# GETTING STARTED

PACE Suite 5.0

Document version: 20181211

# Table of Contents

1 Introduction ..... 6
1.1 About This Document ..... 6
1.2 Glossary ..... 6
2 PACE Overview ..... 7
2.1 Installation & Activation ..... 8
2.1.1 Software Prerequisites ..... 8
2.1.2 Hardware Prerequisites ..... 8
2.1.3 Installing PACE Suite ..... 9
2.1.4 Activating PACE Suite ..... 14
3 How-To Manuals ..... 16
3.1 Create MSI ..... 16
3.1.1 Create Blank MSI ..... 16
3.1.2 Generate MSI from Scratch ..... 38
3.1.3 Convert (Repackage) EXE to MSI ..... 47
3.1.4 Capture System Changes to MSI ..... 67
3.2 Create MST ..... 84
3.2.1 Create Blank MST ..... 84
3.2.2 Create Response MST ..... 108
3.2.3 Capture System Changes to MST ..... 116
3.3 Create MSP ..... 134
3.3.1 Create MSP Based On Difference ..... 134
3.3.2 Save Changes to MSP ..... 141
3.4 Create APPX ..... 168
3.4.1 Create APPX from Scratch ..... 168
3.4.2 Convert (Repackage) EXE, MSI to APPX ..... 177
3.5 Create APPV ..... 192
3.5.1 Create APPV from Scratch ..... 192
3.5.2 Convert (Repackage) EXE, MSI to APPV 5.x ..... 203
3.6 Create THINAPP ..... 221
3.6.1 Create THINAPP from Scratch ..... 221
3.6.2 Convert (Repackage) EXE, MSI to THINAPP ..... 228

3.7 Edit MSI/MST ..... 241
3.7.1 Pre-Condition ..... 242
3.7.2 Files and Folders ..... 244
3.7.2.1 Add System Folder ..... 245
3.7.2.2 Add Subfolder ..... 247
3.7.2.3 Import Files and COM ..... 249
3.7.2.4 Change Compression Type ..... 254
3.7.2.5 Set Permissions ..... 258
3.7.2.6 Remove Permissions ..... 262
3.7.2.7 Move Files, Folders ..... 265
3.7.2.8 Set INSTALLDIR ..... 267
3.7.2.9 Edit INI File ..... 269
3.7.2.10 Edit File Properties ..... 271
3.7.2.11 Edit Folder Properties ..... 273
3.7.3 Shortcuts ..... 275
3.7.3.1 Add Shortcut ..... 276
3.7.3.2 Import Shortcuts ..... 283
3.7.3.3 Move Shortcut ..... 285
3.7.3.4 Edit Shortcut Properties ..... 288
3.7.4 Registry ..... 291
3.7.4.1 Add Registry Key ..... 292
3.7.4.2 Add Registry Value ..... 294
**3.7.4.2.1 Add ActiveSetup ..... 297**
3.7.4.3 Add Value/Key Removal ..... 302
3.7.4.4 Import Registry ..... 304
3.7.4.5 Export Registry ..... 308
3.7.4.6 Set Permissions ..... 309
3.7.4.7 Remove Permissions ..... 313
3.7.4.8 Move Value, Key ..... 316
3.7.4.9 Edit Registry Key ..... 318
3.7.4.10 Edit Registry Value ..... 320
3.7.4.11 Edit Value/Key Removal ..... 324

3.7.5 Custom Actions ..... 326
3.7.5.1 Add Standard Actions ..... 326
3.7.5.2 Add Custom Action ..... 330
**3.7.5.2.1 Run PowerShell script ..... 335**
3.7.5.3 Edit Custom Action ..... 345
3.7.6 Drivers ..... 350
3.7.6.1 Add Driver ..... 350
3.7.6.2 Edit Driver ..... 355
3.7.7 Features and Components ..... 357
3.7.7.1 Add Feature ..... 358
3.7.7.2 Add Component ..... 360
3.7.7.3 Move Feature, Component ..... 362
3.7.7.4 Edit Feature Properties ..... 364
3.7.7.5 Edit Component Properties ..... 366
3.7.7.6 Find Empty Components ..... 370
3.7.7.7 Review Component Resources ..... 372
3.7.8 Properties ..... 374
3.7.8.1 Add Property ..... 375
3.7.8.2 Edit Property ..... 376
3.7.9 Summary Information ..... 378
3.7.9.1 Edit MSI/MST Summary Information ..... 378
3.7.10 Upgrades ..... 380
3.7.10.1 Integrate Upgrades ..... 380
3.7.11 Tables ..... 382
3.7.11.1 Add Tables ..... 383
3.7.11.2 Drop Tables ..... 385
3.7.11.3 Import Tables (IDT) ..... 389
3.7.12 Merge Modules ..... 392
3.7.12.1 Import Merge Modules (MSM) ..... 392
3.8 Edit Project ..... 397
3.8.1 Pre-Condition ..... 397
3.8.2 File and Folders ..... 399

