APPV virtualization (isolation) level for a folder.
- Set Registry Virtualization Level, described in section 3.8.8.4
  Change the default APPV virtualization (isolation) level for a registry key.
- Manage Apps, FTA, Shortcuts, described in section 3.8.8.5
  Include/exclude captured apps, add new ones, manage their shortcuts and FTA.

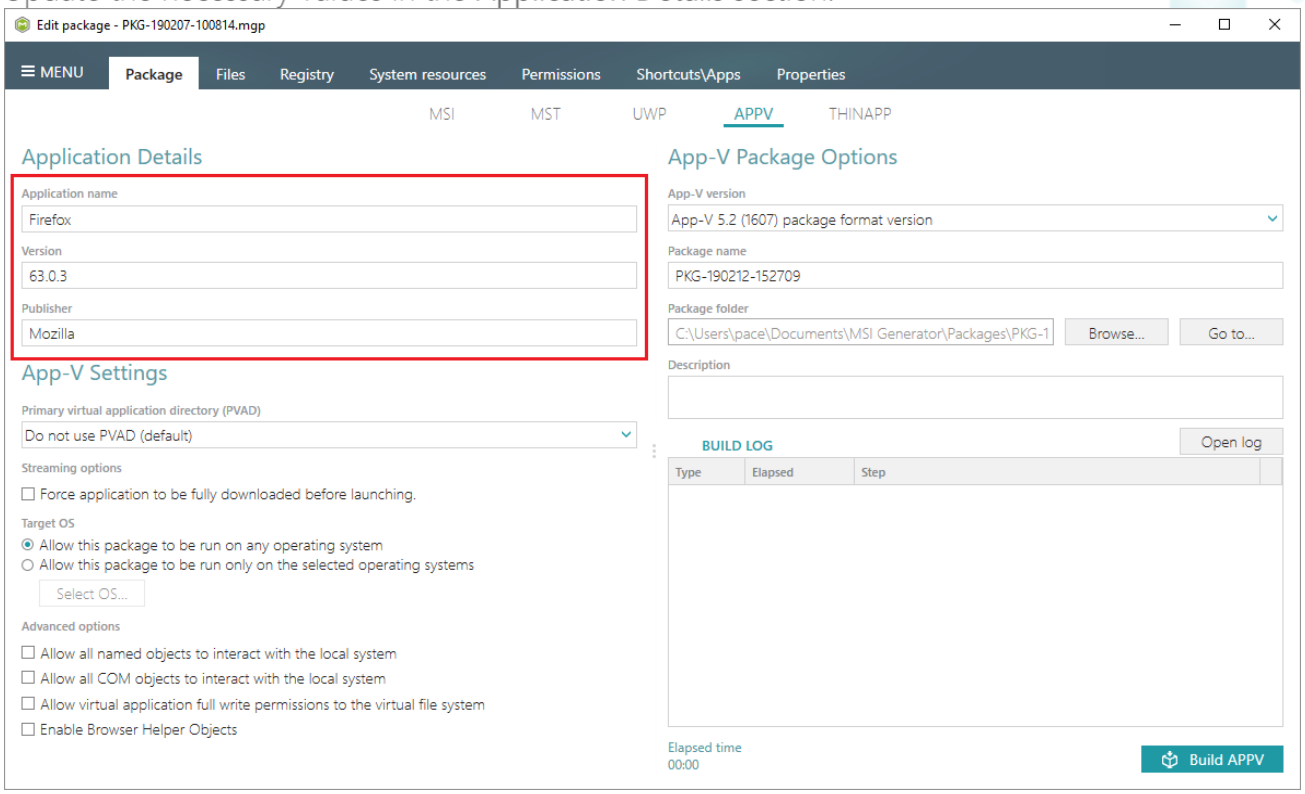
3.8.8.1 Set Application Details

Set the application details, such as app name, version and publisher of the APPV package.

- Go to the Package -> APPV tab.



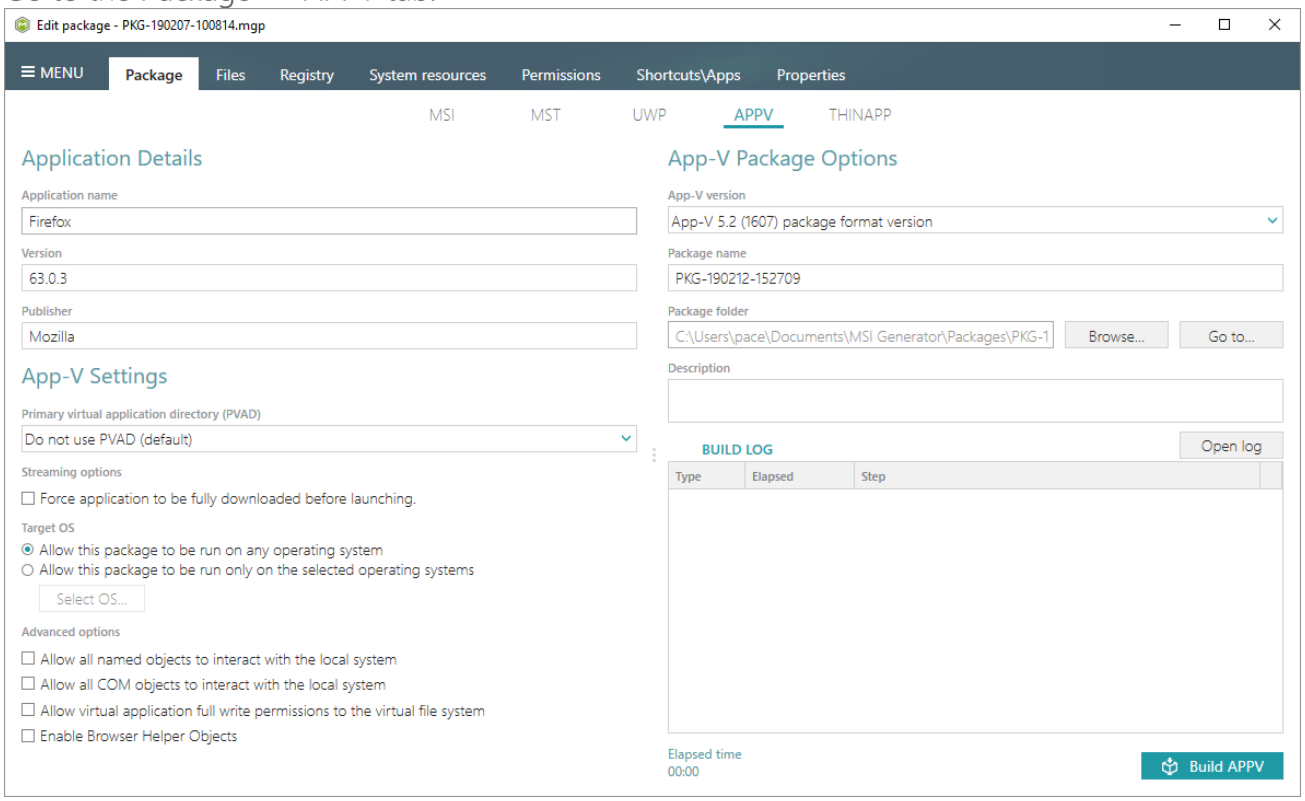
- Update the necessary values in the Application Details section.



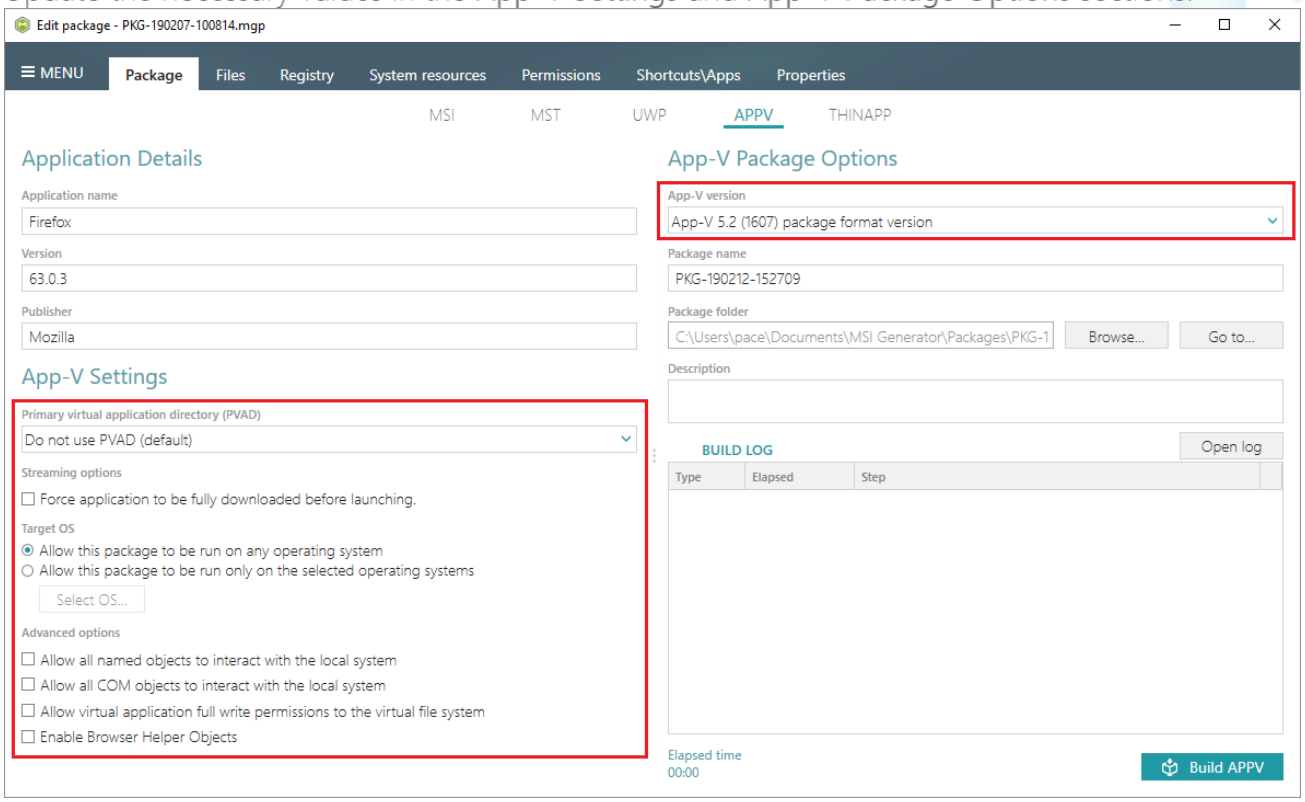
3.8.8.2 Manage Package Settings

Select the package settings such as APPV version, Primary virtual application directory (PVAD), target OS, streaming and some advanced options.

- Go to the Package -> APPV tab.



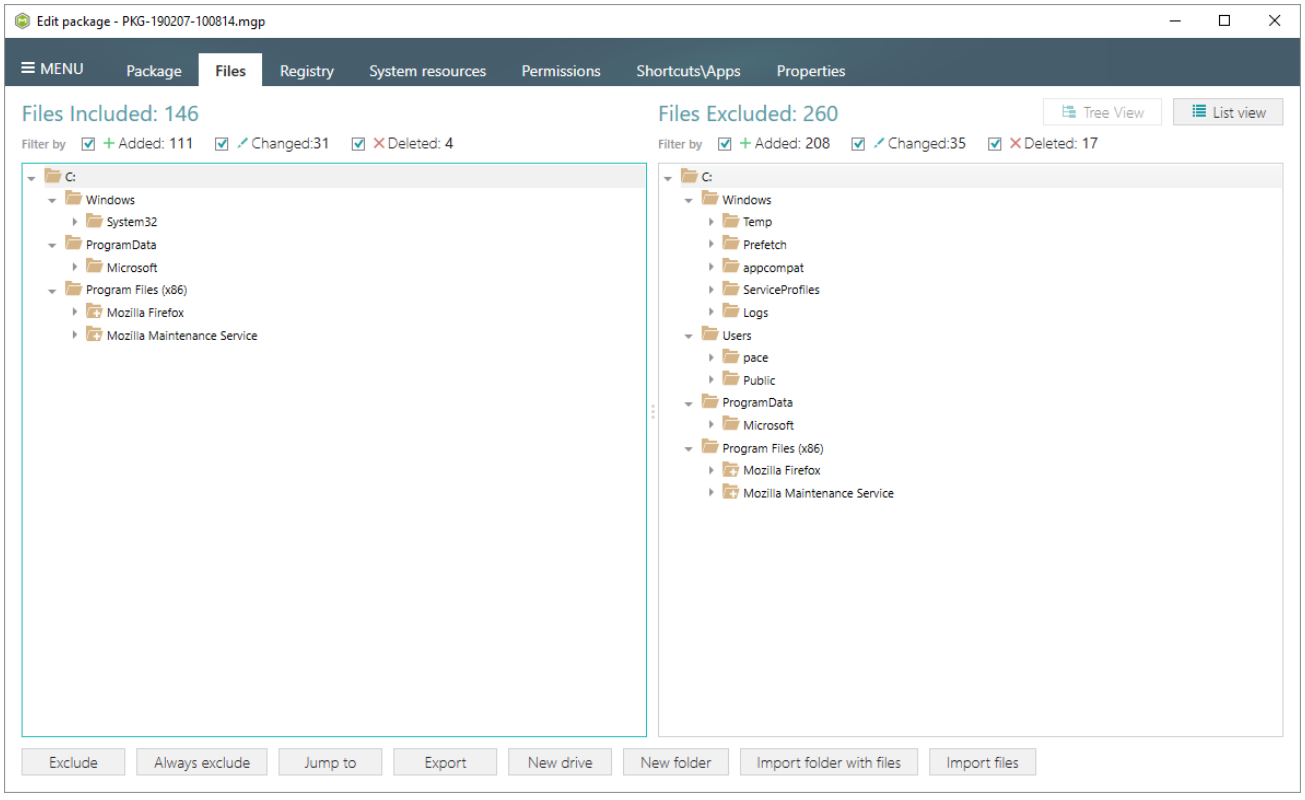
- Update the necessary values in the App-V Settings and App-V Package Options sections.



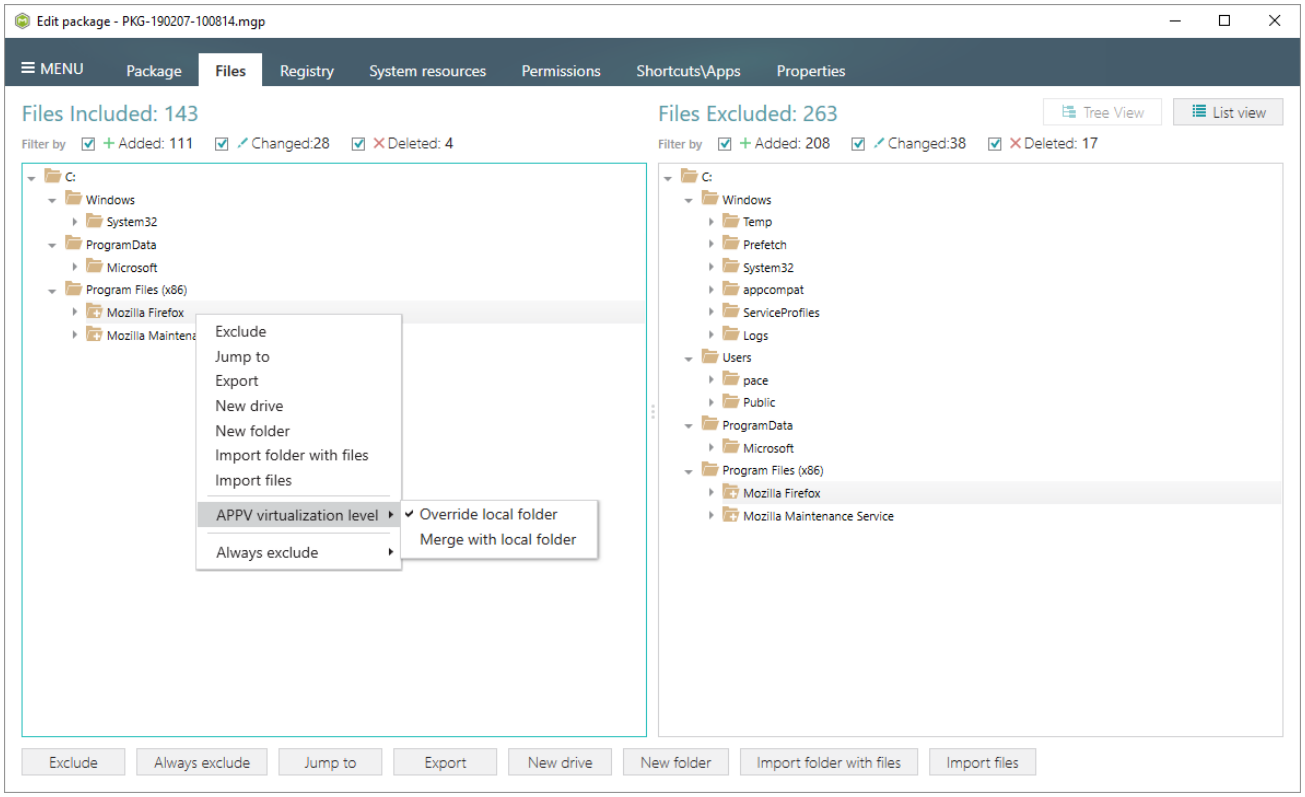
3.8.8.3 Set Folder Virtualization Level

Change the default APPV virtualization (isolation) level for a folder.

- Go to the Files tab.



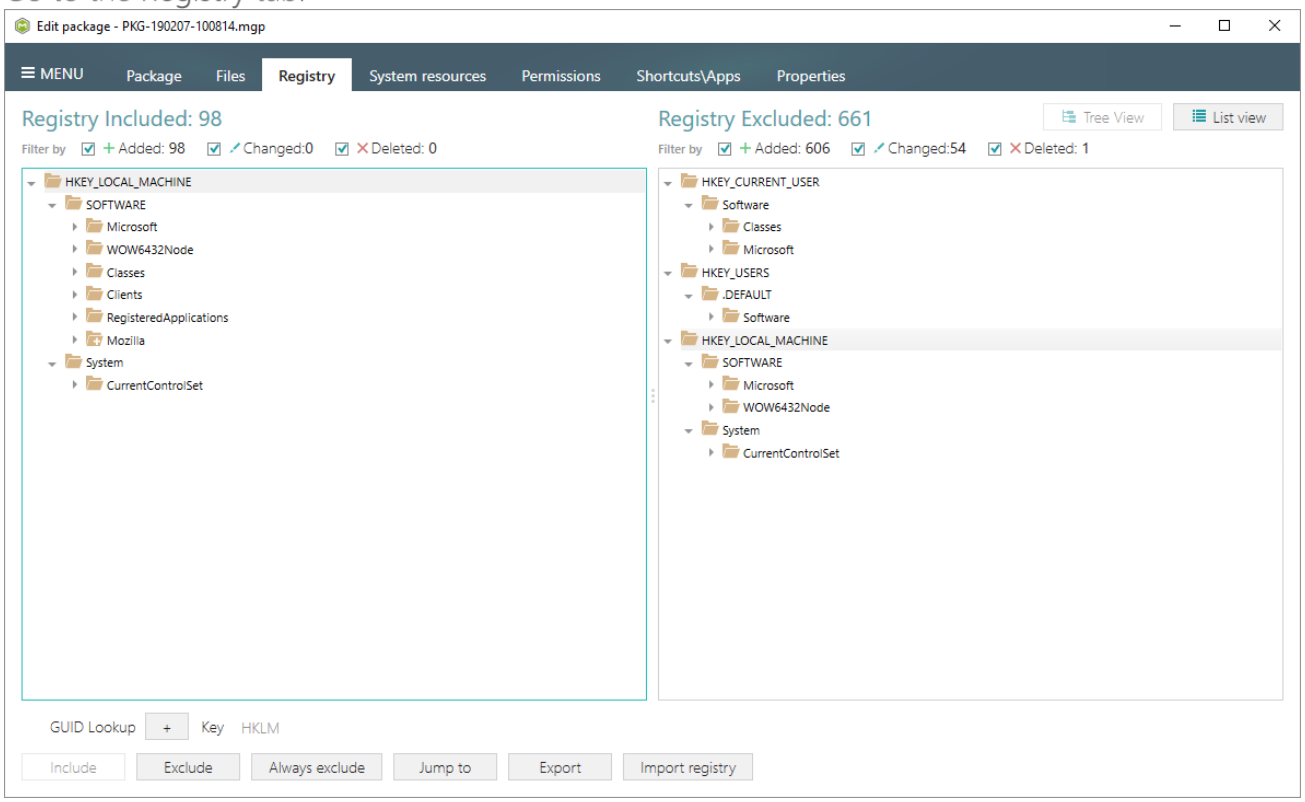
- Select the necessary virtualization level from the context menu of a folder.
  NOTE If the parent folder has the **Override local folder** level, its child folders could not have the **Merge with local folder** level and this case the **APPV virtualization level** submenu will be disabled.



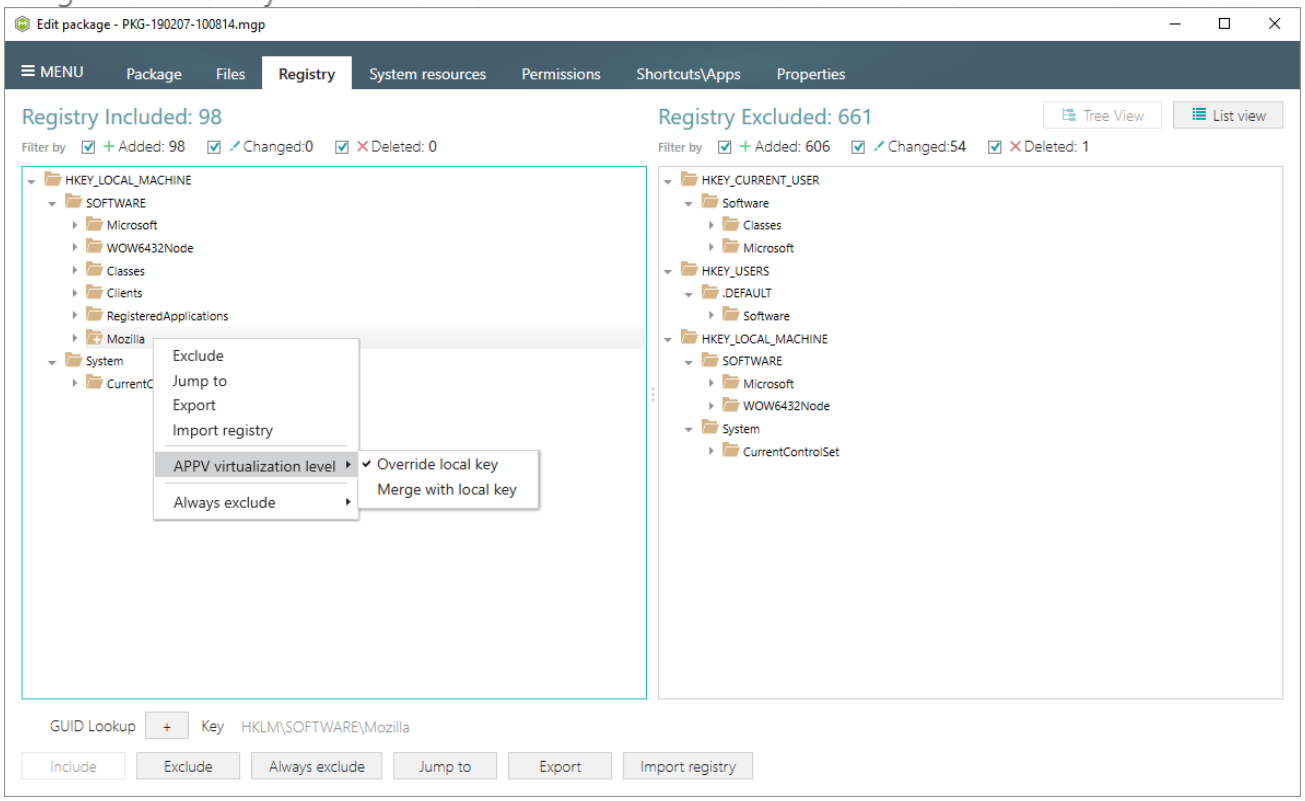
3.8.8.4 Set Registry Virtualization Level

Change the default APPV virtualization (isolation) level for a registry key.