3.8.2.1 Exclude Files and Folders ..... 399
3.8.2.2 Include Files and Folders ..... 400
3.8.2.3 Add Custom Folder ..... 401
3.8.2.4 Import Files ..... 403
3.8.3 Registry ..... 405
3.8.3.1 Exclude Registry Entries ..... 405
3.8.3.2 Include Registry Entries ..... 406
3.8.3.3 Import Registry ..... 407
3.8.3.4 Export Registry ..... 409
3.8.4 Shortcuts ..... 410
3.8.4.1 Import Shortcut ..... 410
3.9 Edit APPV ..... 413
3.10 Deploy (Publish) to SCCM ..... 422
3.11 Generate Package Report ..... 429
3.12 Validate Package ..... 432
3.13 Get Package Complexity ..... 435
3.14 Create Discovery Documentation ..... 439
4 References ..... 443
4.1 Contact ..... 443
4.2 Social Networks ..... 443

# 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. |

# 2 PACE Overview

PACE Suite consists of the following components:

| Tool name | Major functions |
|---|---|
| | • Repackage any existing installation into Windows Installer (MSI), UWP app package (APPX) or a virtualization format (Microsoft App-V 5.x, and VMware ThinApp).<br>• Auto-detection of embedded installers.<br>• Create a response transform<br>• Tune any existing Windows Installer package by applying your settings and saving them as a Windows Installer transform (MST).<br>• Include the excluded files/registry back to a package. |
| | • Manage the contents of an MSI in a tree view<br>• Import the desired resources<br>• Manage Custom Actions<br>• Integrate Device Drivers<br>• Use a smart and advanced MSI database editor, with formatted string autocompleting, Excel-like formula bar, row reference tracking, and more.<br>• Create patches (MSP)<br>• Manage permission settings for file system and registry<br>• Handle upgrades easily – just let MSI Editor know which MSI you want to be upgraded at runtime.<br>• Publish your applications to Microsoft SCCM 2007, 2012, 2016.<br>• See the estimated complexity of an app right away.<br>• Generate package documentation containing the details about your package and configuration.<br>• 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<br><br>NOTE Building Universal Windows Platform app packages (APPX) is available under Windows 10 or Windows Server 2016 or newer. |
| Middleware | Microsoft .NET Framework 4.6.2 (https://www.microsoft.com/en-us/download/details.aspx?id=53345) or higher.<br><br>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.<br><br>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

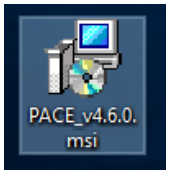
Ensure that a target system fulfils the minimum hardware conditions:

| | |
|---|---|
| Processor | 32-bit (x86) or 64-bit (x64) processor at 2 GHz or greater<br><br>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<br>NOTE Additional storage is required for your projects and packages and depends on their size and quantity. |
| Display | 1280 x 768 resolution or higher. |

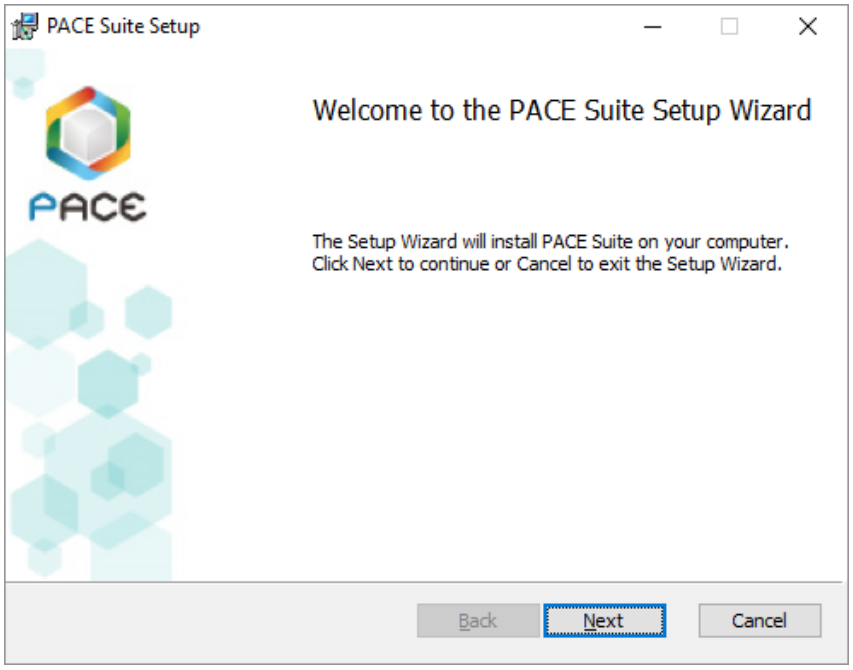
### 2.1.3 Installing PACE Suite

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.