- Go to the **Registry** tab.



- Select the necessary virtualization level from the context menu of a registry key.
  NOTE If the parent key has the **Override local key** level, its child keys could not have the **Merge with local key** level and this case the **APPV virtualization level** submenu will be disabled.



3.8.8.5 Manage Apps, FTA, Shortcuts

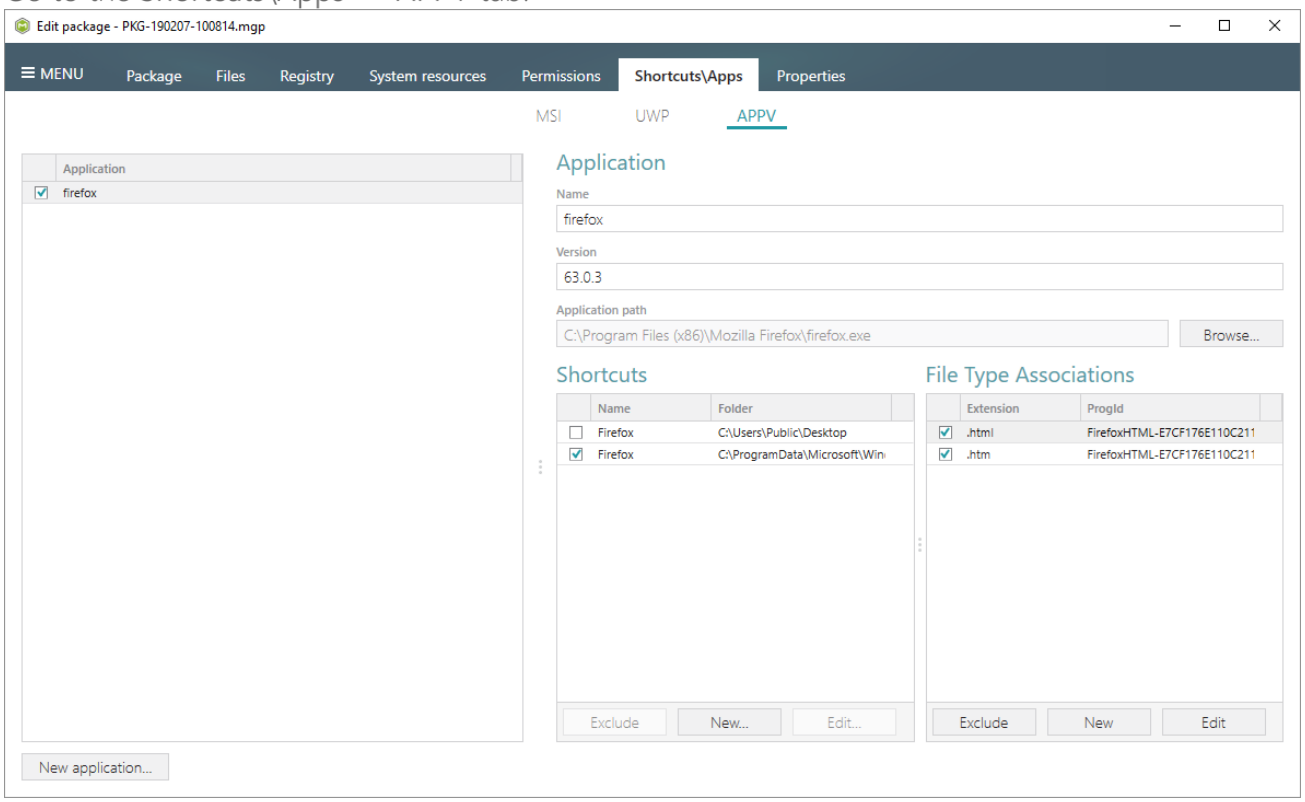
Choose a scenario that better suits your needs:

- **Exclude App**, described in section 3.8.8.5.1
  Exclude the unnecessary app from the package.
- **Include Excluded App**, described in section 3.8.8.5.2
  Include, the previously excluded app, back to the package.
- **Add App**, described in section 3.8.8.5.3
  Add a new app to the package.
- **Add App Shortcut**, described in section 3.8.8.5.4
  Add an app shortcut to the package.
- **Add App FTA**, described in section 3.8.8.5.5
  Add an app FTA (File Type Association) to the package.

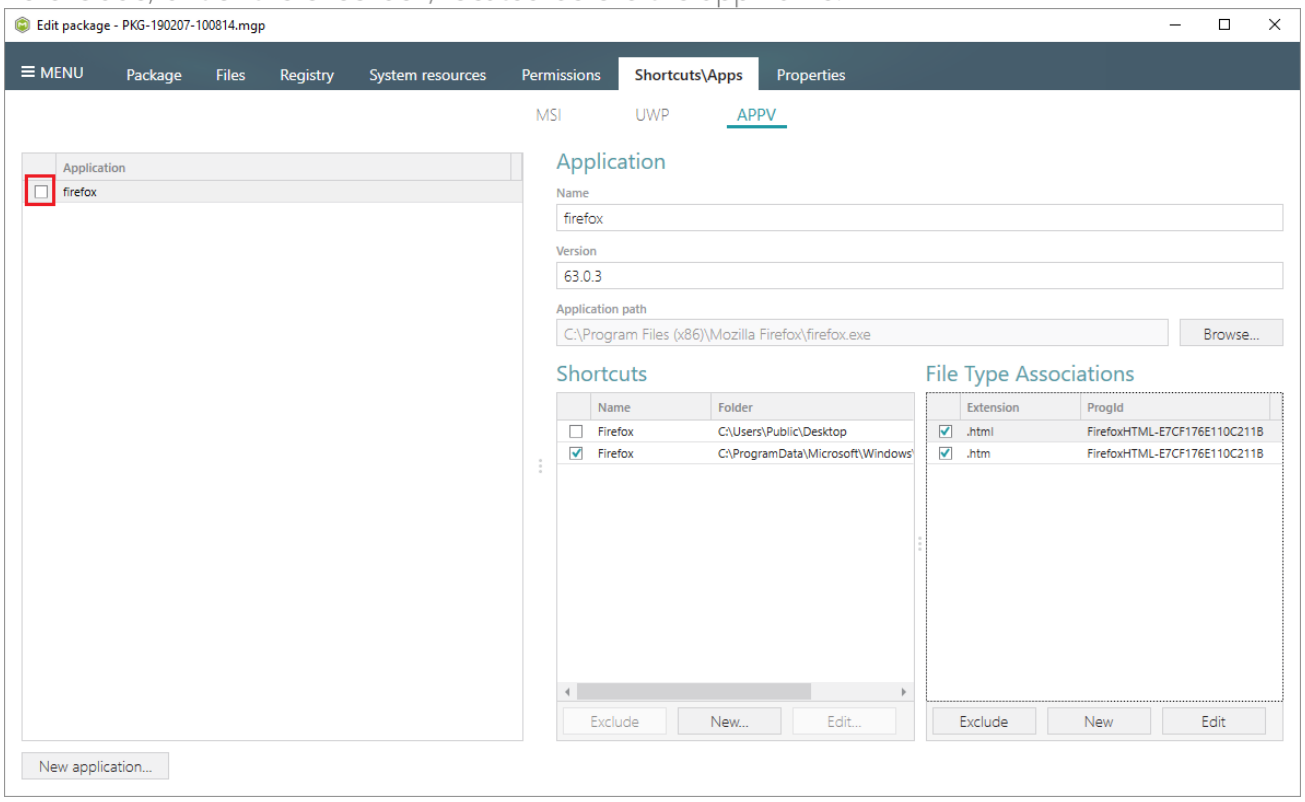
3.8.8.5.1 Exclude App

Exclude the unnecessary app from the package.

- Go to the Shortcuts\Apps -> APPV tab.



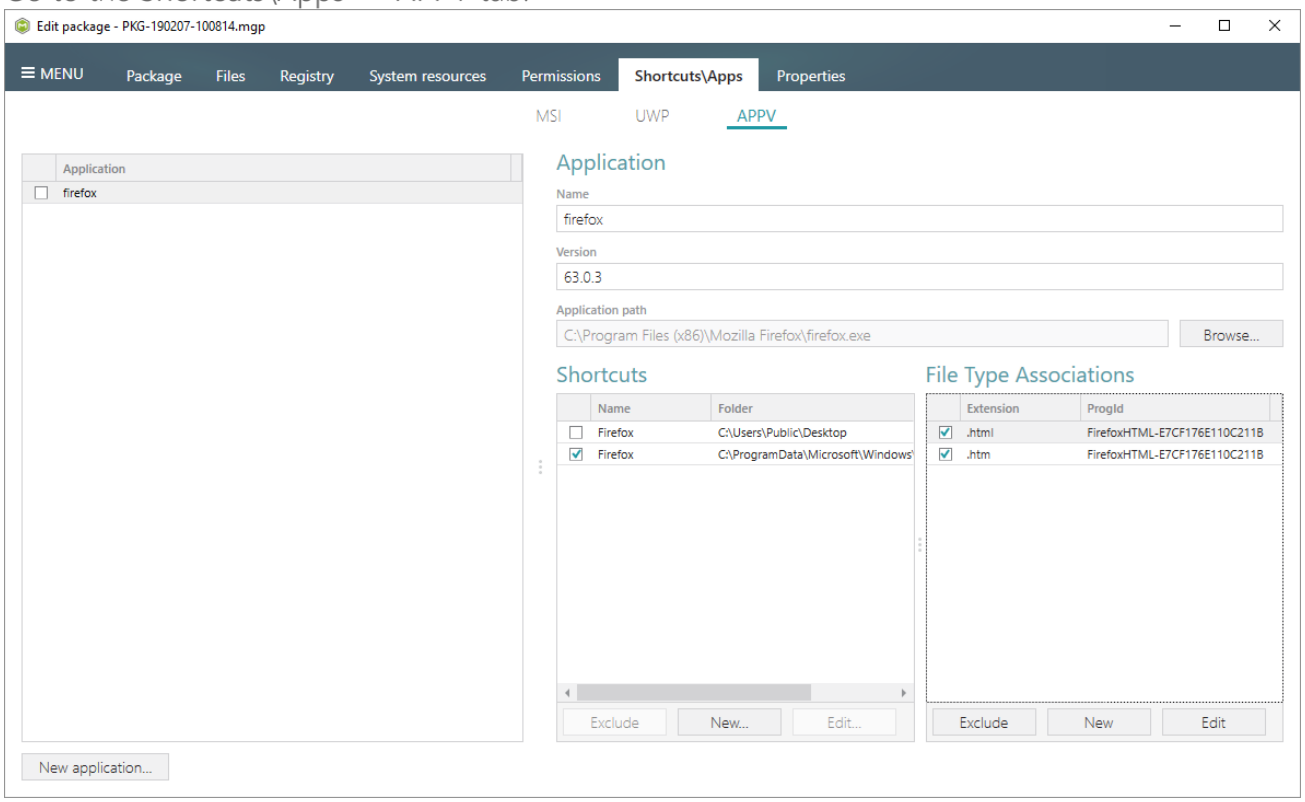
- To exclude, untick the checkbox, located before the app name.



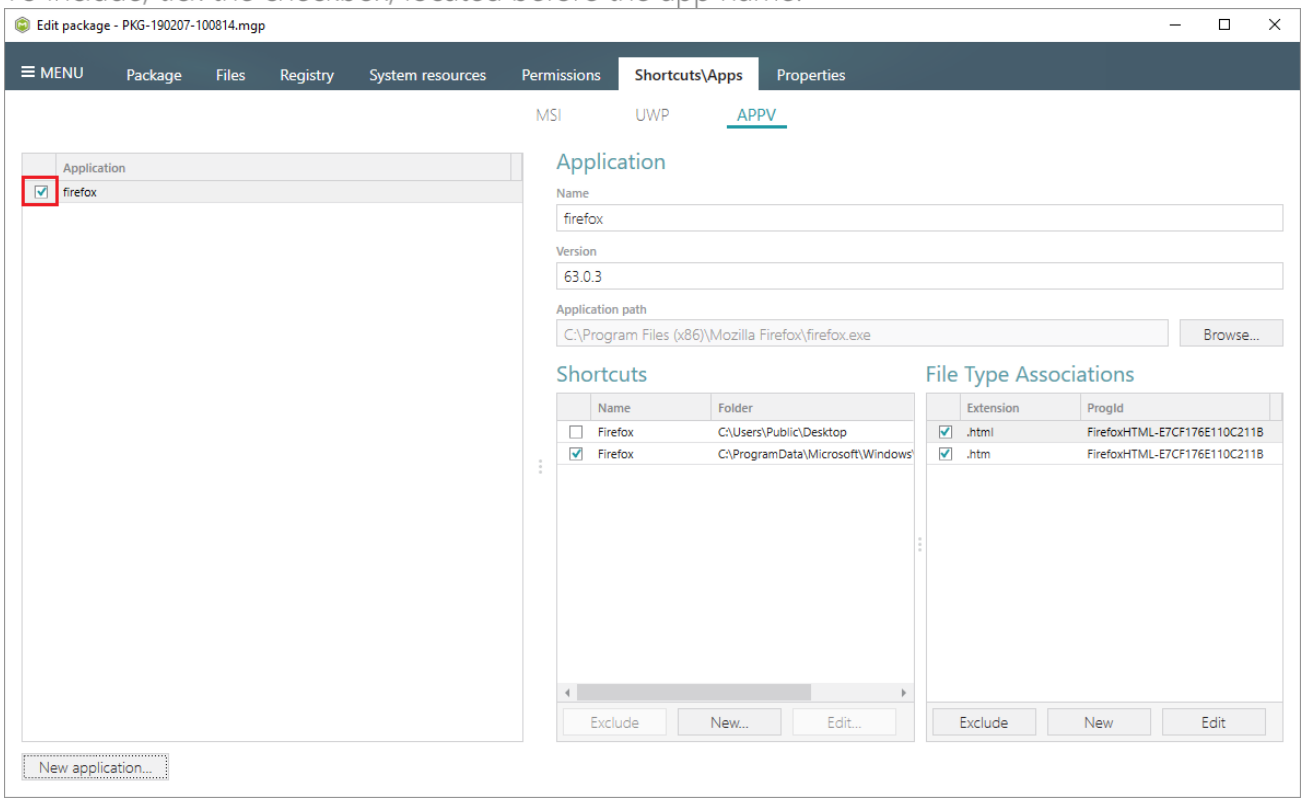
3.8.8.5.2 Include Excluded App

Include, the previously excluded app, back to the package.

- Go to the Shortcuts\Apps -> APPV tab.



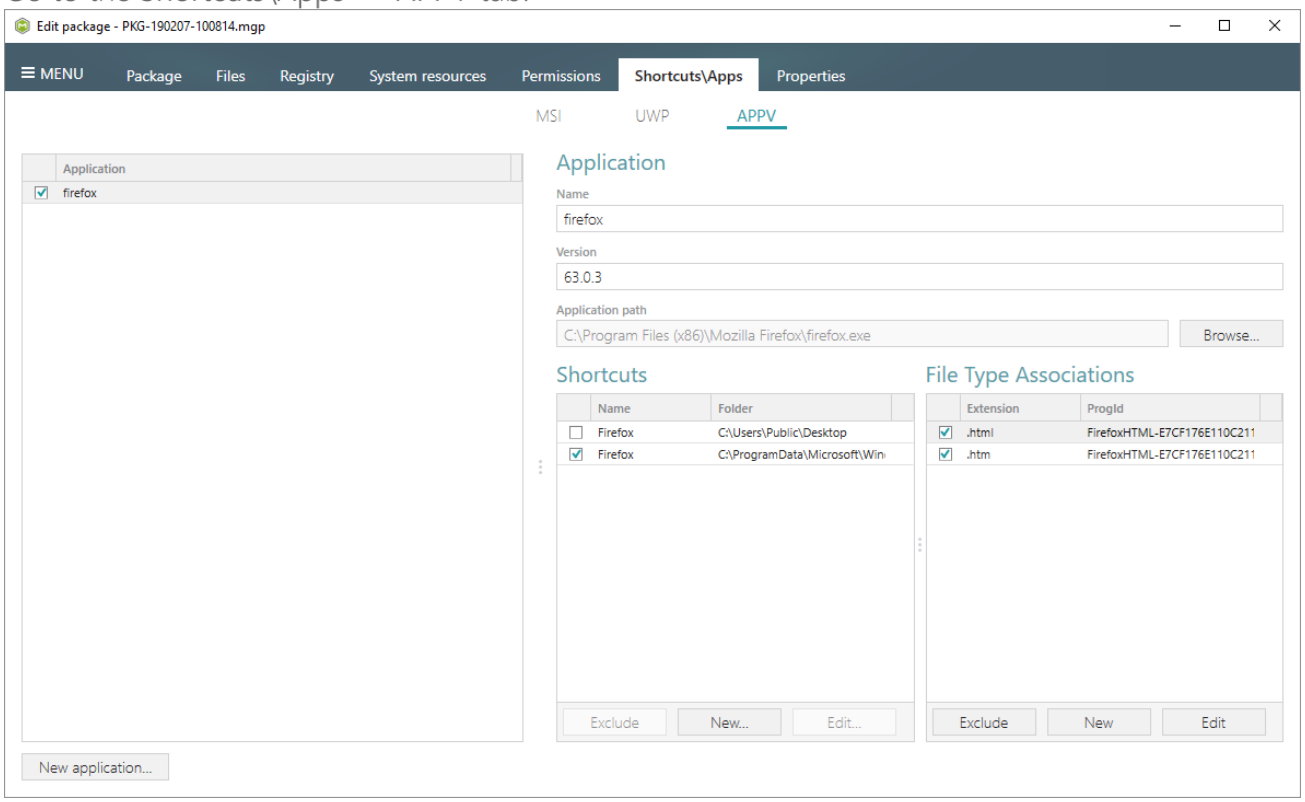
- To include, tick the checkbox, located before the app name.



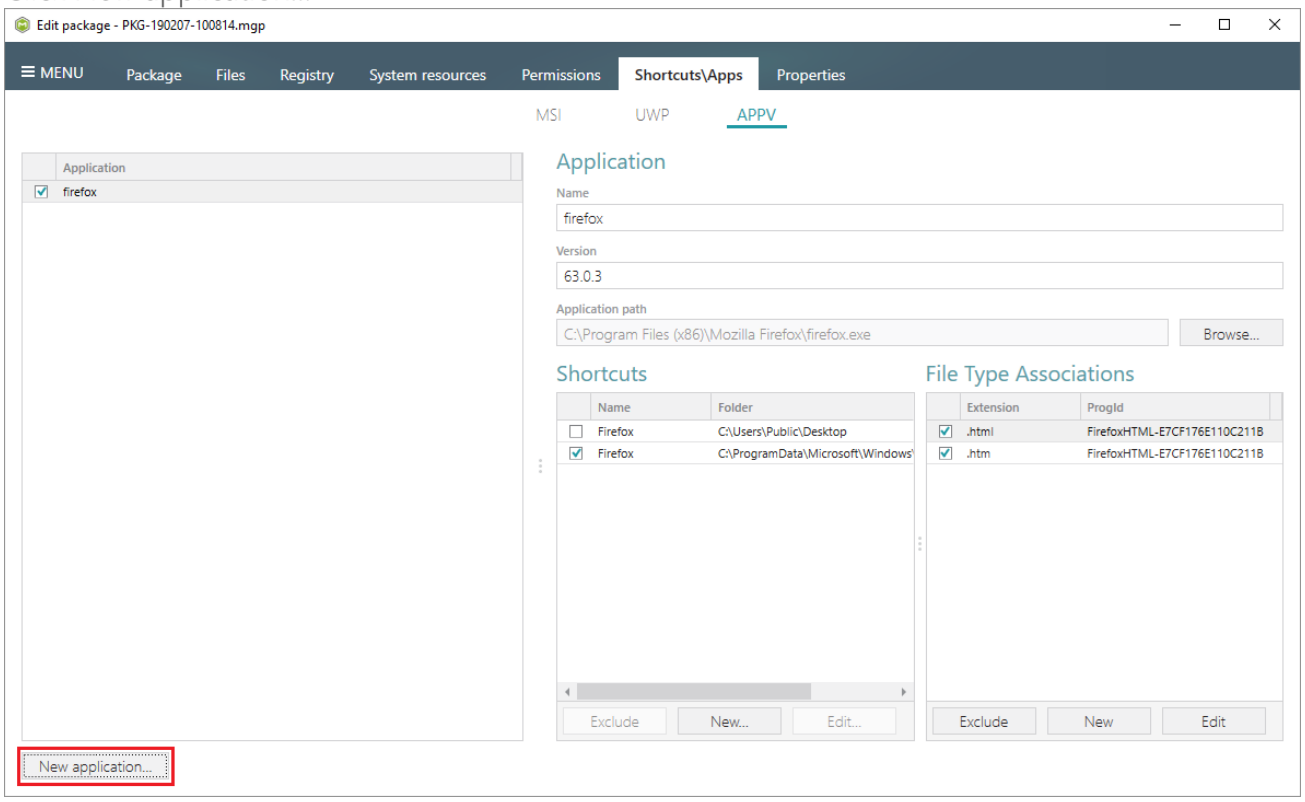
3.8.8.5.3 Add App

Add a new app to the package.