[1]. Run the downloaded PACE Suite installater.

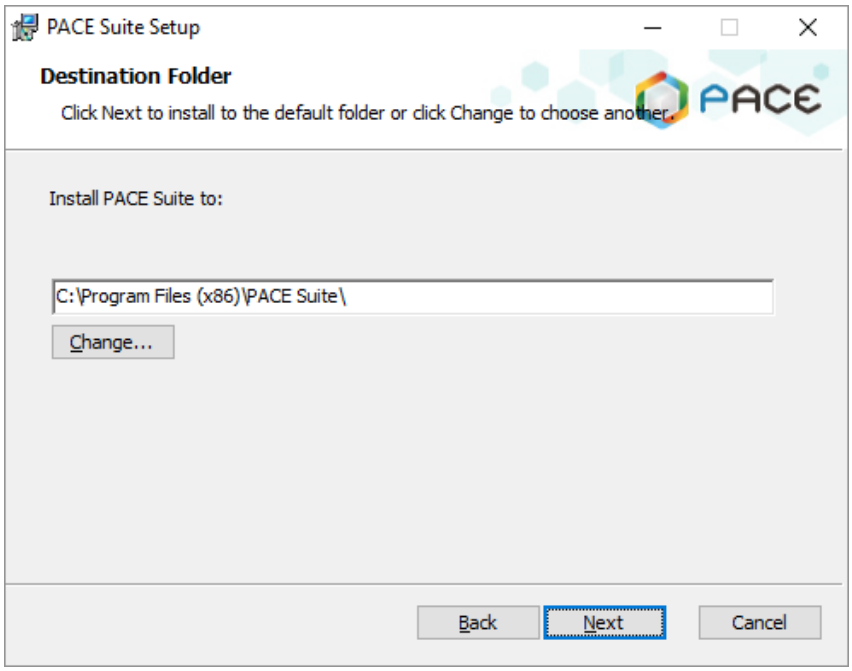



[2]. Click Next.

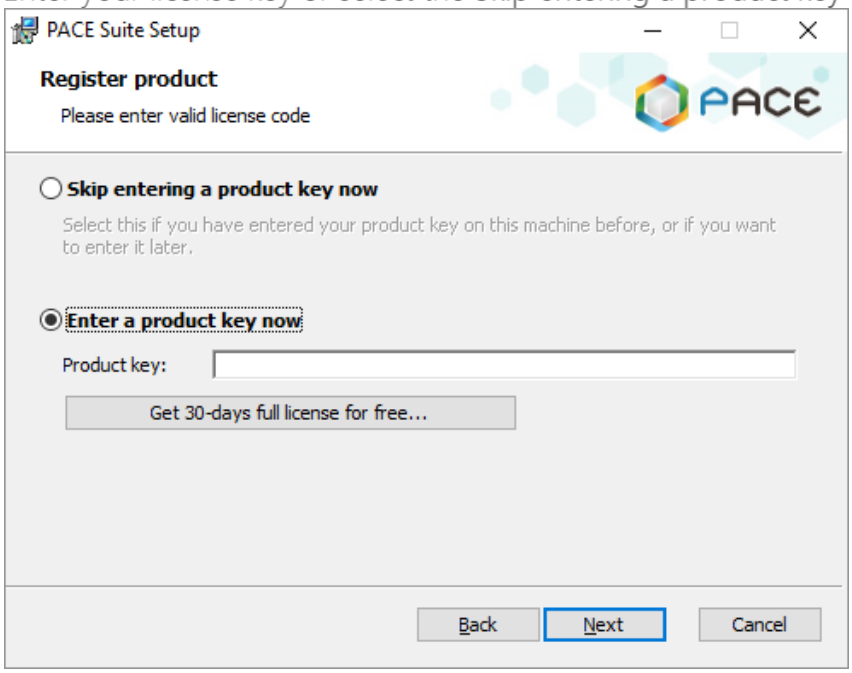



[3]. Read and accept the END USER LICENSE AGREEMENT and then click Next.