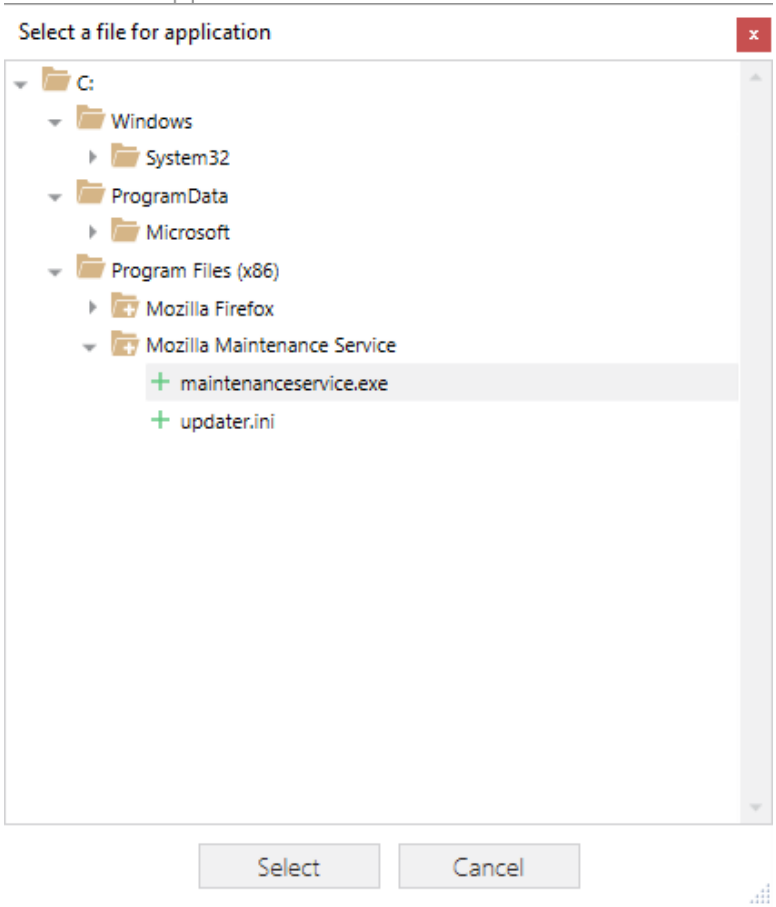
- Go to the Shortcuts\Apps -> APPV tab.



- Click New application...



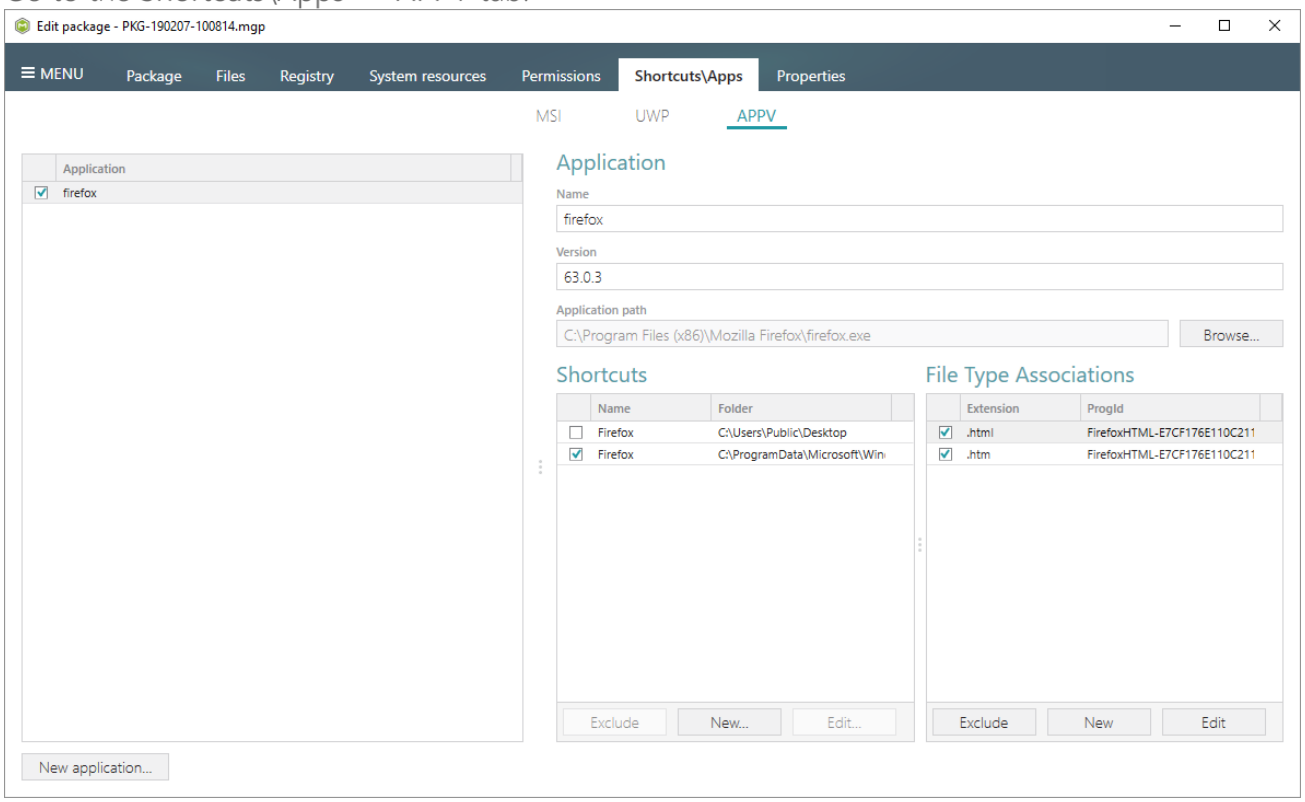
- Choose an app executable file and click **Select**.



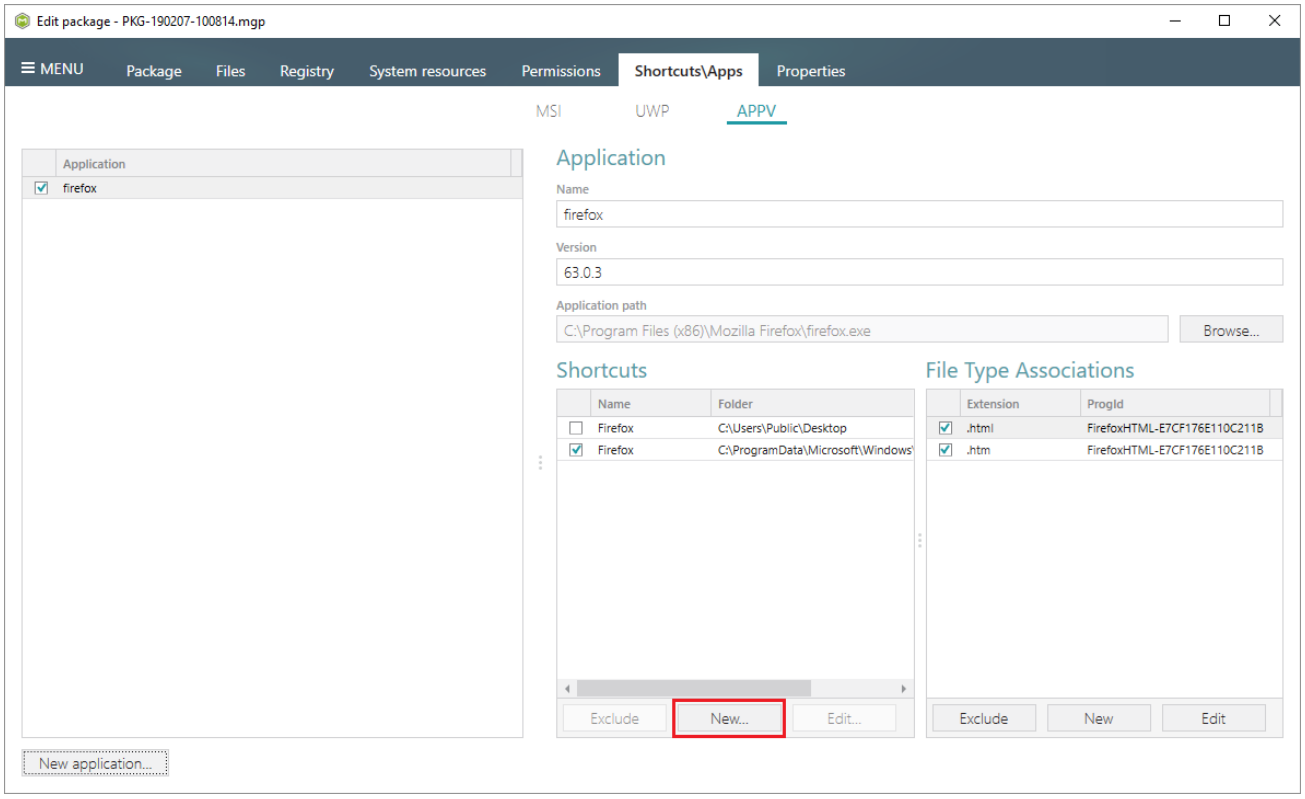
3.8.8.5.4 Add App Shortcut

Add an app shortcut to the package.

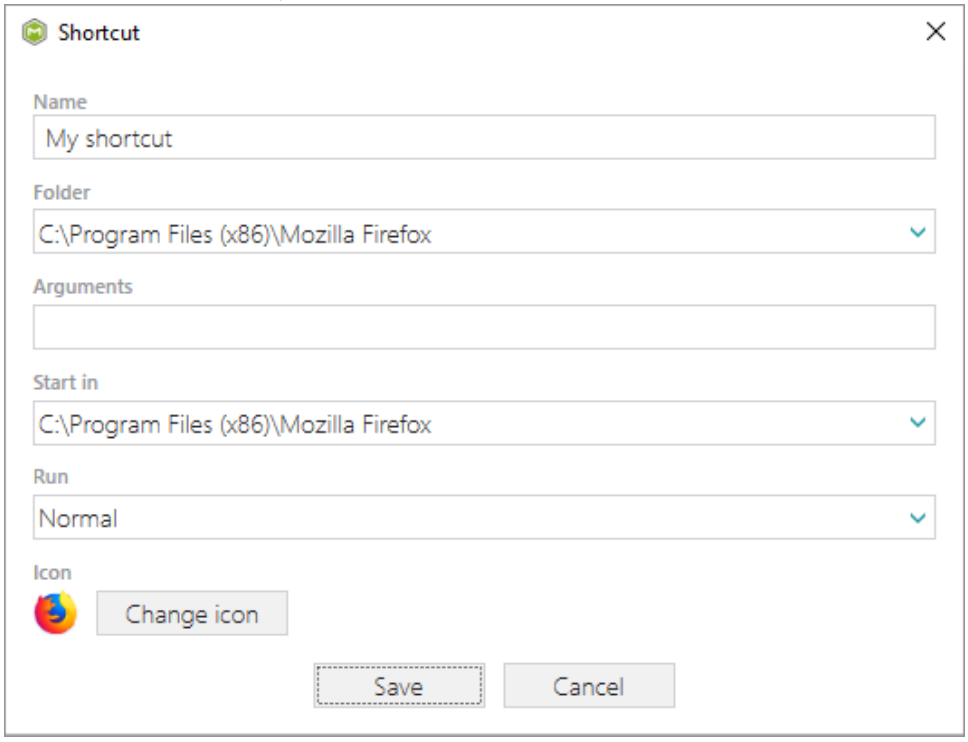
- Go to the Shortcuts\Apps -> APPV tab.



- Choose an app from the list, for which you want to add a shortcut and click New... in the Shortcuts section.



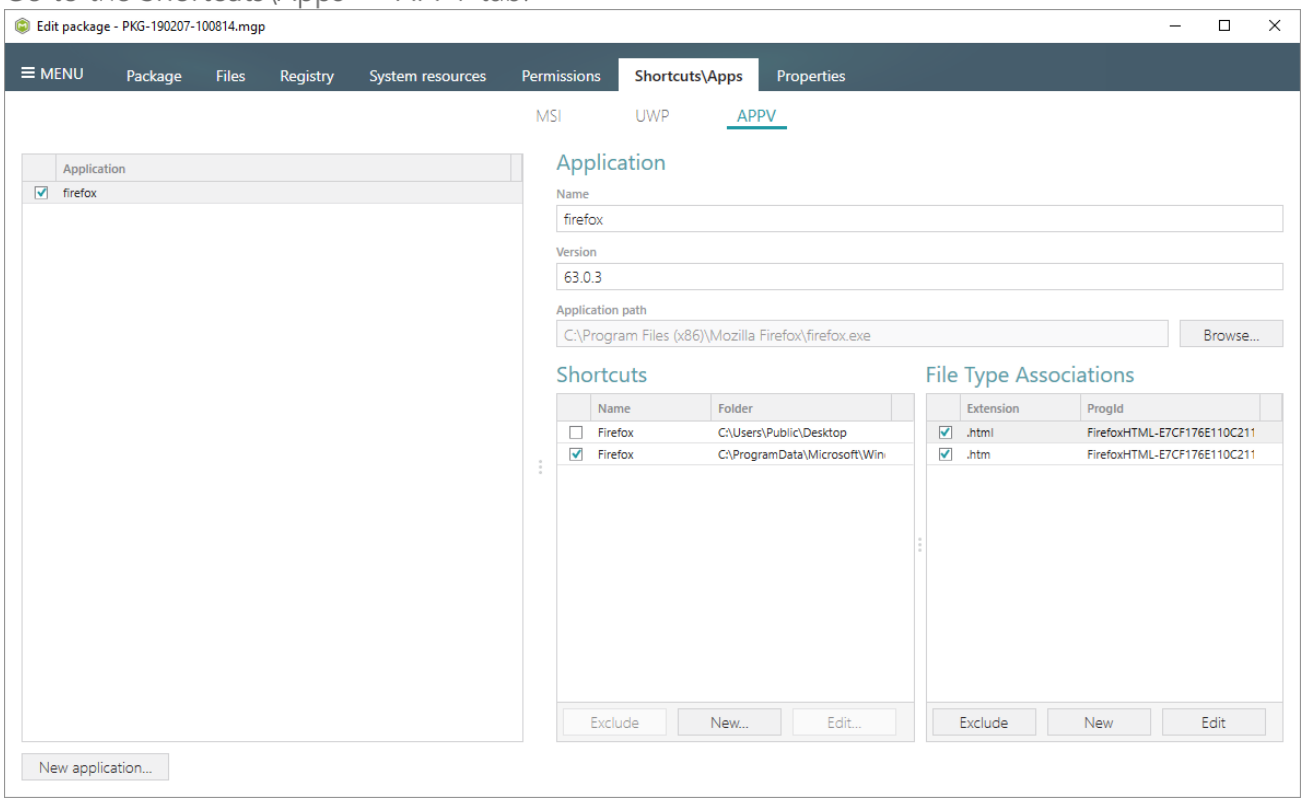
- Enter a shortcut name, select its destination folder, start in (working directory), run parameter and icon. Afterwards, click Save.



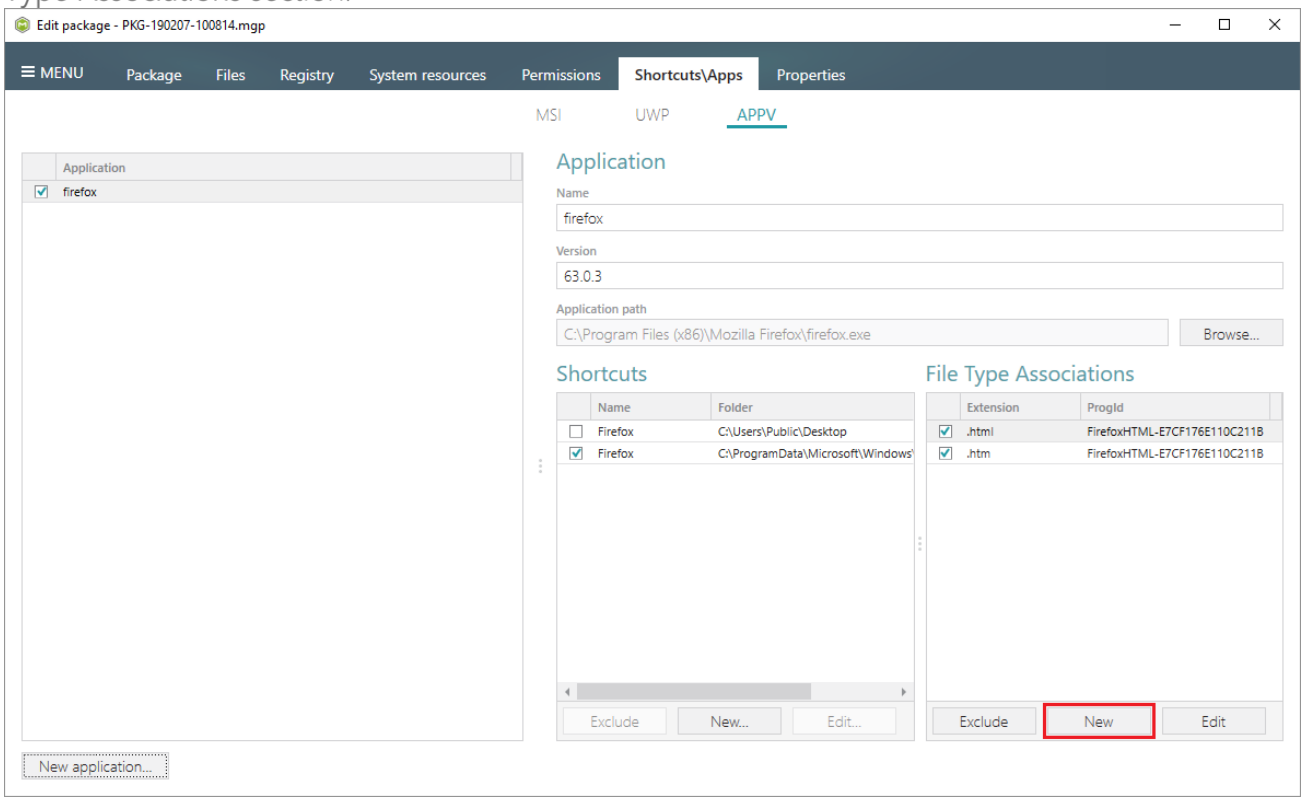
3.8.8.5.5 Add App FTA

Add an app FTA (File Type Association) to the package.

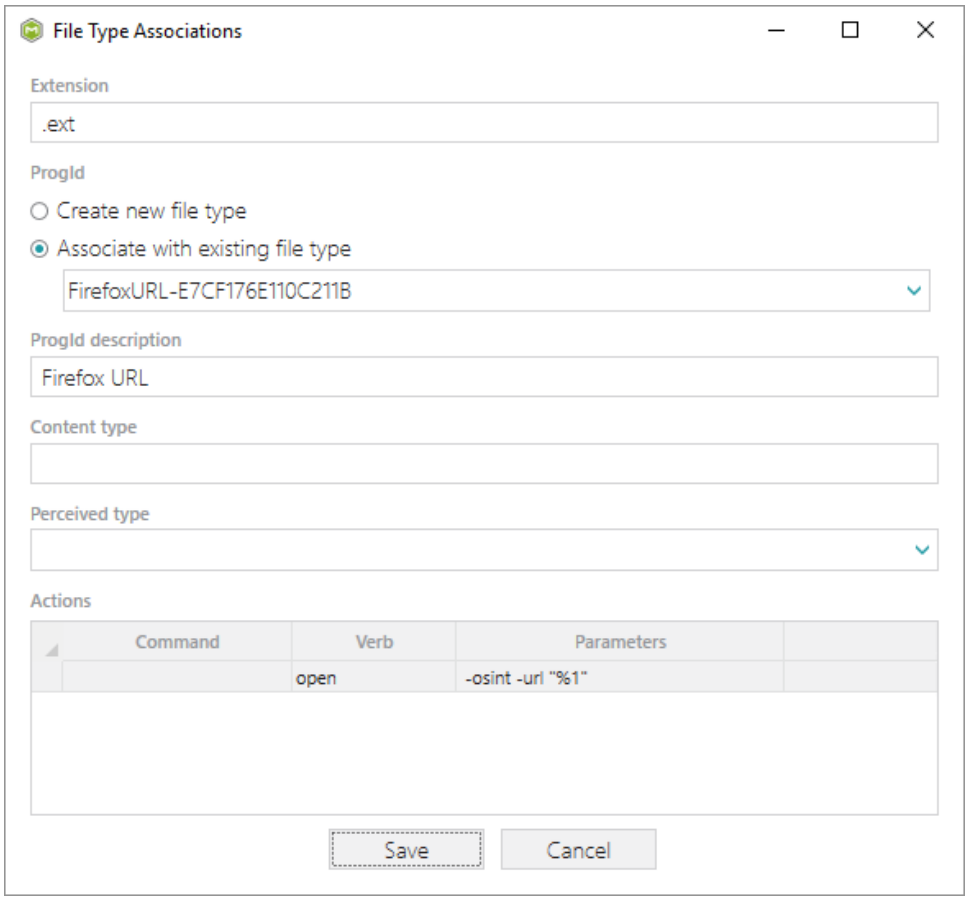
- Go to the Shortcuts\Apps -> APPV tab.



- Choose an app from the list, for which you want to add an FTA and click **New...** in the **File Type Associations** section.



- Specify an extension, ProgId, content and perceived types, context menu commands and click Save.