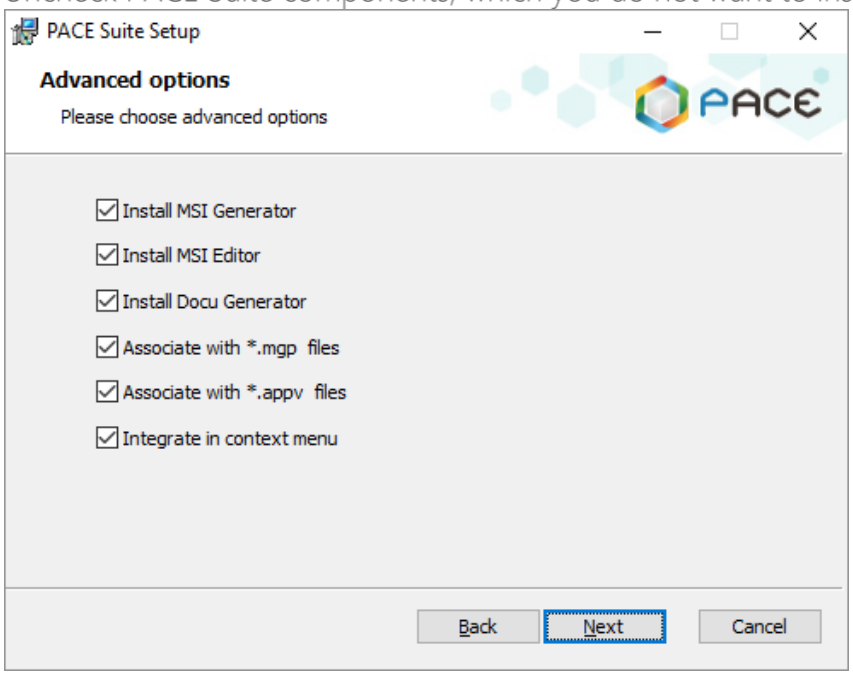
[4]. Leave the default installation location or choose an alternative one and click **Next**.



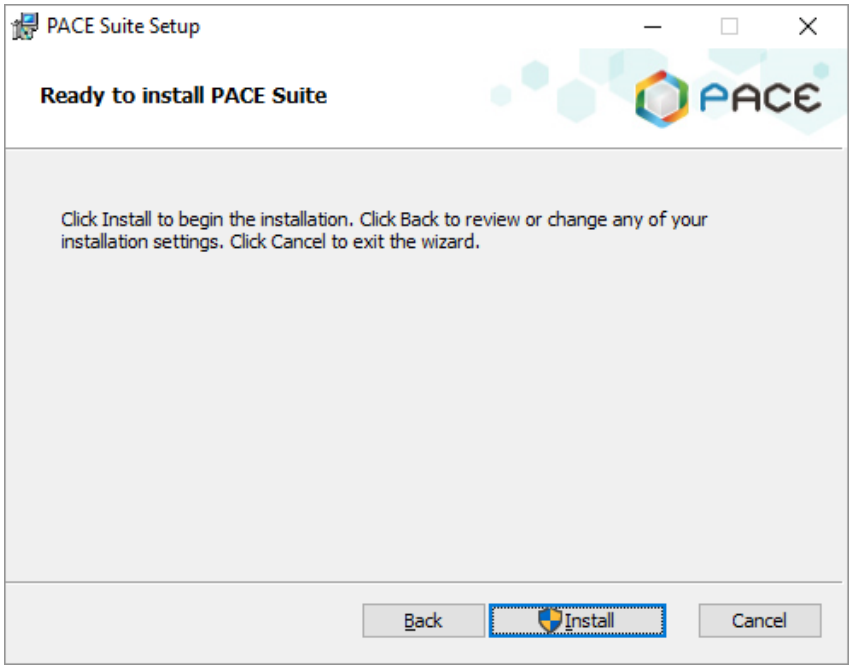

[5]. Enter your license key or select the **Skip entering a product key now** option and click **Next**.