3.8.9 THINAPP Settings

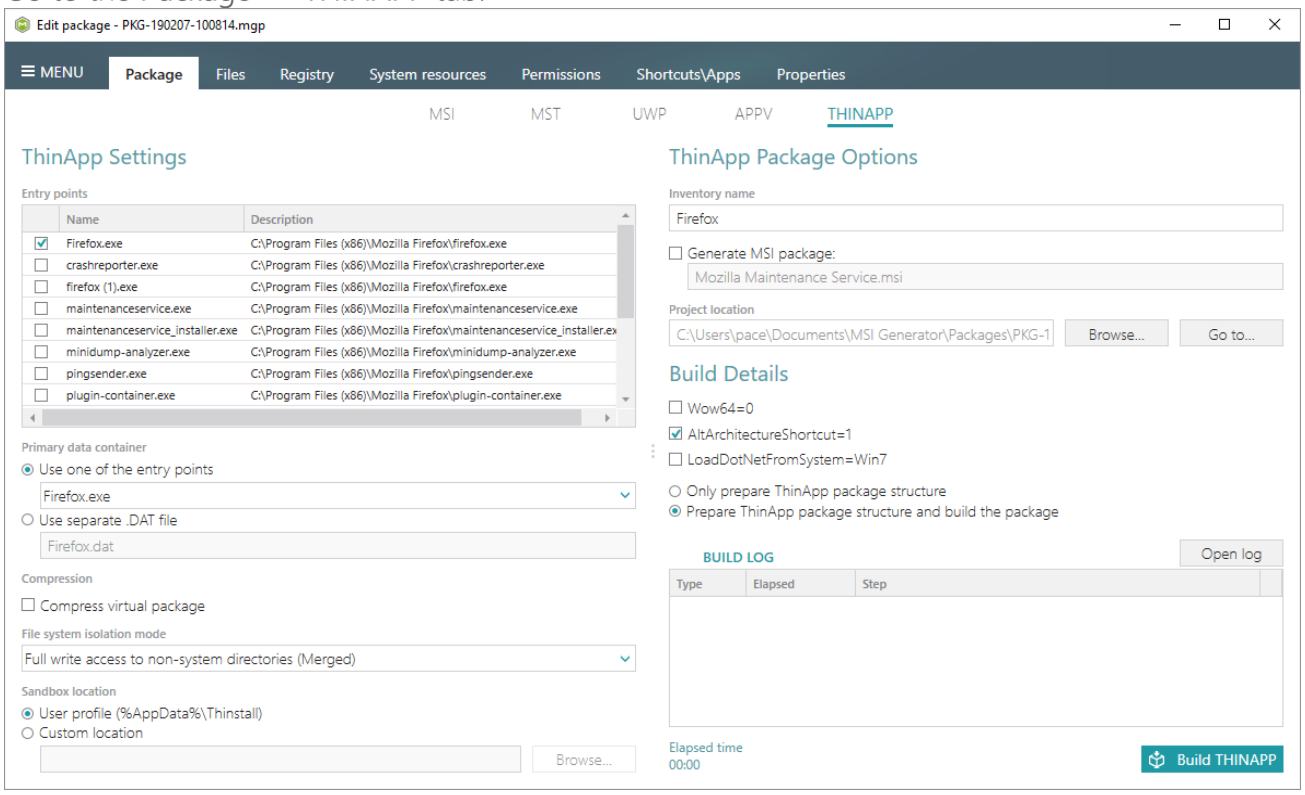
Choose a scenario that better suits your needs:

- Set Entry Points, described in section 3.8.9.1
  Select which of package executable (EXE) files that will be shortcuts into the virtual environment.
- Define Primary Data Container, described in section 3.8.9.2
  Define how the virtual file system, virtual registry and ThinApp runtime will be stored in the package: as a separate file (usually DAT) or within one of entry points.

3.8.9.1 Set Entry Points

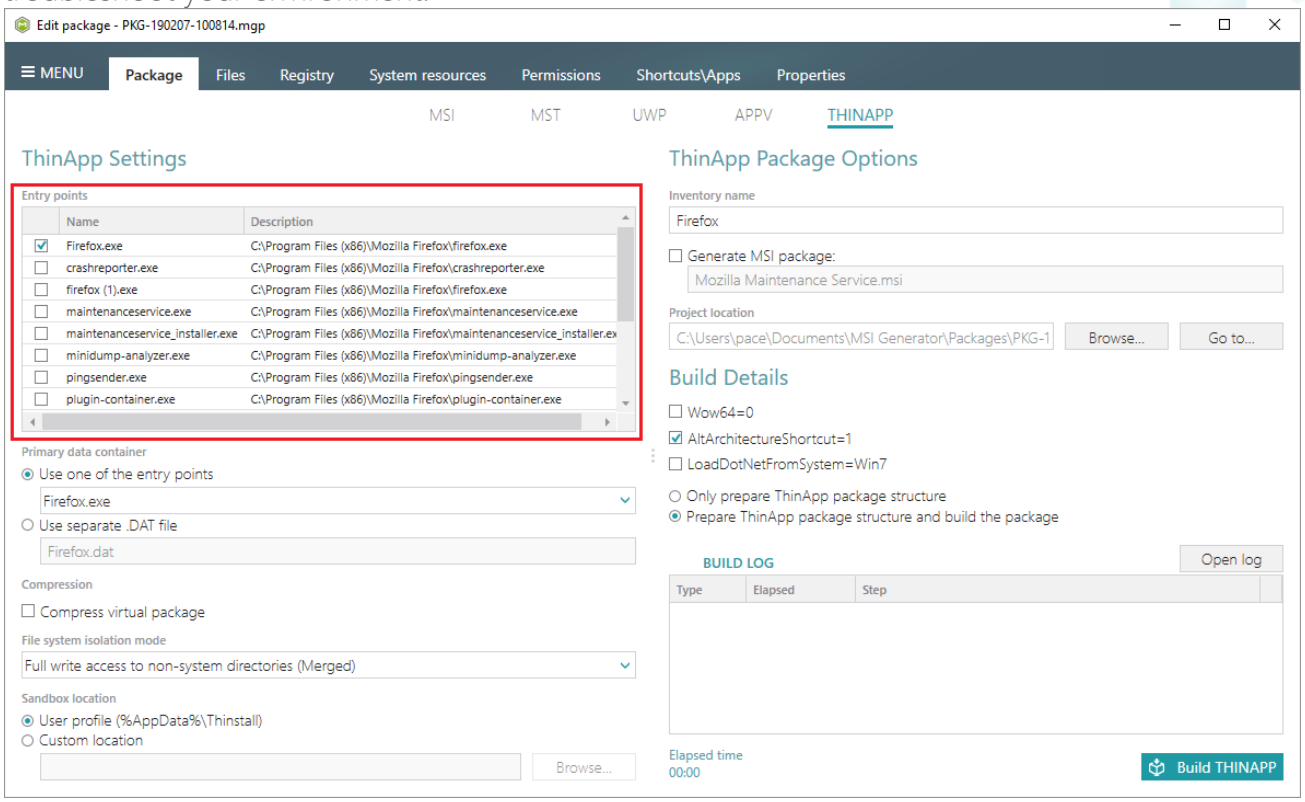
Select which of package executable (EXE) files that will be shortcuts into the virtual environment. For information, see Defining Entry Points as Shortcuts into the Virtual Environment.

- Go to the Package -> THINAPP tab.



- Select executable files from the Entry points list, which you want to define as shortcuts into the virtual environment that will start the virtual application.
  NOTE In the end of the list you can find the cmd.exe and regedit.exe entry points, used to

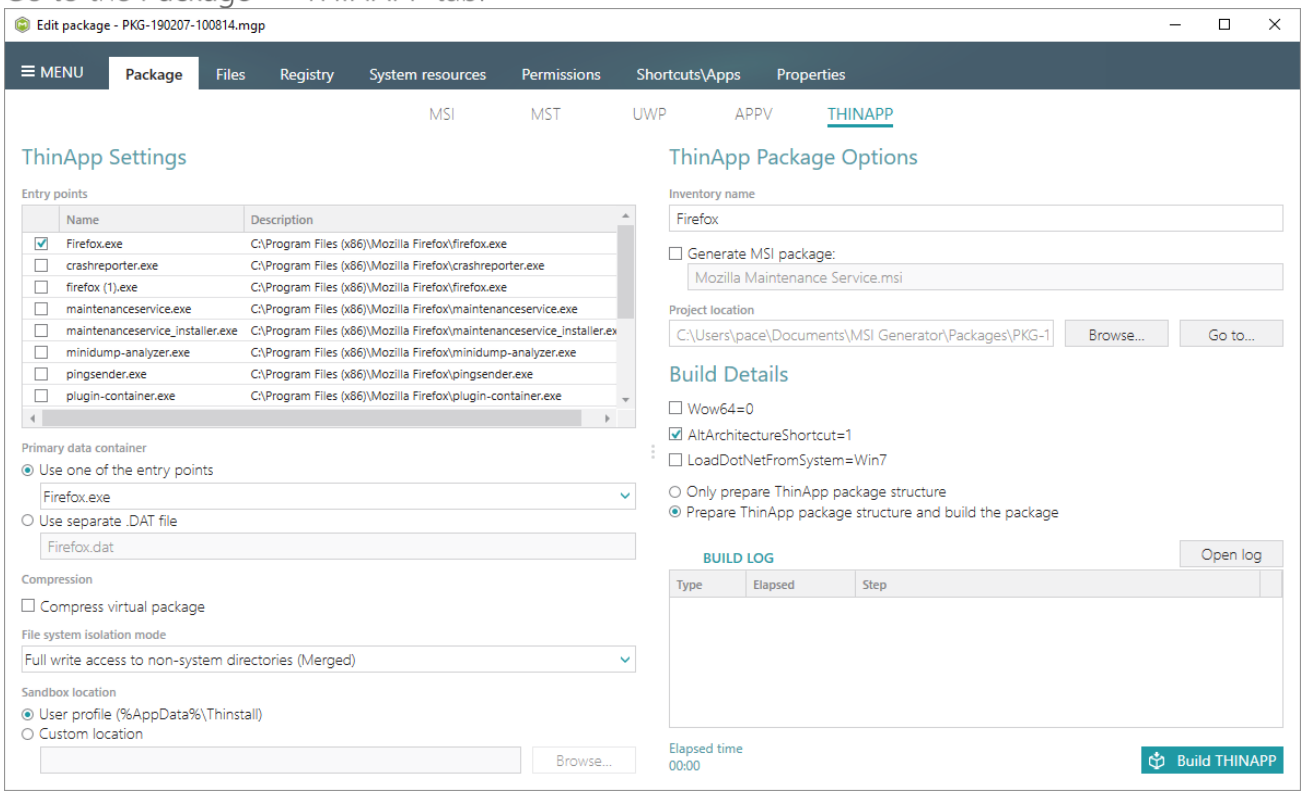
troubleshoot your environment.



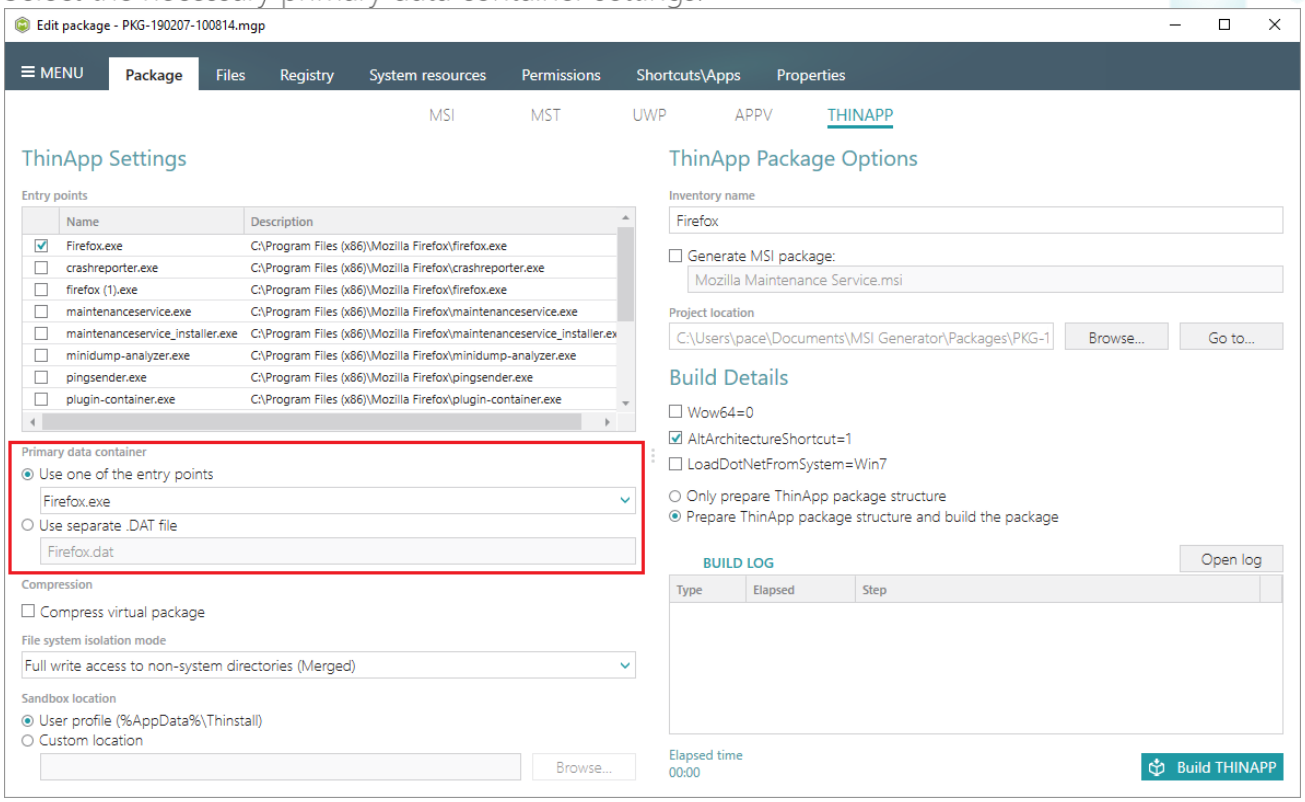
3.8.9.2 Define Primary Data Container

Define how the virtual file system, virtual registry and ThinApp runtime will be stored in the package: as a separate file (usually DAT) or within one of entry points. For information, see Defining the Primary Data Container.

- Go to the Package -> THINAPP tab.

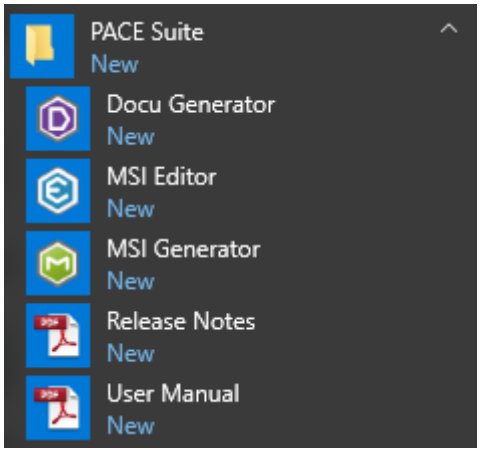


- Select the necessary primary data container settings.

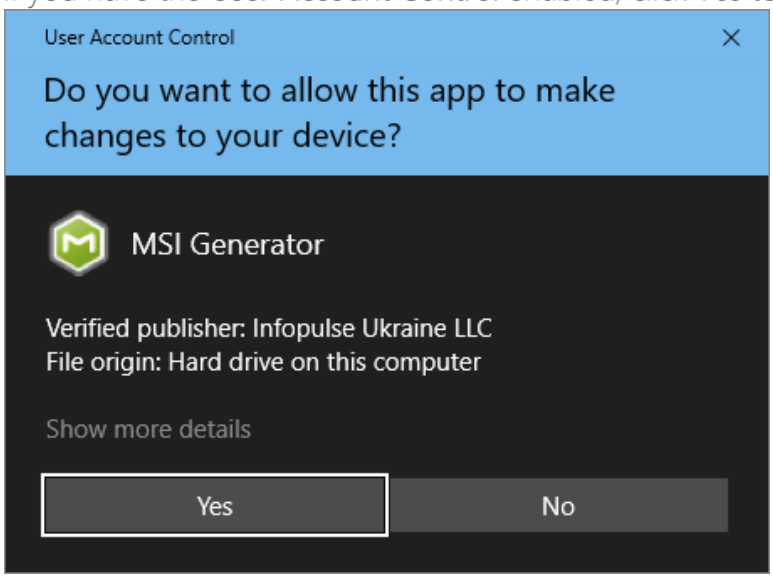


3.9 Edit APPV

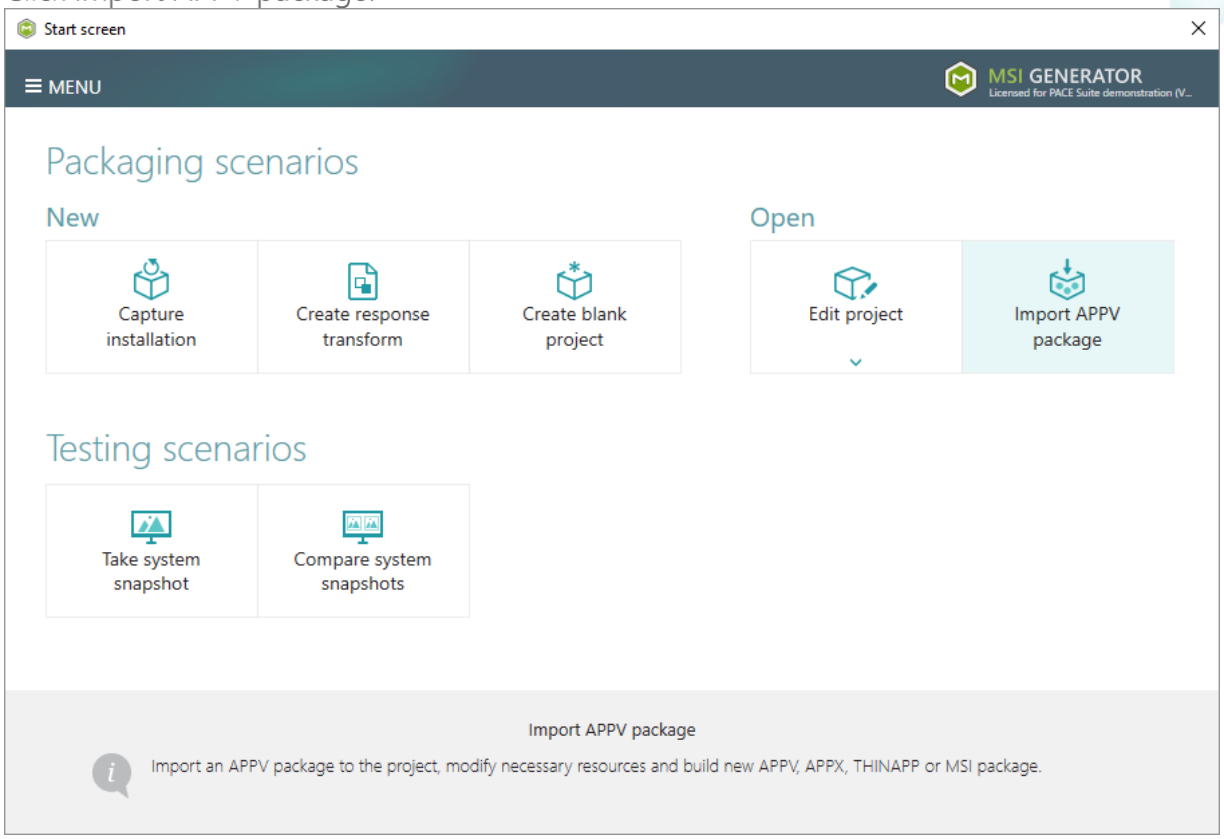
- Launch MSI Generator from the desktop of the start menu shortcut.



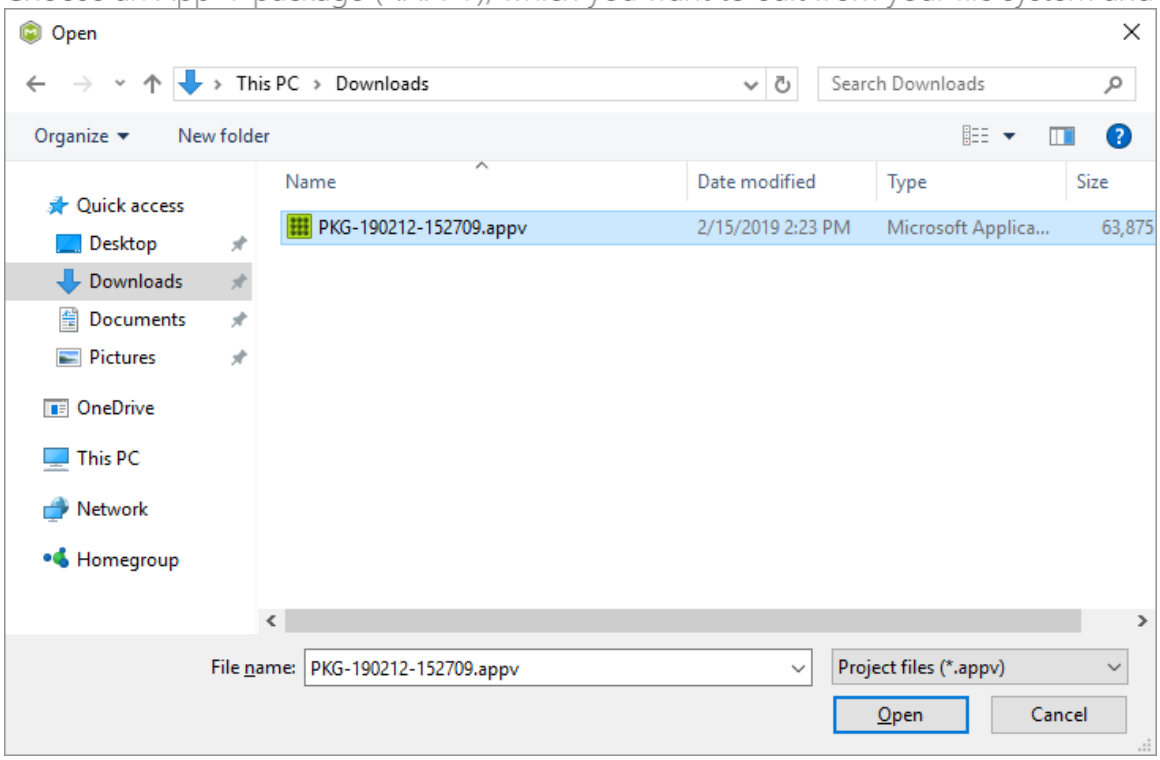
- If you have the User Account Control enabled, click Yes to launch the application.



- Click Import APPV package.

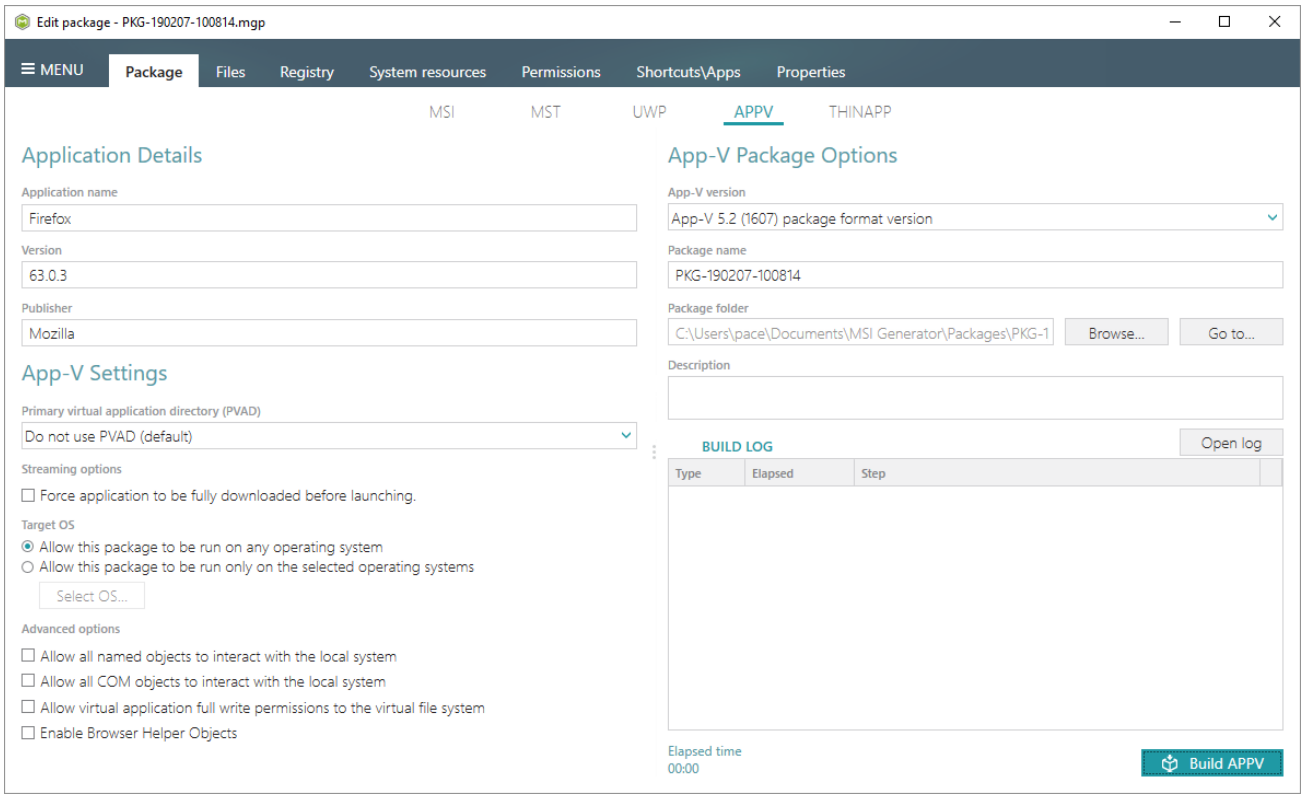


- Choose an App-V package (*.APPV), which you want to edit from your file system and Open.

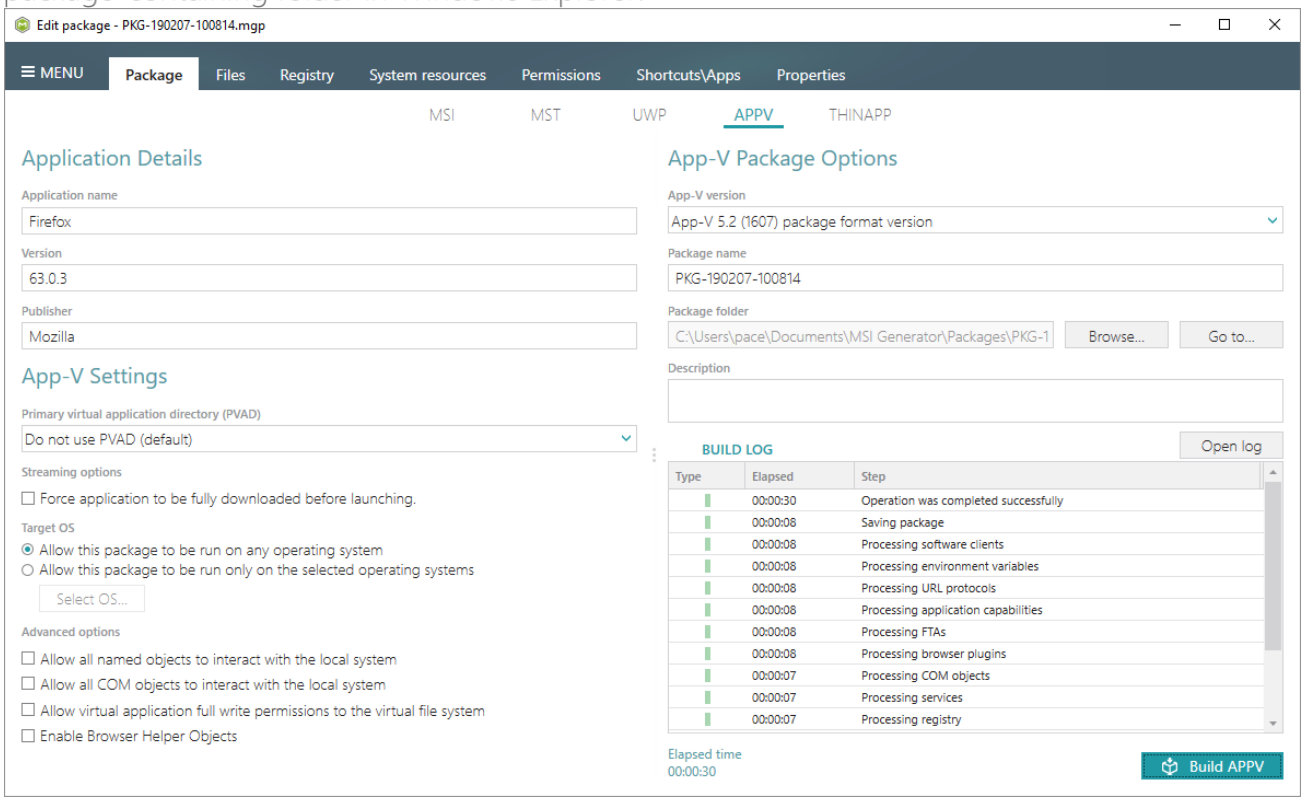


- For managing project resources such as files, folders, registry, and shortcuts, refer to the respective section of this document:
    - Regarding File and Folders see section 3.8.2
    - Regarding Registry see section 3.8.3

- For managing APPV Settings such as application details, target OS, PVAD, streamin settings, APPV version, virtualization level for the folders and registry keys, APPV apps, their shortcuts and file type associations (FTA), see section 3.8.8
- Finaly, to generate an APPV package navigate to the Package -> APPV tab and click Build APPV.



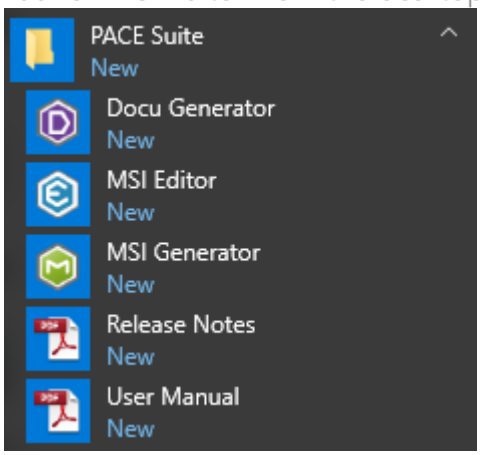
- Once the package is built, click Go to..., located next to the Project folder field, to open the package-containing folder in Windows Explorer.



3.10 Deploy MSI to SCCM

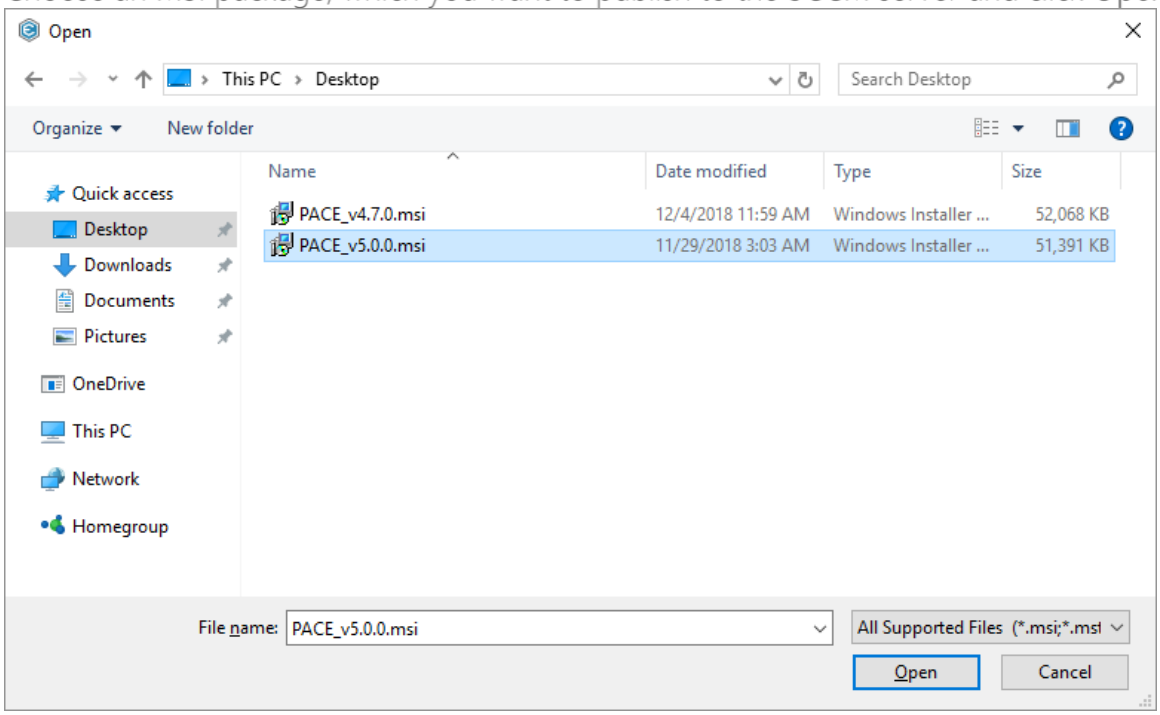
NOTE For publishing MSI packages to Microsoft SCCM 2007 (SP1/SP2/R1/R2) or SCCM 2012 (SP1/SP2/R2/R2 SP1) server, ensure that your system contains Microsoft Management Framework 3.0 and Windows Remote Management (WinRM) service enabled.

- Launch MSI Editor from the desktop or the start menu shortcut.

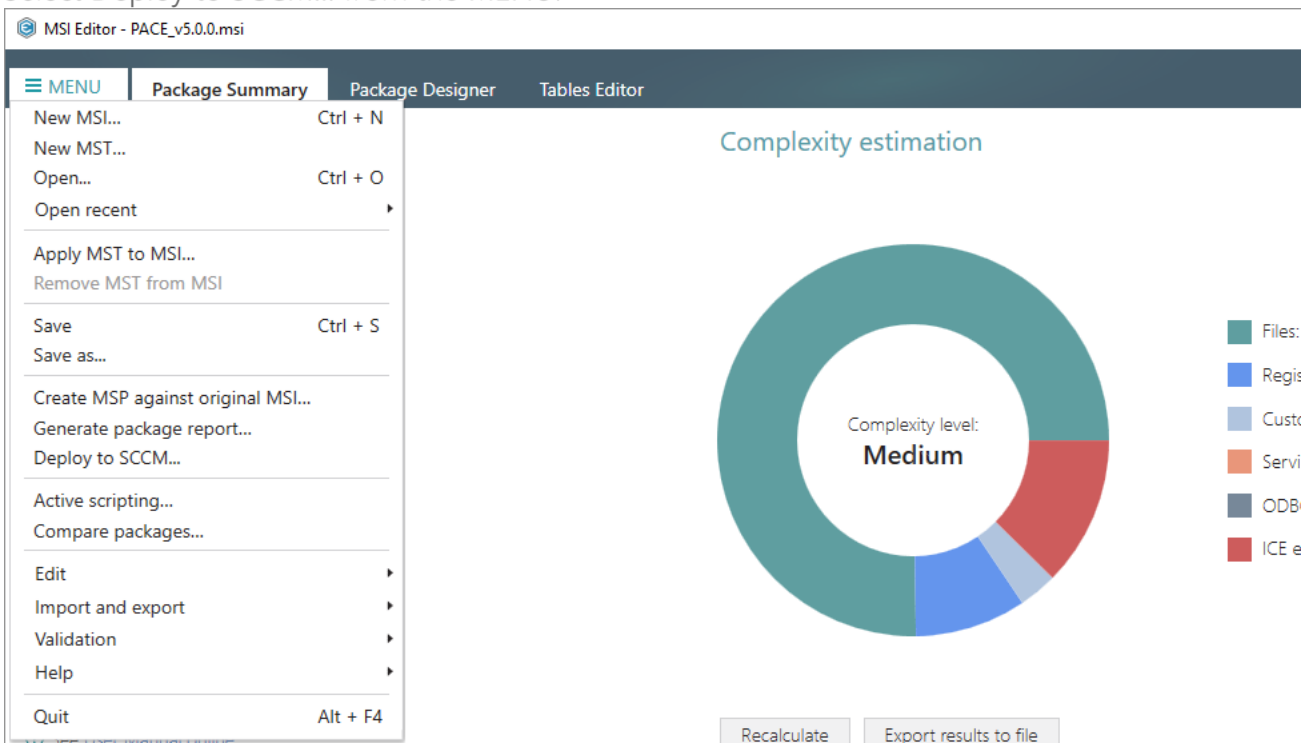


- Select Open... from MENU.

- Choose an MSI package, which you want to publish to the SCCM server and click Open.

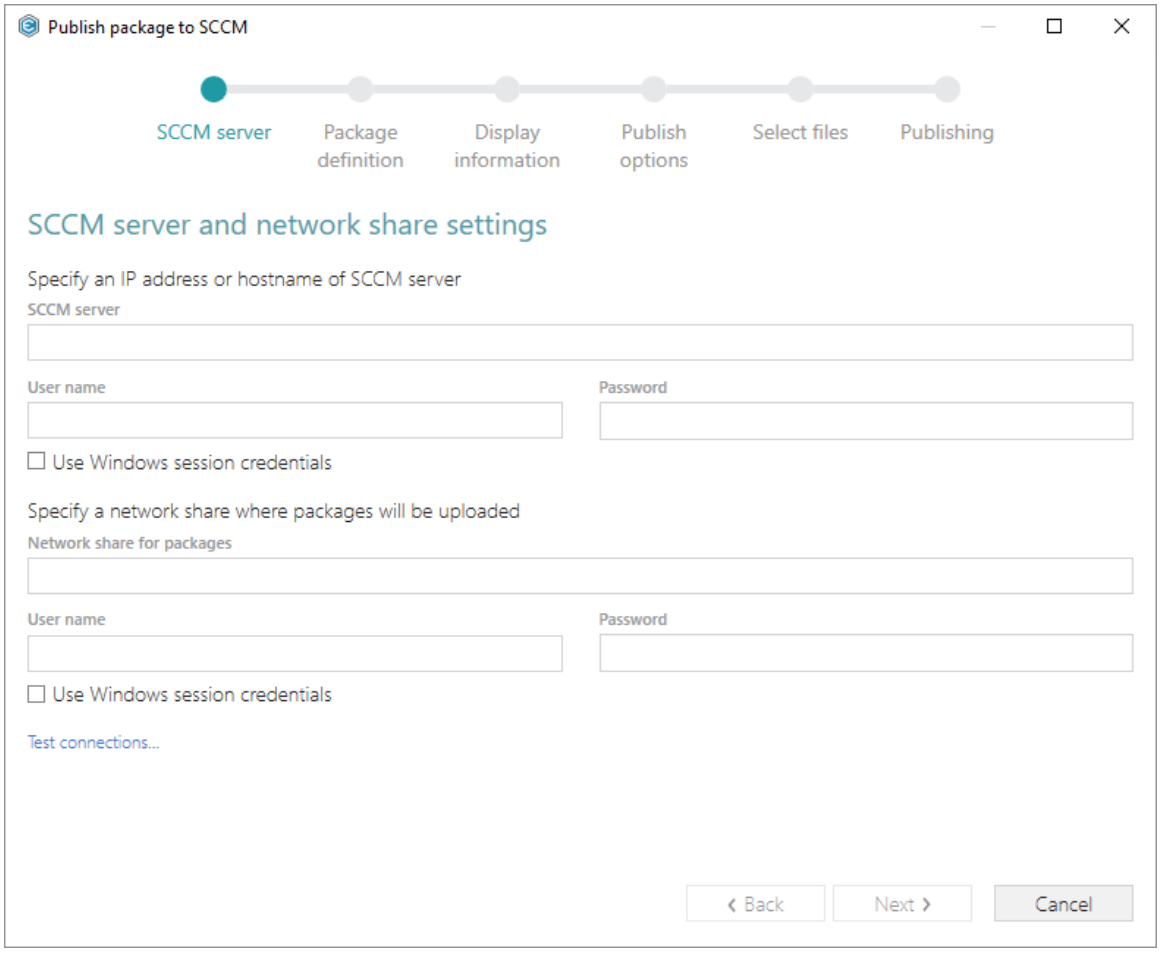


- Select Deploy to SCCM... from the MENU.

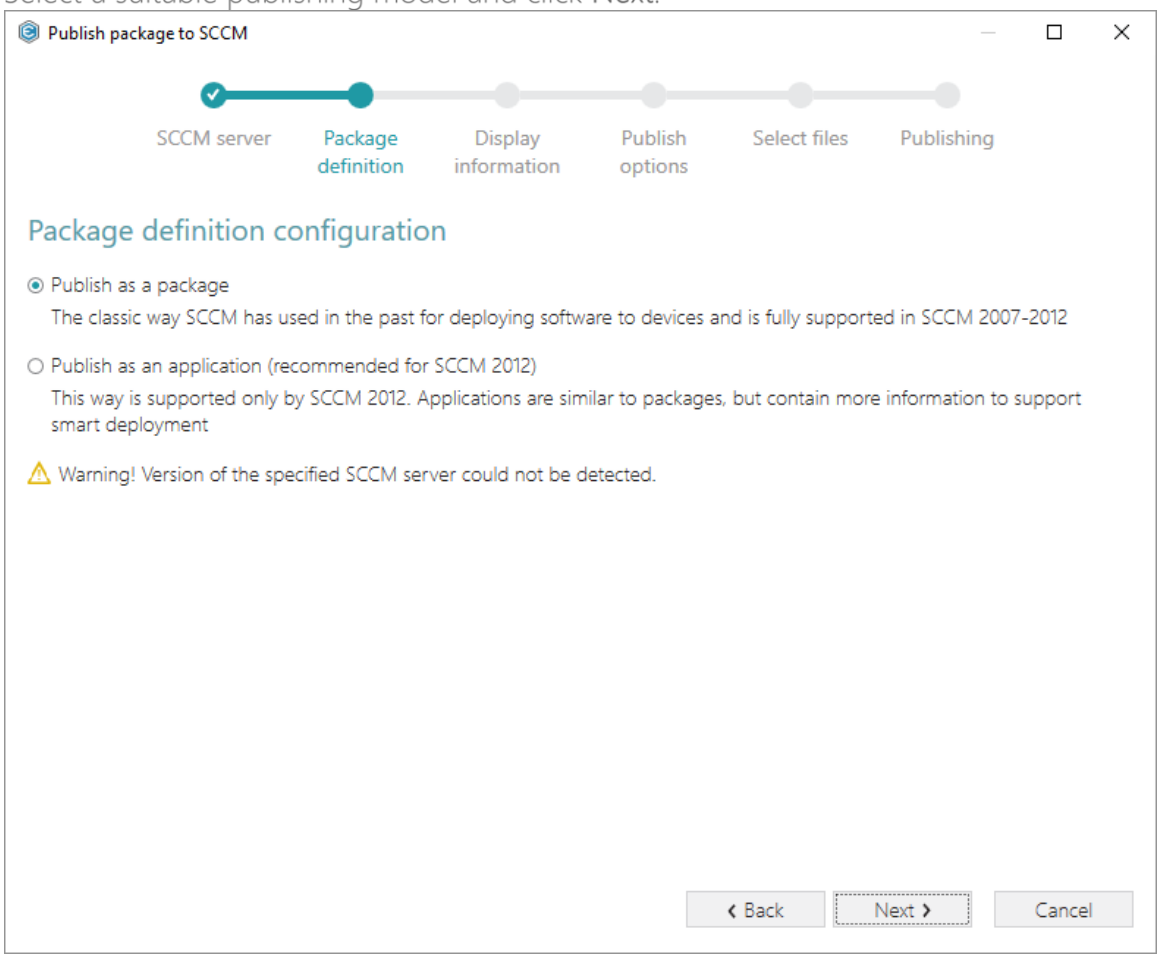


- Specify an IP address or a hostname of your SCCM server, user name and password for connecting to the SCCM. Enter a network share, where the package should be uploaded, and credentials for connecting to the network share. Click Next to choose a publishing model. Click

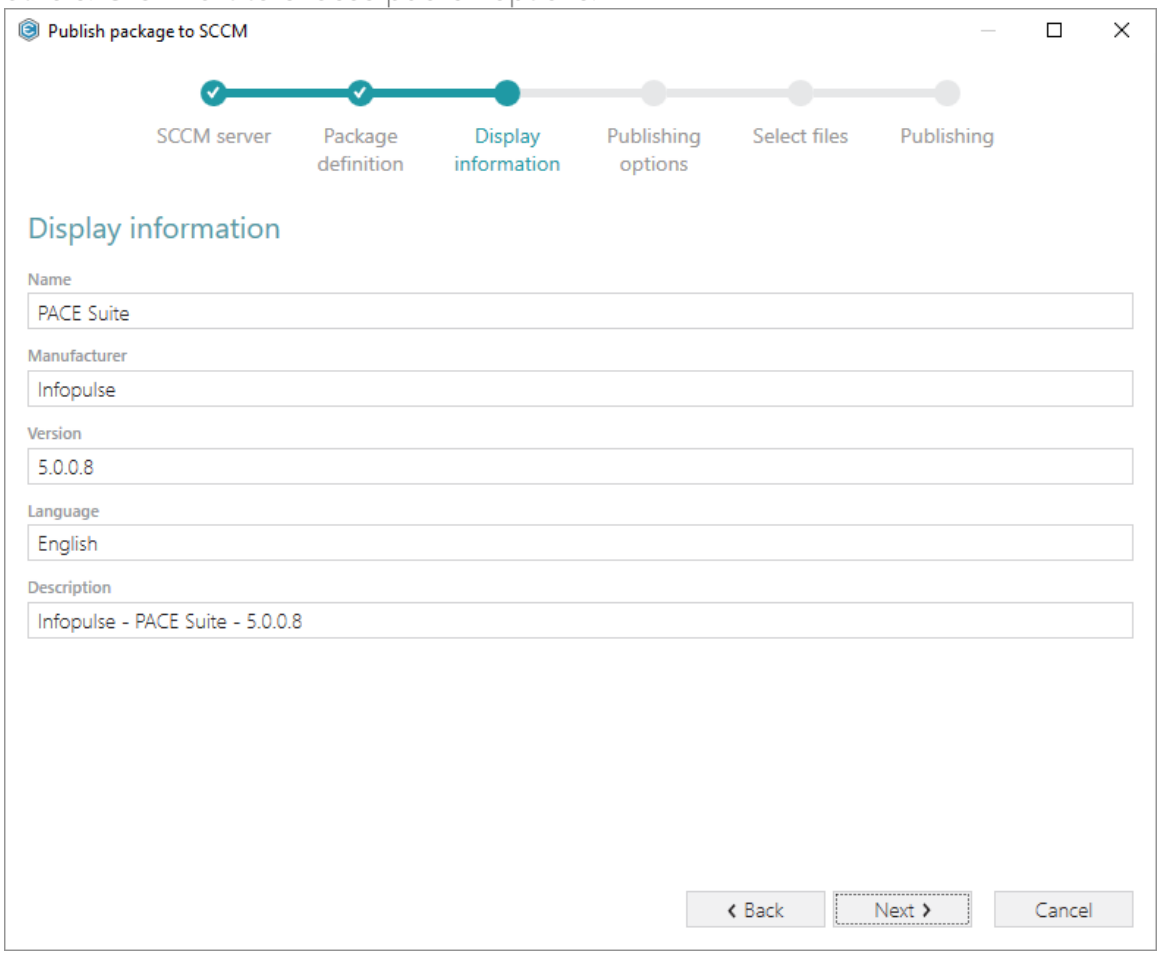
Test connections... to check connections to the SCCM server and network share.