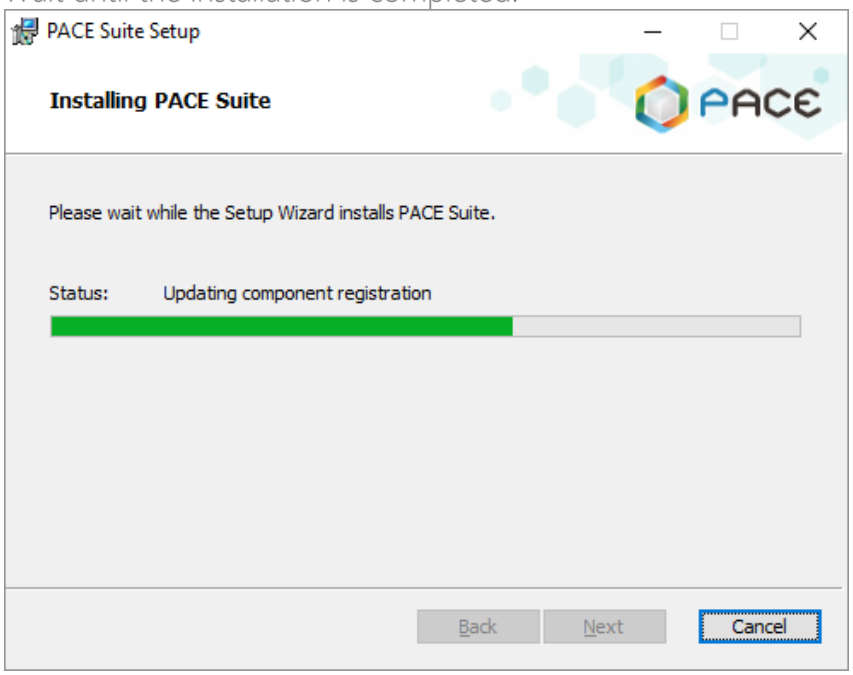
[6]. Uncheck PACE Suite components, which you do not want to install, and click **Next**.



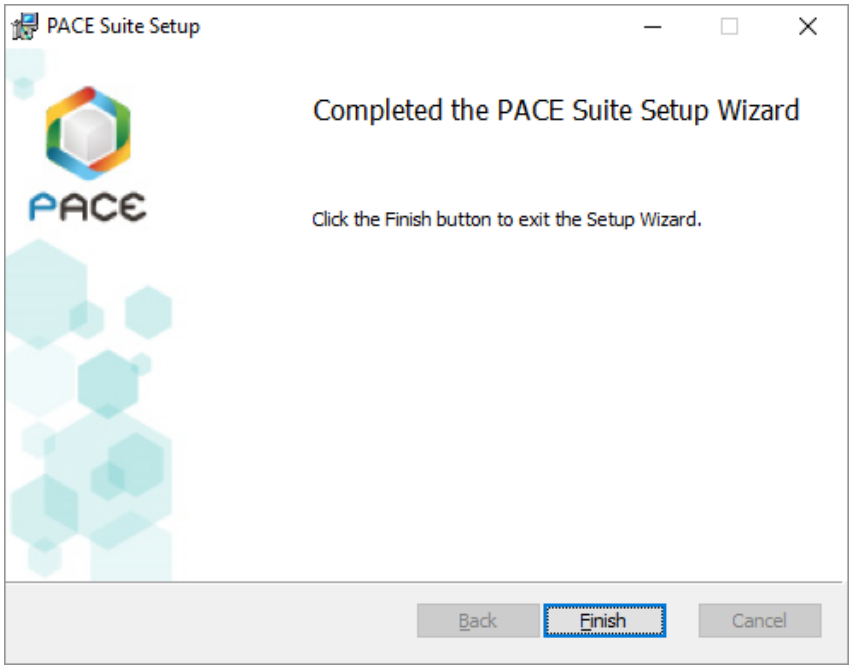

[7]. Click **Install**.



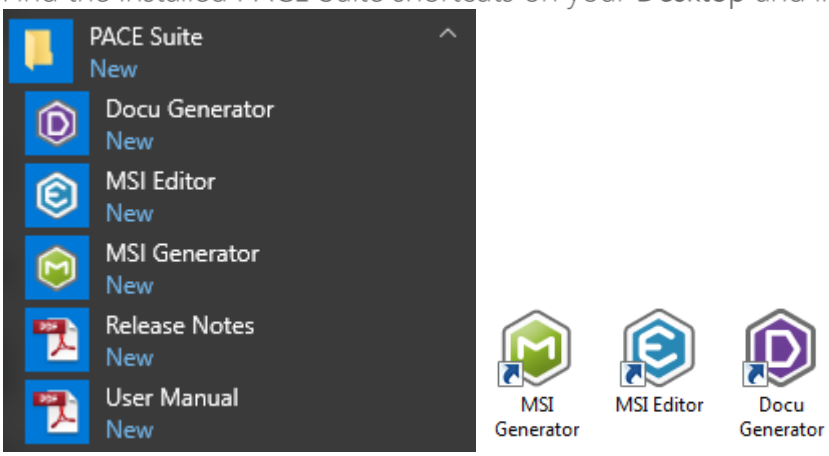
[8]. Wait until the installation is completed.