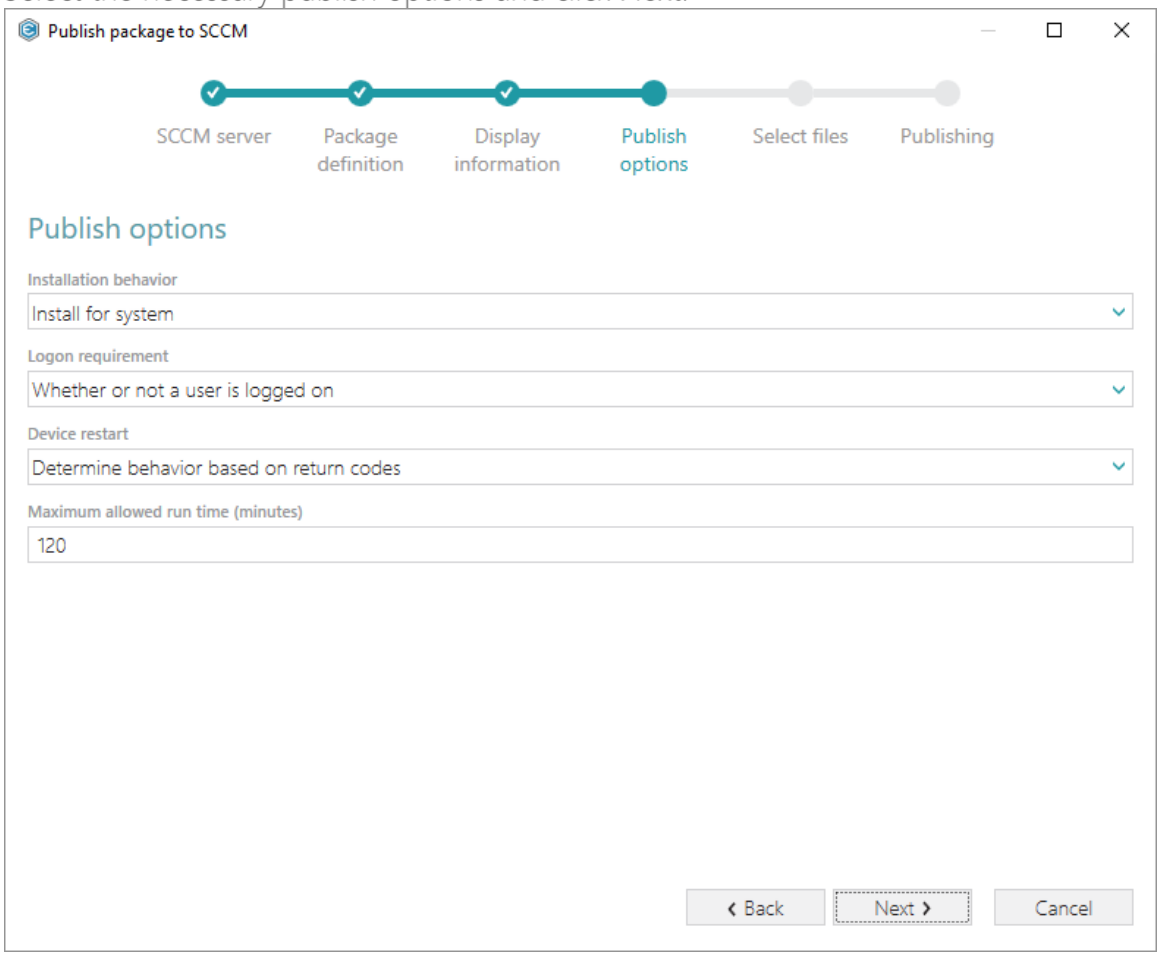
- Select a suitable publishing model and click Next.



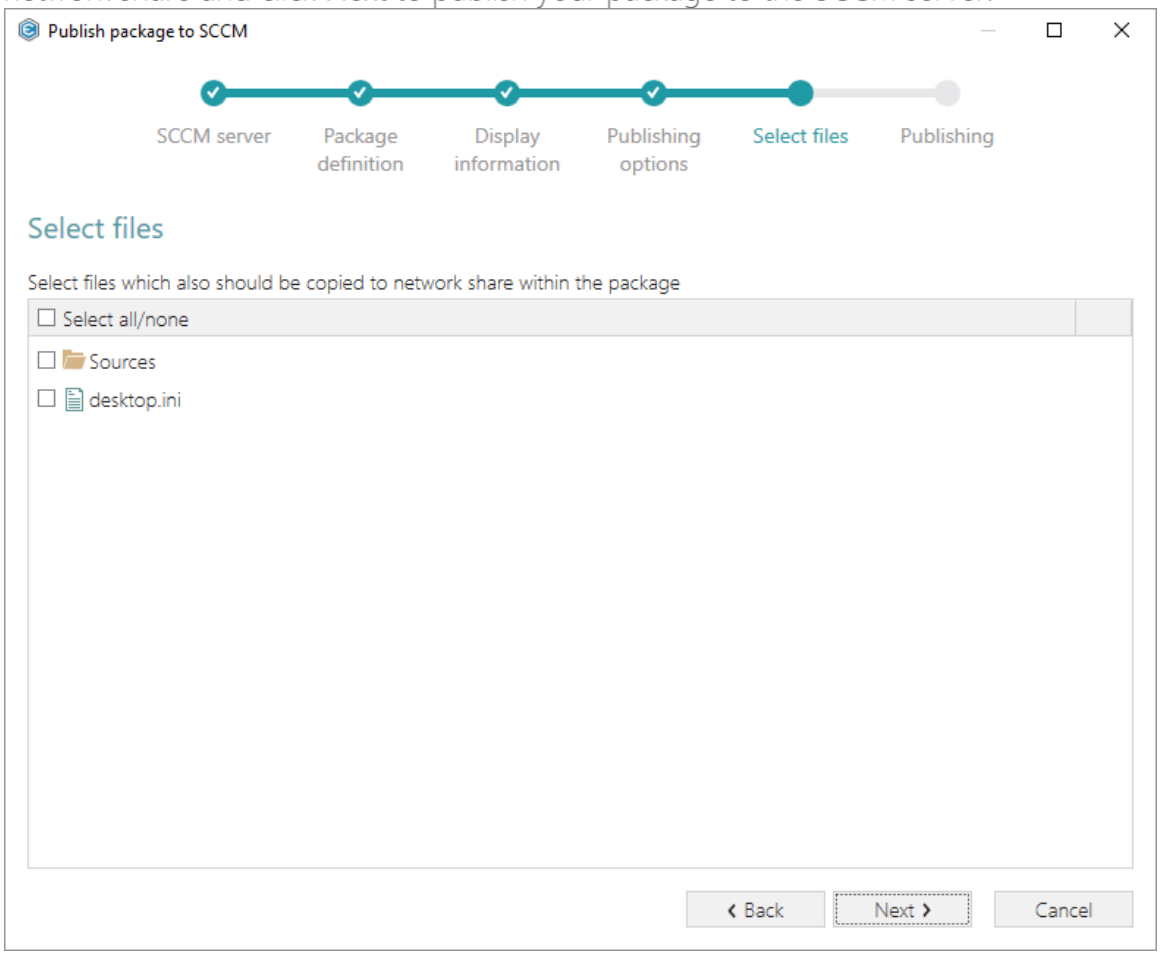
- Review and update package display information such as name, manufacturer, version and others. Click Next to choose publish options.



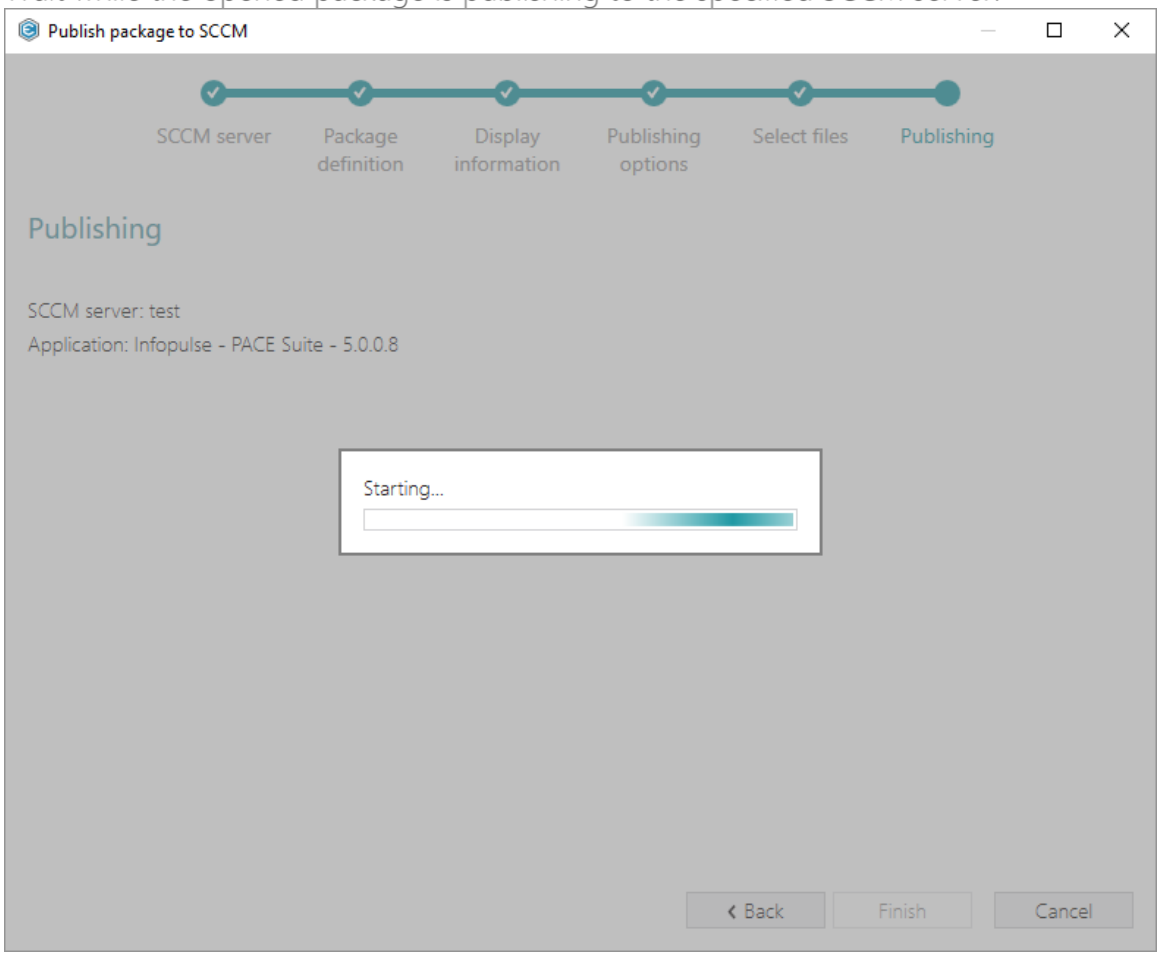
- Select the necessary publish options and click Next.



- Select which of files and folders, located in the package folder, should be copied to the network share and click Next to publish your package to the SCCM server.



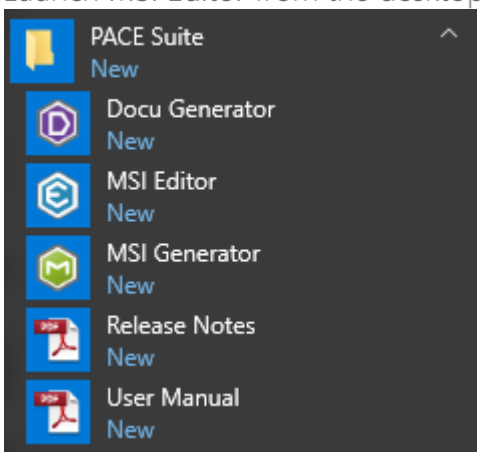
- Wait while the opened package is publishing to the specified SCCM server.



3.11 Generate MSI Package Report

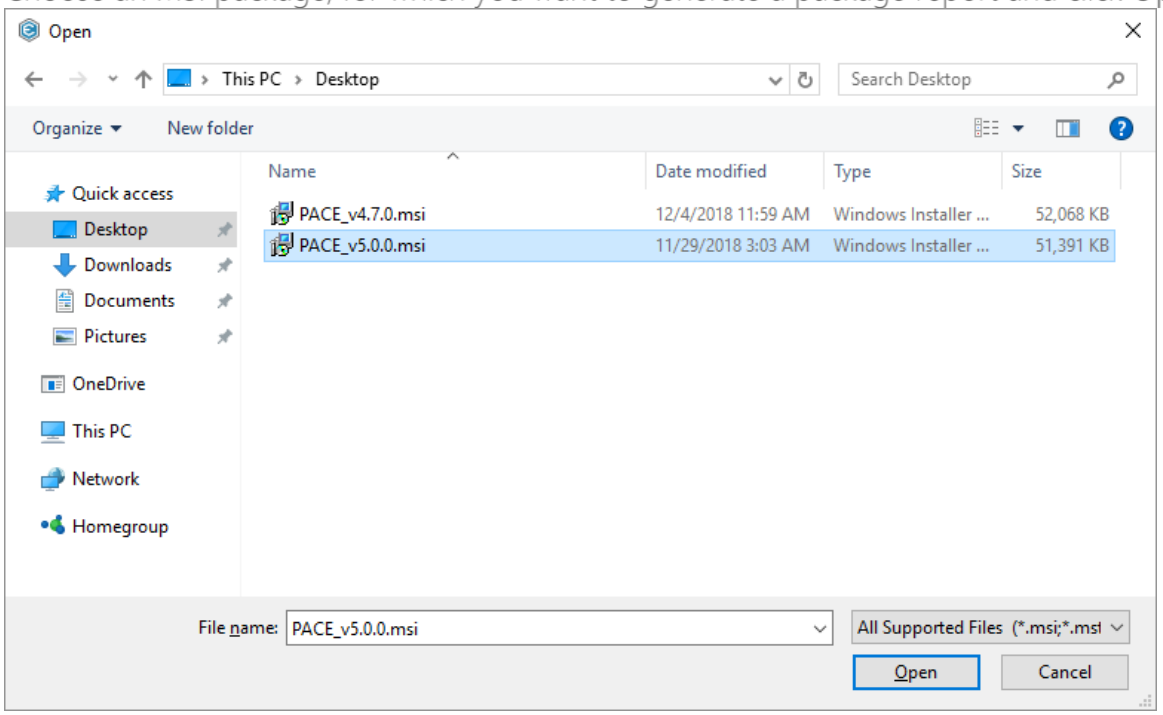
NOTE PACE Suite supports Word Document (.DOCX) and Excel Workbook (.XLSX) format types and allows generating a report with no Microsoft Office installed on the system.

- Launch MSI Editor from the desktop or the start menu shortcut.

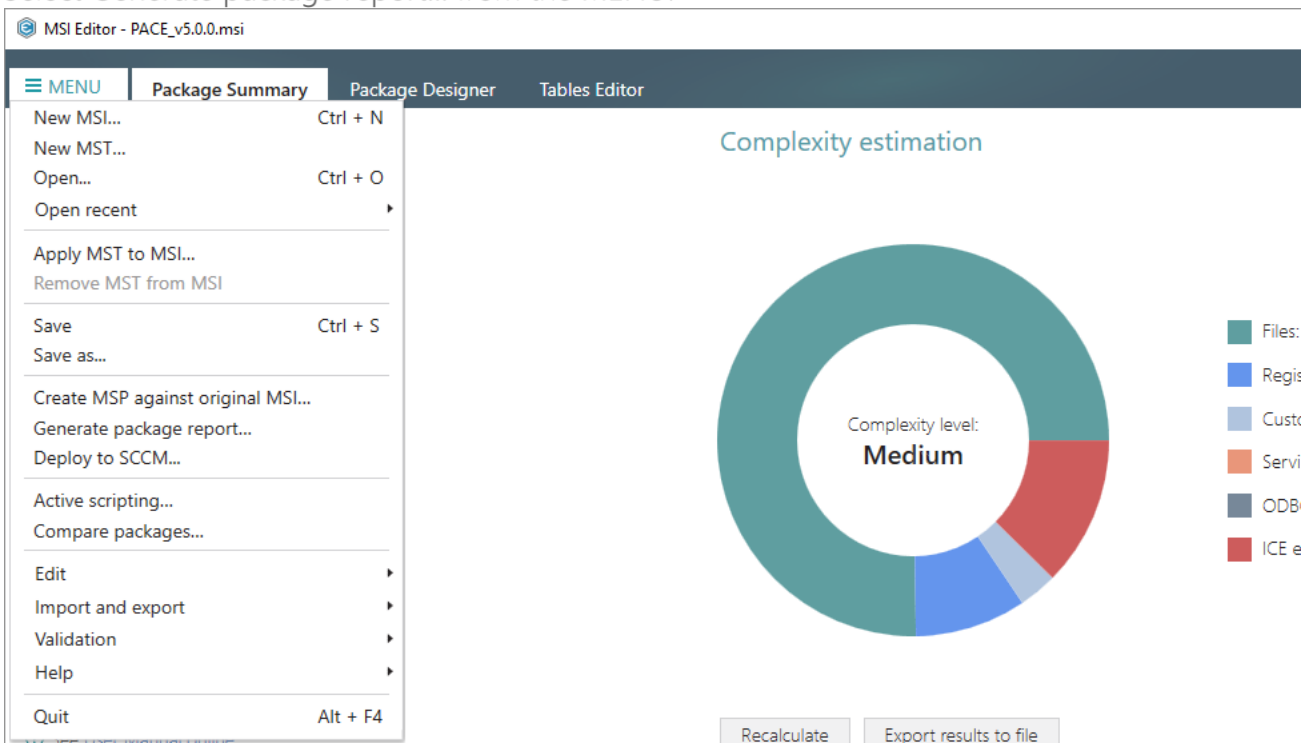


- Select Open... from MENU.

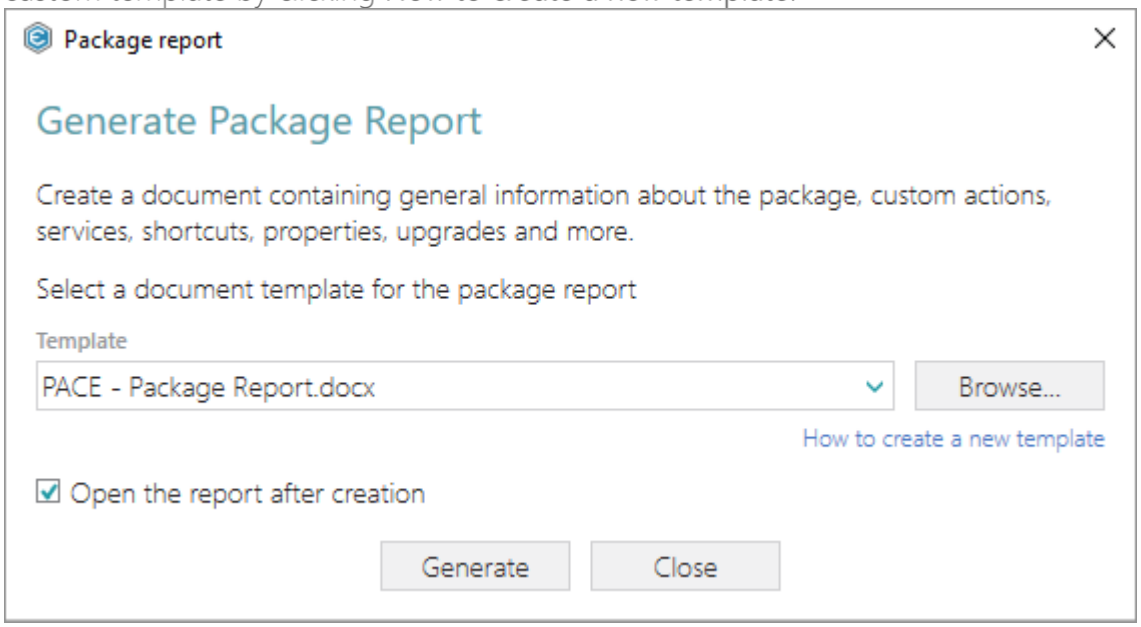
- Choose an MSI package, for which you want to generate a package report and click Open.



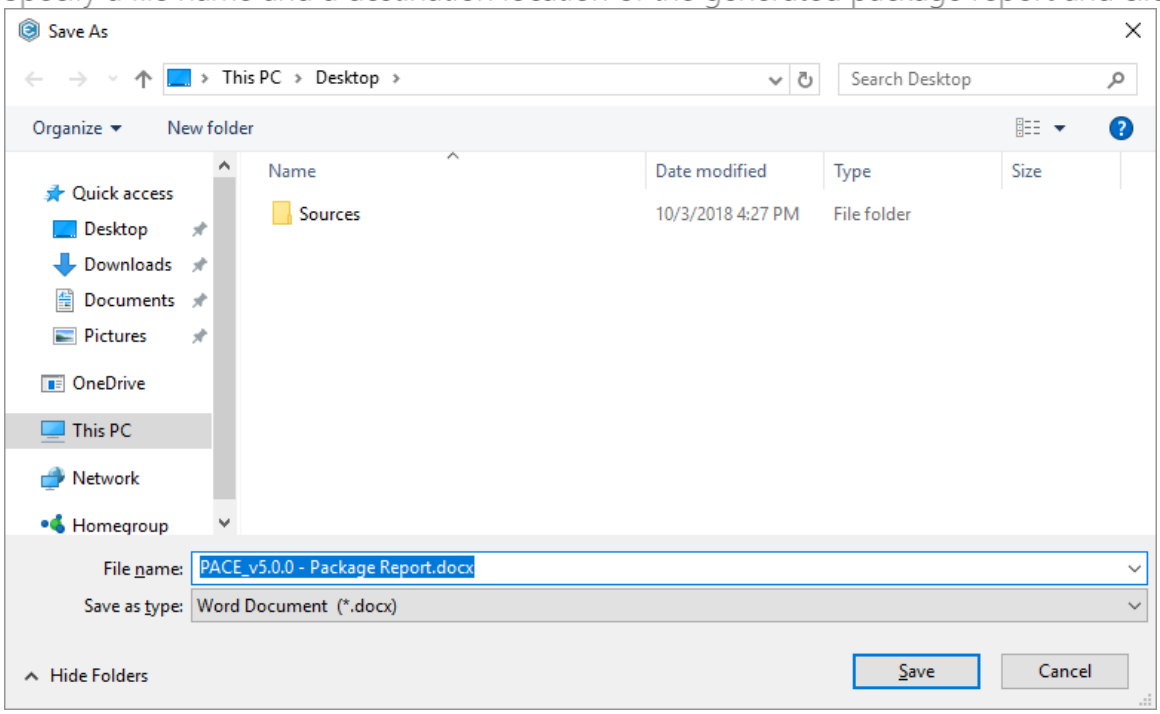
- Select Generate package report... from the MENU.