[9]. Click Finish.



[10]. Find the installed PACE Suite shortcuts on your **Desktop** and in the **Start Menu**.

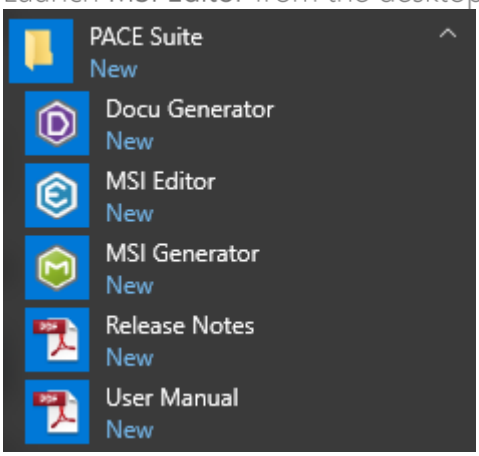


### 2.1.4 Activating PACE Suite

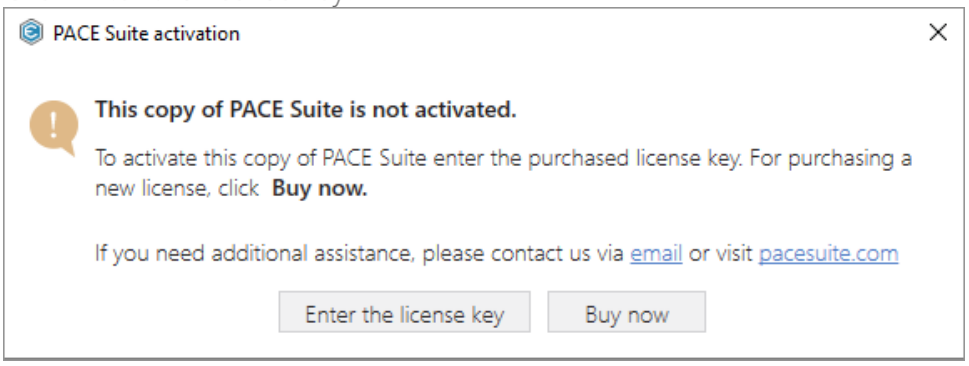
If you have installed PACE Suite with the **Skip entering a product key now** option or your current license is expired, you can activate your copy of PACE Suite with the provided license key.

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.