- Select a template for your report and click Generate. Find more information about creating a custom template by clicking How to create a new template.



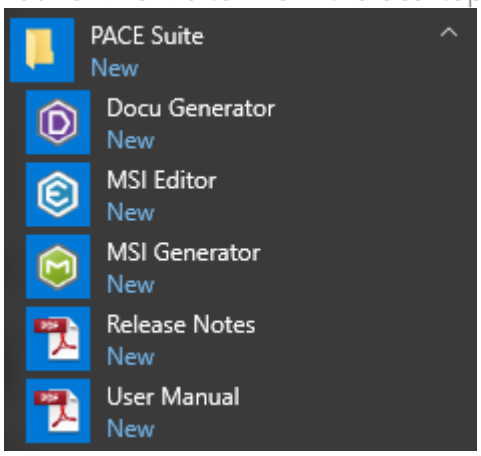
- Specify a file name and a destination location of the generated package report and click Save.



3.12 Validate MSI Package

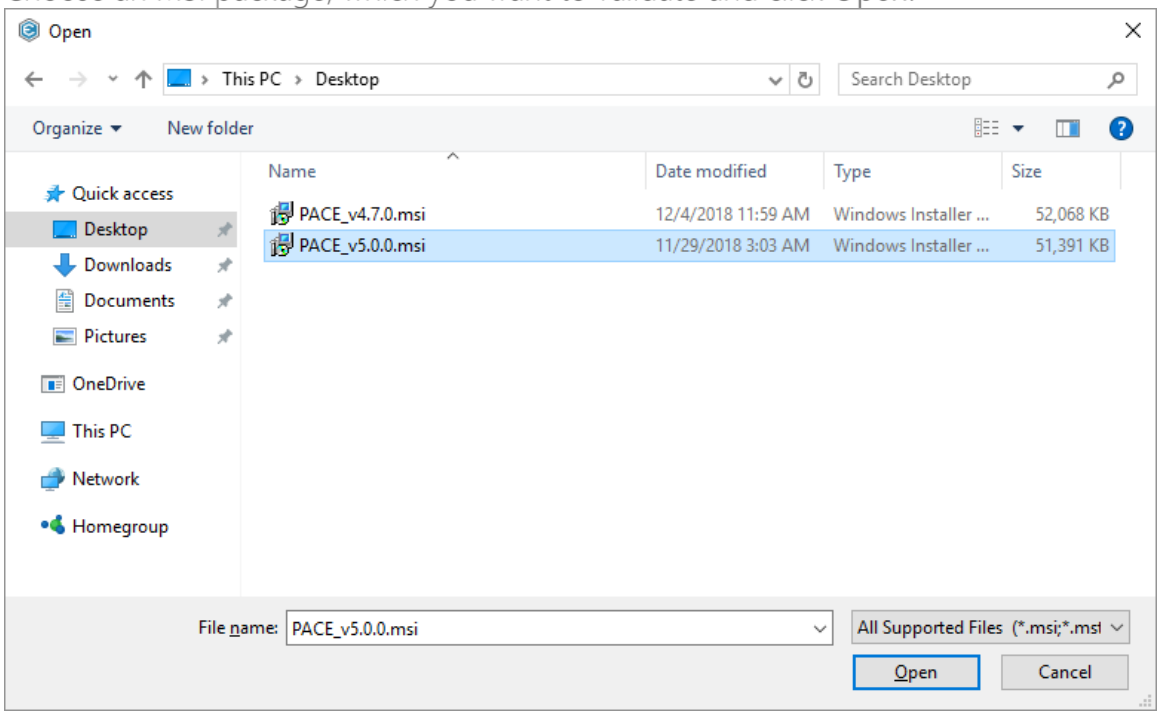
Validate the MSI package using the Internal Consistency Evaluators (ICE).

- Launch MSI Editor from the desktop or the start menu shortcut.

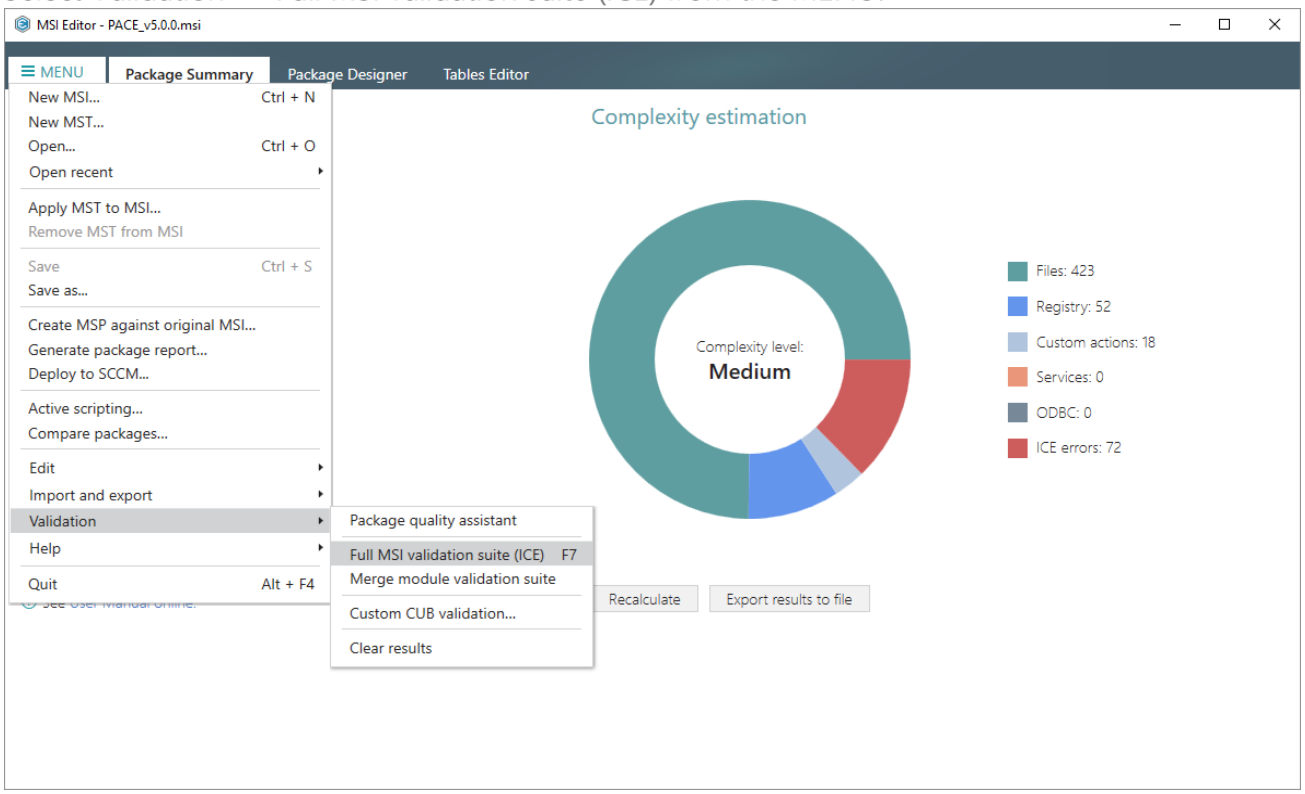


- Select Open... from MENU.

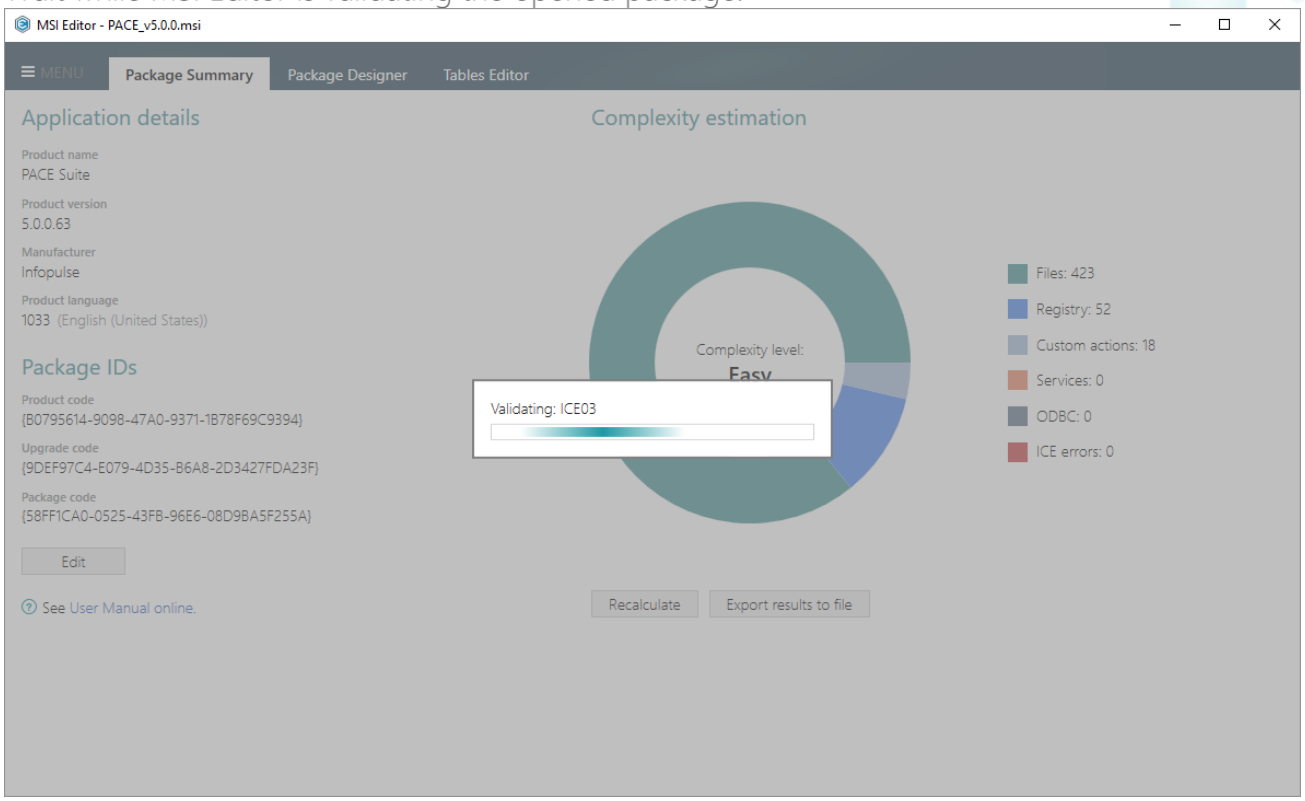
- Choose an MSI package, which you want to validate and click Open.



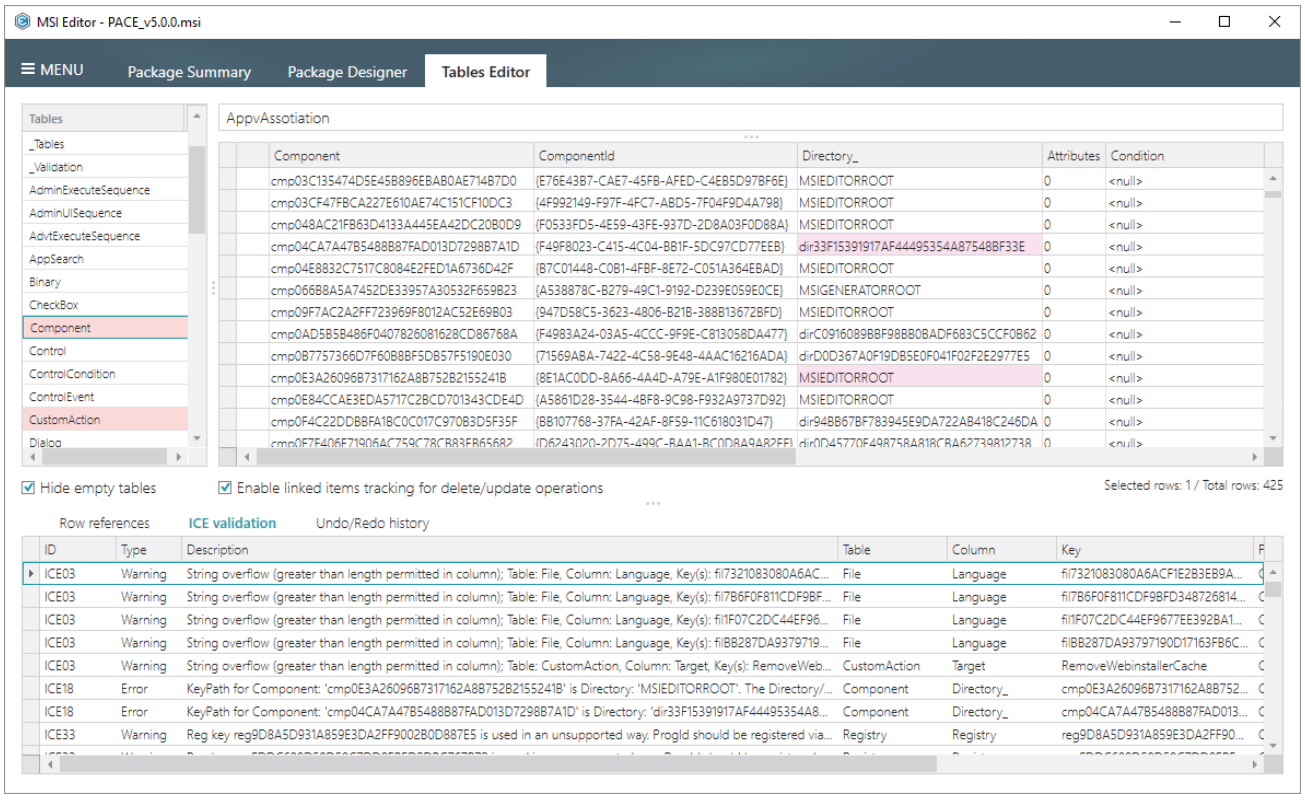
- Select Validation -> Full MSI validation suite (ICE) from the MENU.



- Wait while MSI Editor is validating the opened package.

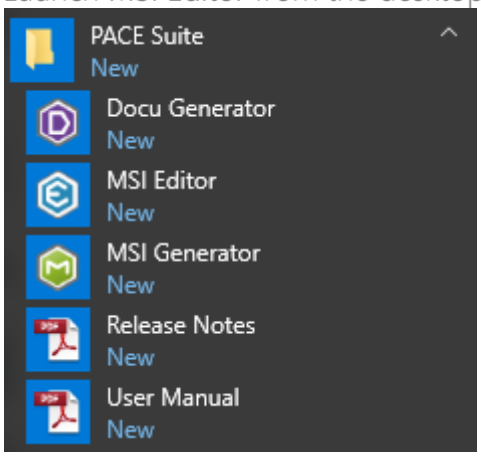


- Find the validation results in the ICE validation section of the Tables Editor tab.



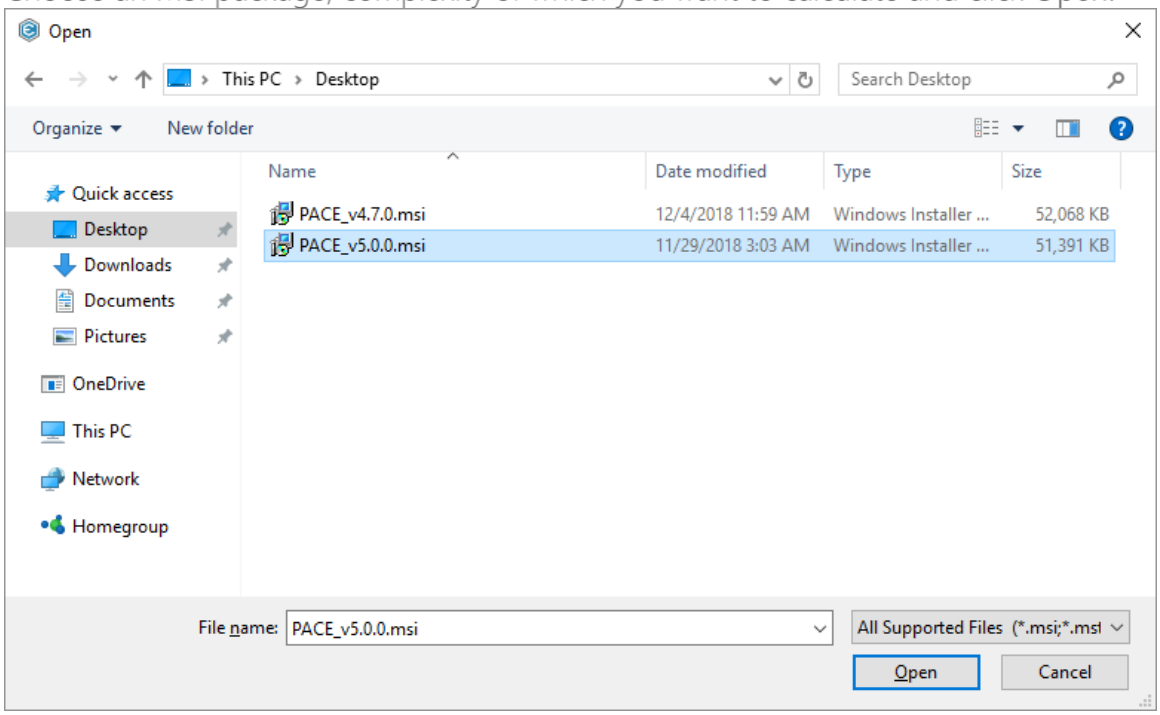
3.13 Get MSI Package Complexity

- Launch MSI Editor from the desktop or the start menu shortcut.

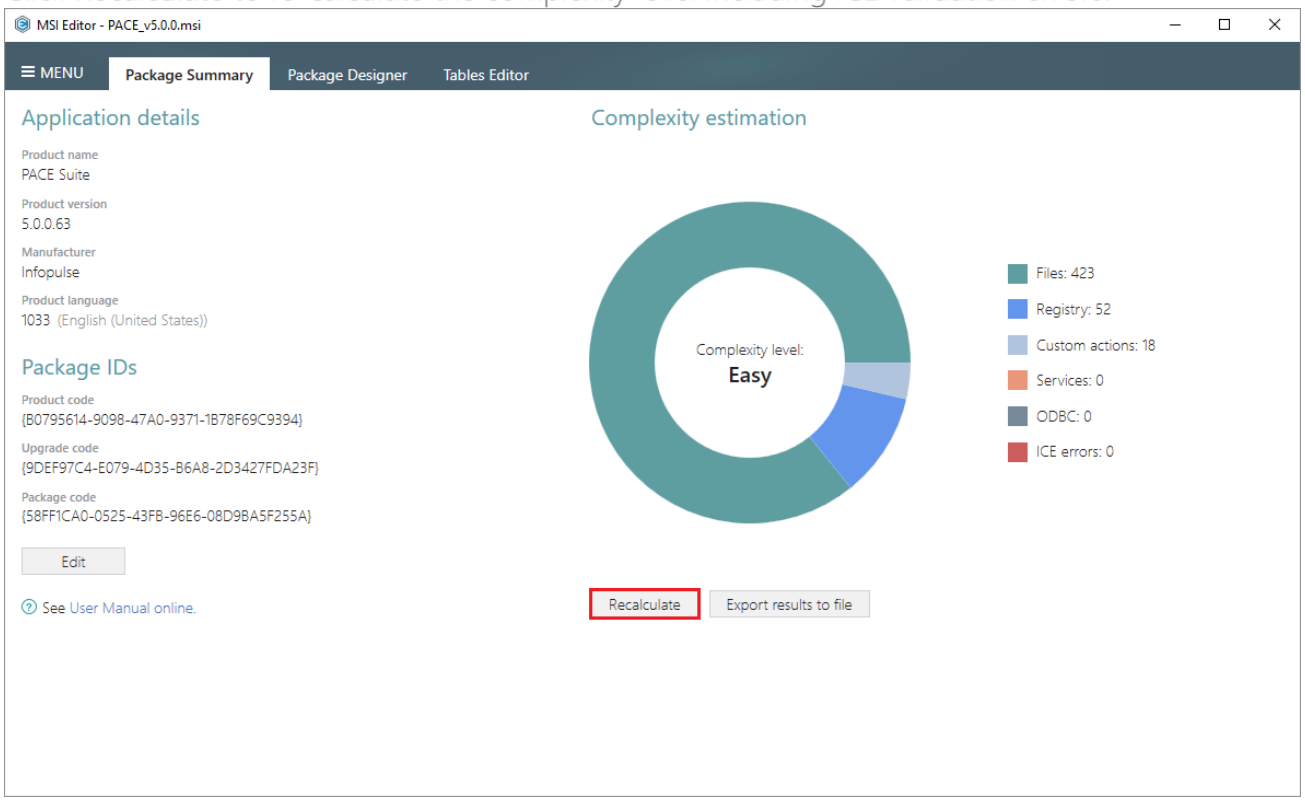


- Select Open... from MENU.

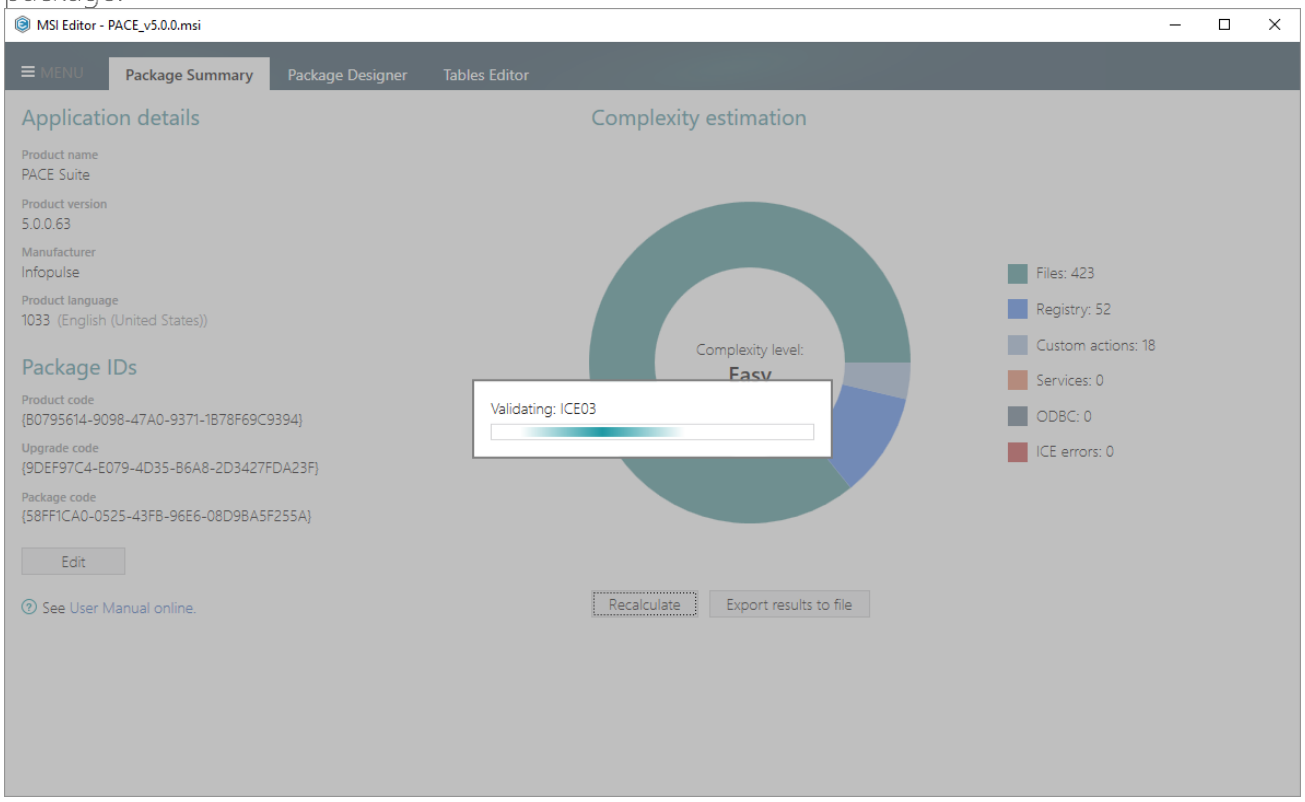
- Choose an MSI package, complexity of which you want to calculate and click Open.



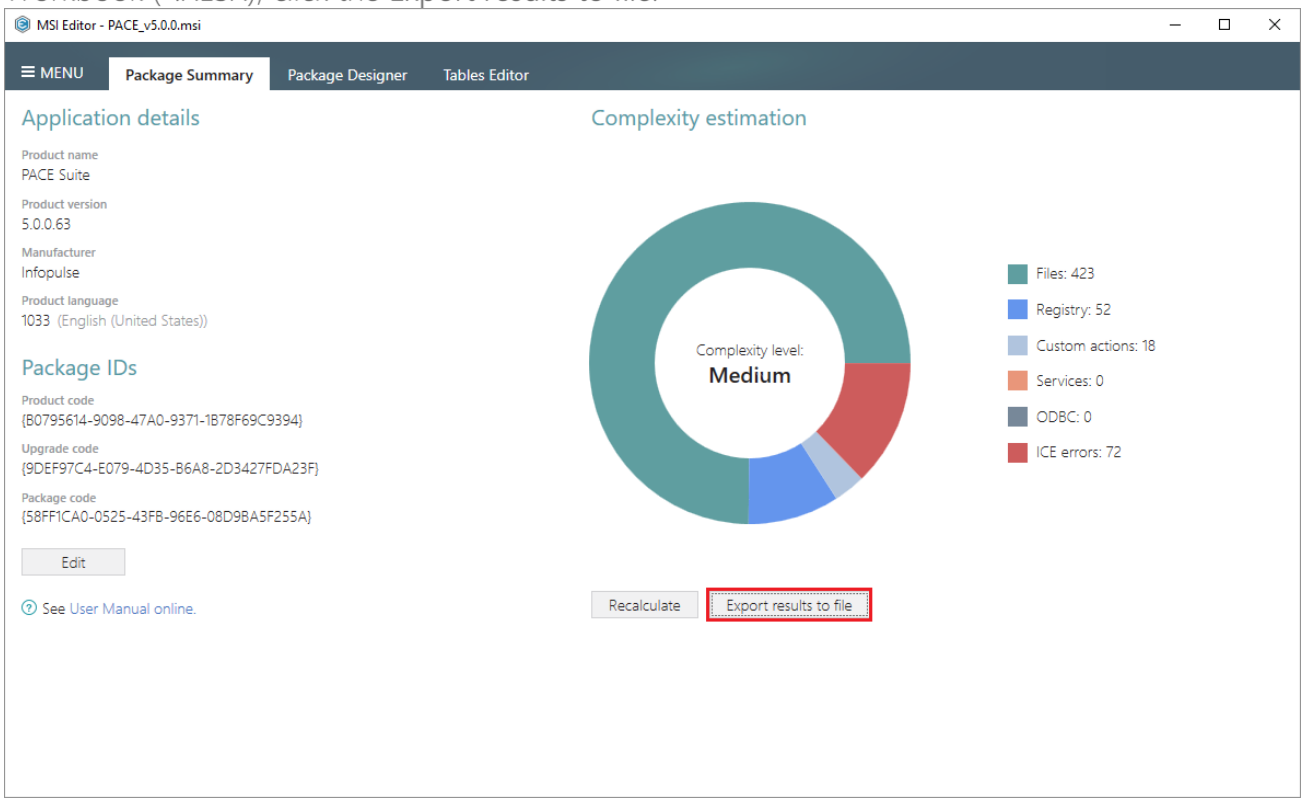
- Click Recalculate to re-calculate the complexity level including ICE validation errors.



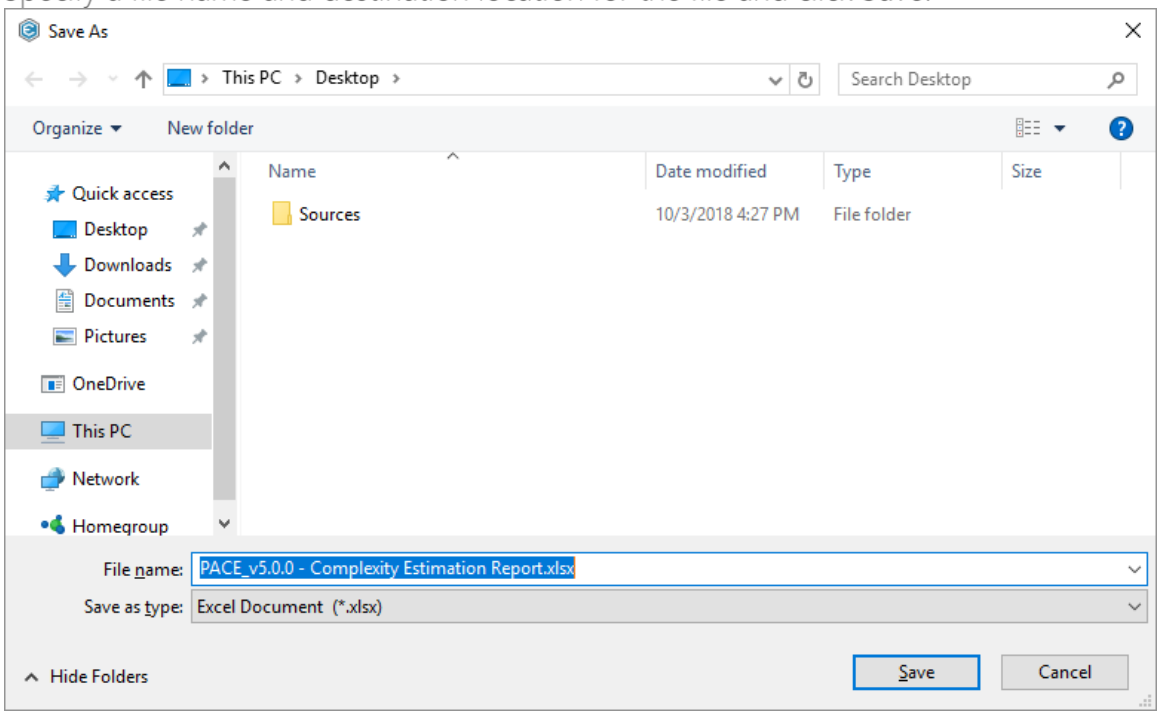
- Wait while MSI Editor is validating the package to get number of ICE errors of the opened MSI package.



- Find the complexity level and number of package resources on the Complexity estimation section of the **Package Summary** tab. To save the complexity estimation results to Excel Workbook (*.XLSX), click the **Export results to file**.



- Specify a file name and destination location for the file and click Save.

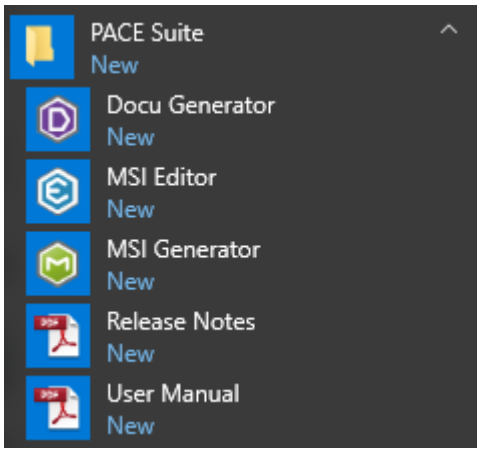


3.14 Create Discovery Documentation

Record all your on-screen actions into a nice looking document with screenshots and annotations. Such document could contain installation and configuration instructions and the like.

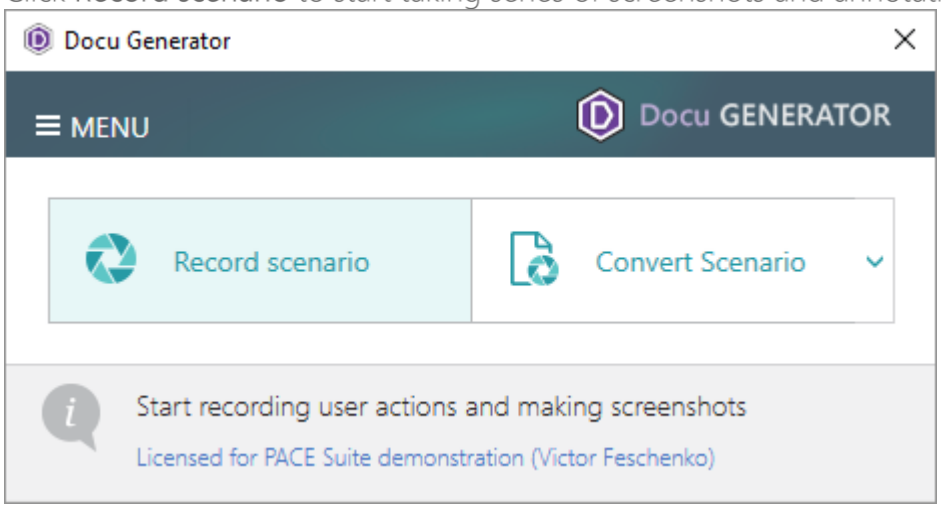
NOTE In order to take screenshots of application, which is launched with elevated permissions, launch Docu Generator with elevated permissions as well.

- Launch Docu Generator.



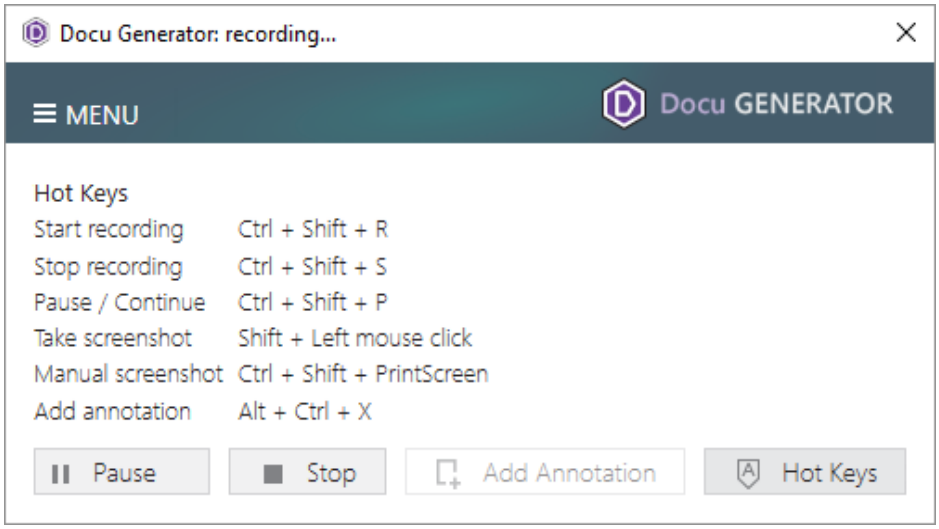
In order to launch Docu Generator with elevated permissions, press Shift and select Run as administrator from the context menu of the Docu Generator's shortcut.

- Click Record scenario to start taking series of screenshots and annotations.

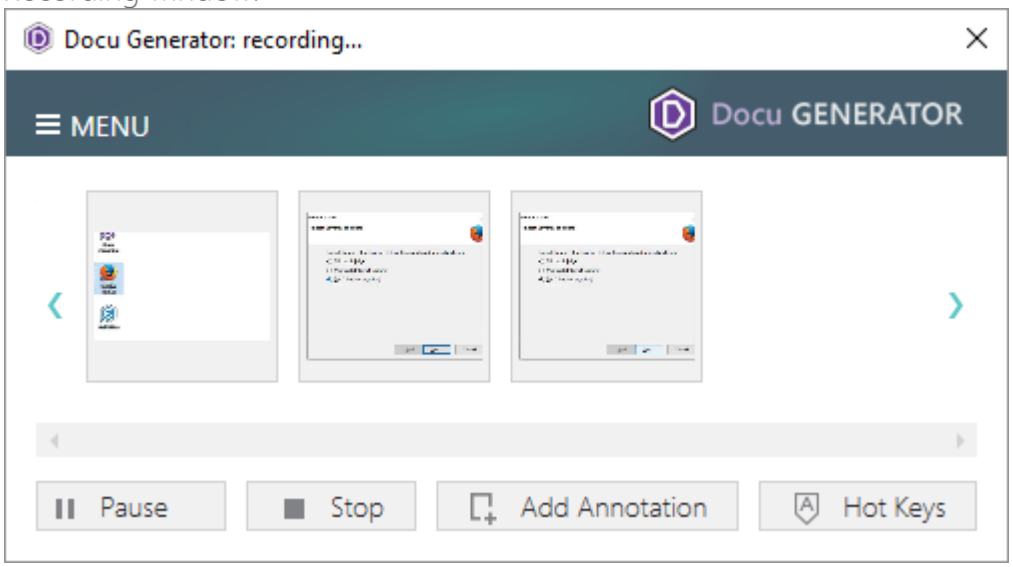


- Hover your mouse pointer on the Hot Keys button to check which of hot keys are used to take screenshots. Press the Shift key + Left mouse click to take a screenshot of an active window, dialog or screen area. Press Ctrl + Shift + PrintScreen, then click and drag the mouse pointer to select a rectangular area you want to grab. A screenshot is done when you release the

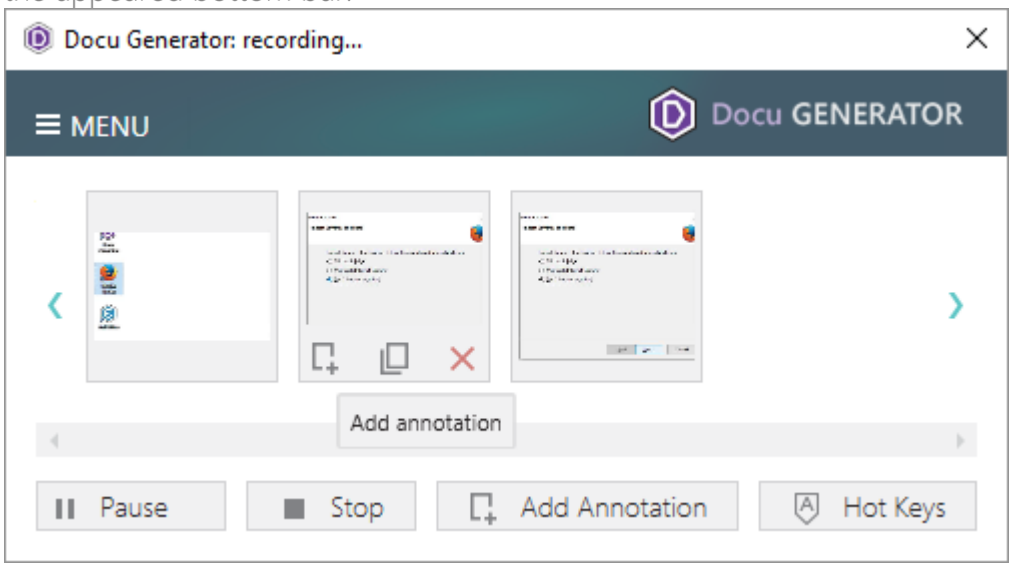
mouse button.



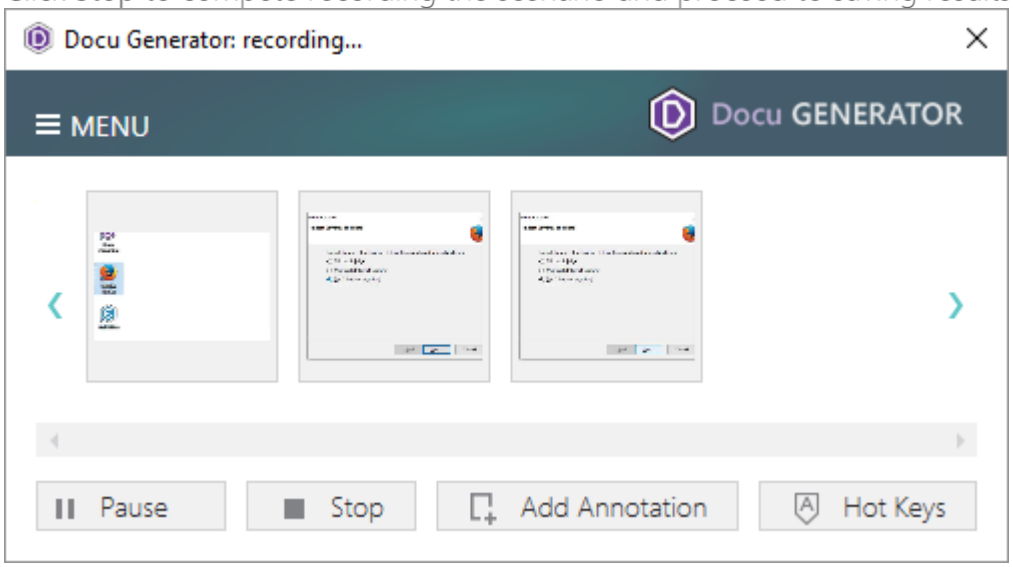
- Now, do the actions you want to record into the document. Launch executables, go through dialogs, select options and enter values just telling Docu Generator when to take another screenshot, as described above. A preview of each new screenshot immediately appears in the Recording window.



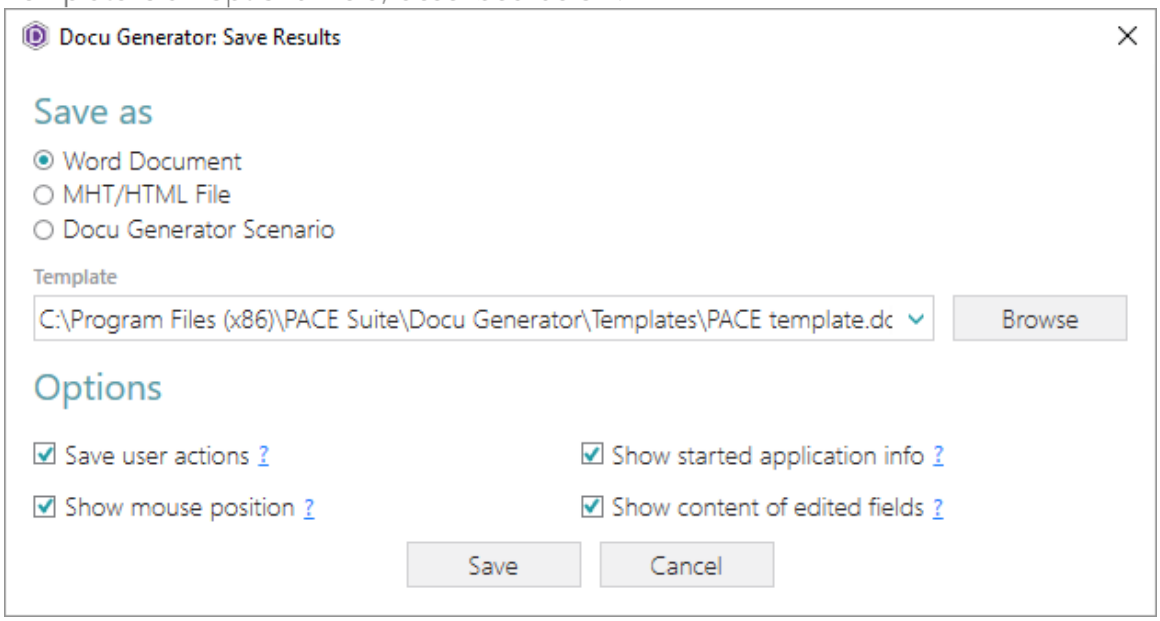
- In order to Add annotation, Copy to clipboard or Remove a screenshot, hover mouse pointer on a preview of the screenshot in the Recording window and select the respective function, in the appeared bottom bar.



- Click Stop to compete recording the scenario and proceed to saving results.



- Select an output format of the document, saving options and click Save to create document. A Template is an optional field, described below.



Find options description below in table.

Option	Description
Save user actions	This option adds the information about names of controls (which were clicked on) and names of the captured windows and dialog boxes to the document.
Show mouse position	This option adds information about mouse pointer position and highlights it with a red circle on screenshots to the document.
Show started application info	This option adds information about paths and command line arguments of the launched application to the document.
Show contents of edited fields	This option adds contents of input fields of the captured windows and dialog boxes to the document.
Template	Docu Generator supports Microsoft Word (DOT/DOTX) and HTML/MHT templates to be used for document creation. To use a template, specify its file path in the "Save Results" window. In order to mark a position where screenshots and comments are to be placed to, add the following textual placeholder into a template: [SCR] You can specify placeholder [SCR] in the existing table in a template. In this case, the placeholder points to a cell where screenshots should start, while a column to the right of it will be populated with comments. If the placeholder is located outside a table, then a new table for screenshots and comments will be created. A Docu Generator-default table style will be used.

If a placeholder is not found, screenshots and comments will be appended to the end of the template. You will be warned about this in the Save Results window

NOTE To let you return to your scenario later, Docu Generator always creates DSCN file along with the output file in an output folder.

4 References

4.1 Contact

Product website	http://pacesuite.com
Support email	pace-support@infopulse.com
Support Portal	http://pacesuite.com/support
Educational videos	YouTube channel

4.2 Social Networks

ITNINJA The IT Professional's Software Community	itninja.com/pace
	@pacesuite
	fb.com/pacesuite
	https://www.linkedin.com/groups/PACE-Suite-APS-community-4765703