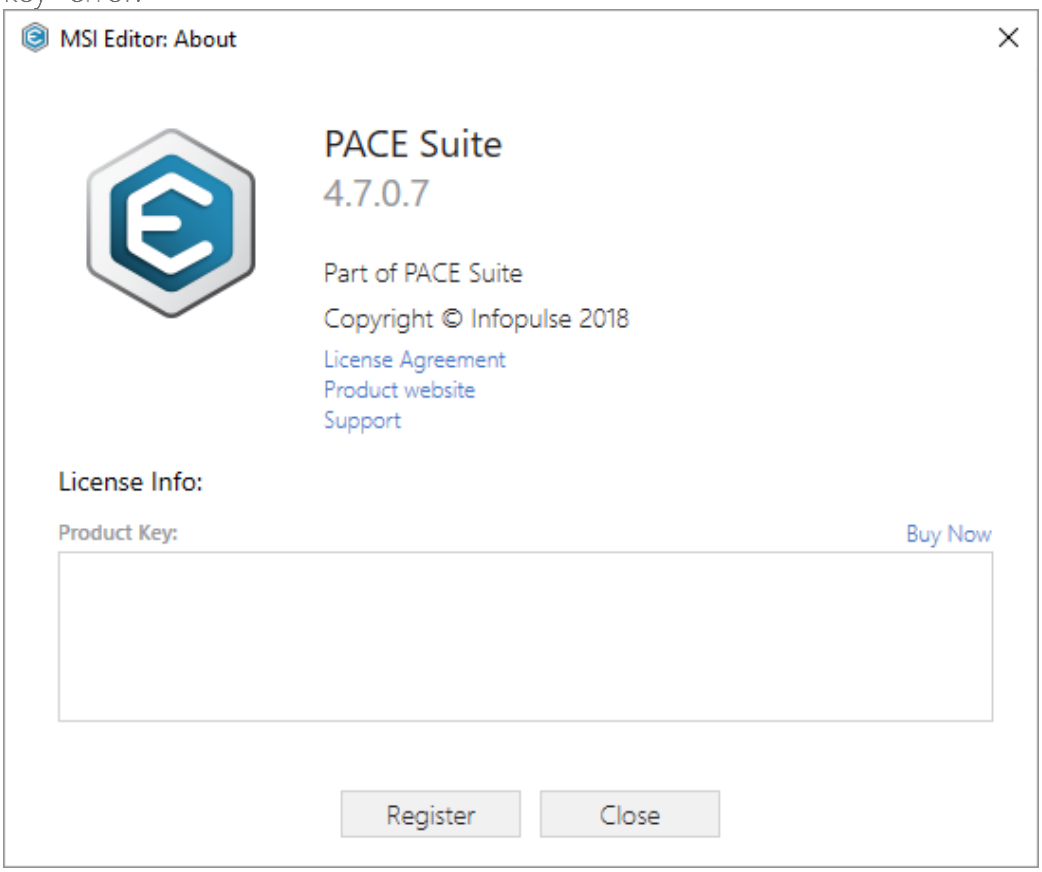



- Click Enter the license key.

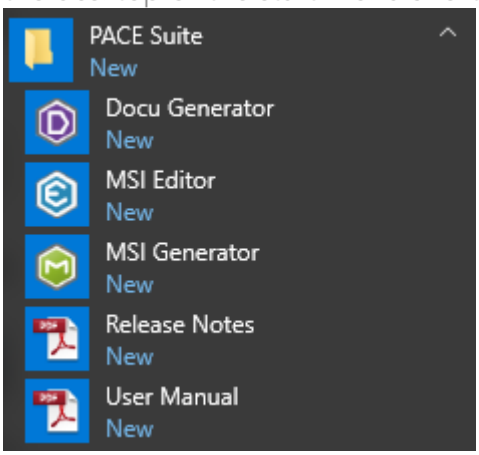



- Enter your license key and click **Register**. Note that if you are using the desktop version of Microsoft Outlook, ensure that the Auto Remove Line Breaks feature is disabled. Otherwise, your keys may be displayed incorrectly, and often that is the cause of obtaining the "Invalid

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

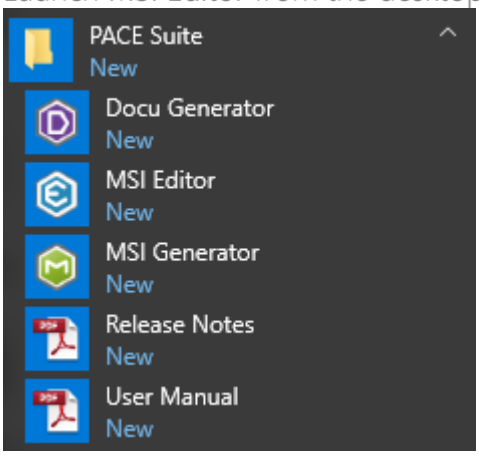
Choose a scenario that better suits your needs:

- Create Blank MSI, described in section 3.1.1
  Create a new empty MSI database in MSI Editor, add folders, files, registry entries and shortcuts directly to the MSI package.
- Generate MSI from Scratch, described in section 3.1.2
  Create a new empty project in MSI Generator, add folders, files, registry entries and shortcuts to this project and then generate the MSI package from this project.
- Convert (Repackage) EXE to MSI, described in section 3.1.3
  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 file system and registry permission changes.
- Capture System Changes to MSI, described in section 3.1.4
  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 file system and registry permission changes.

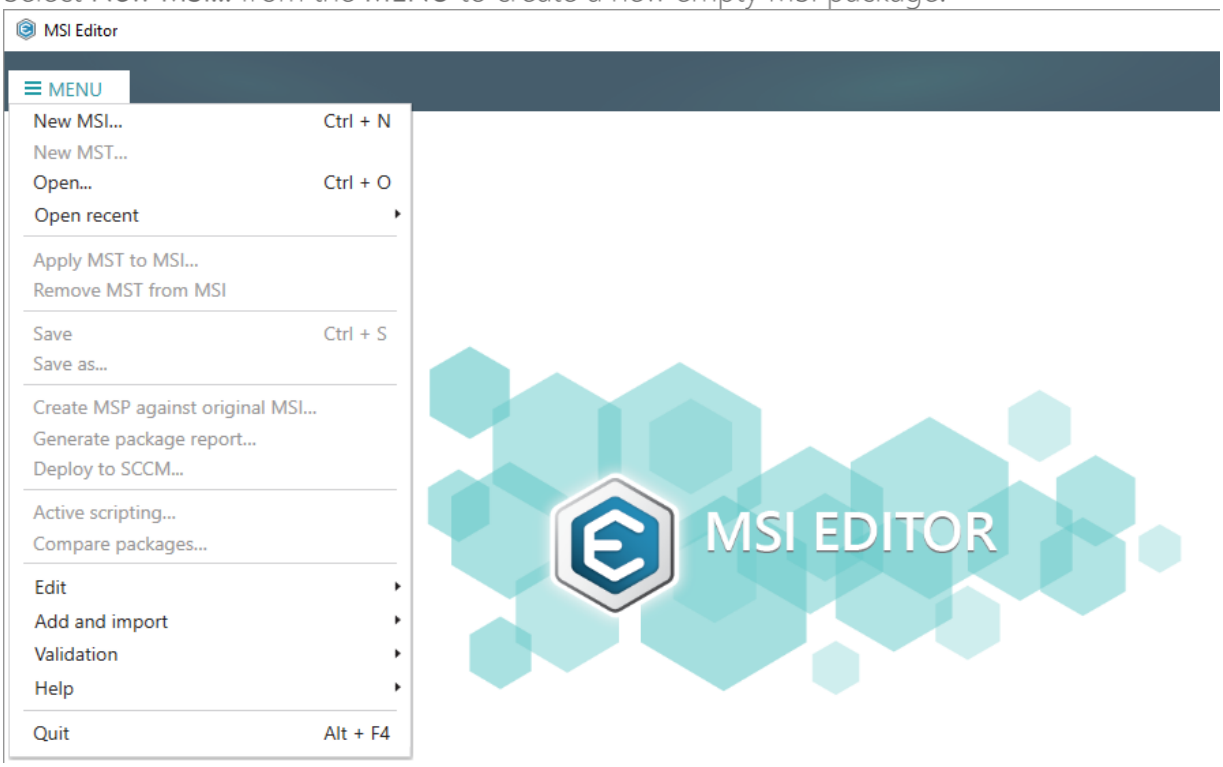
### 3.1.1 Create Blank MSI

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

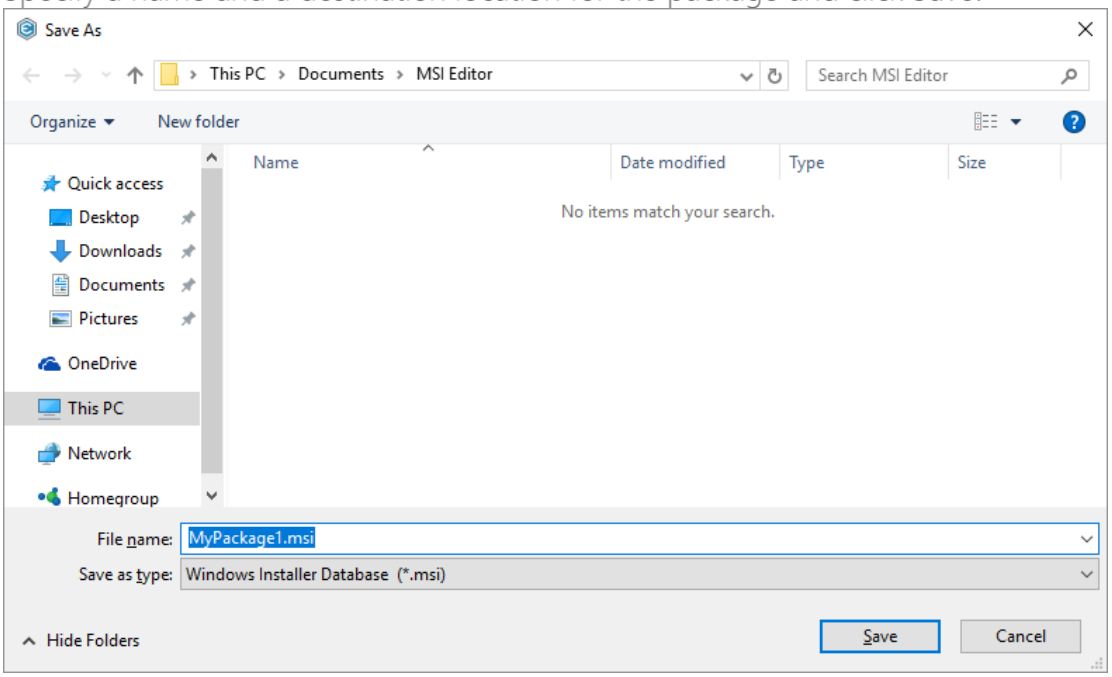
- Launch MSI Editor from the desktop or the start menu shortcut.



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

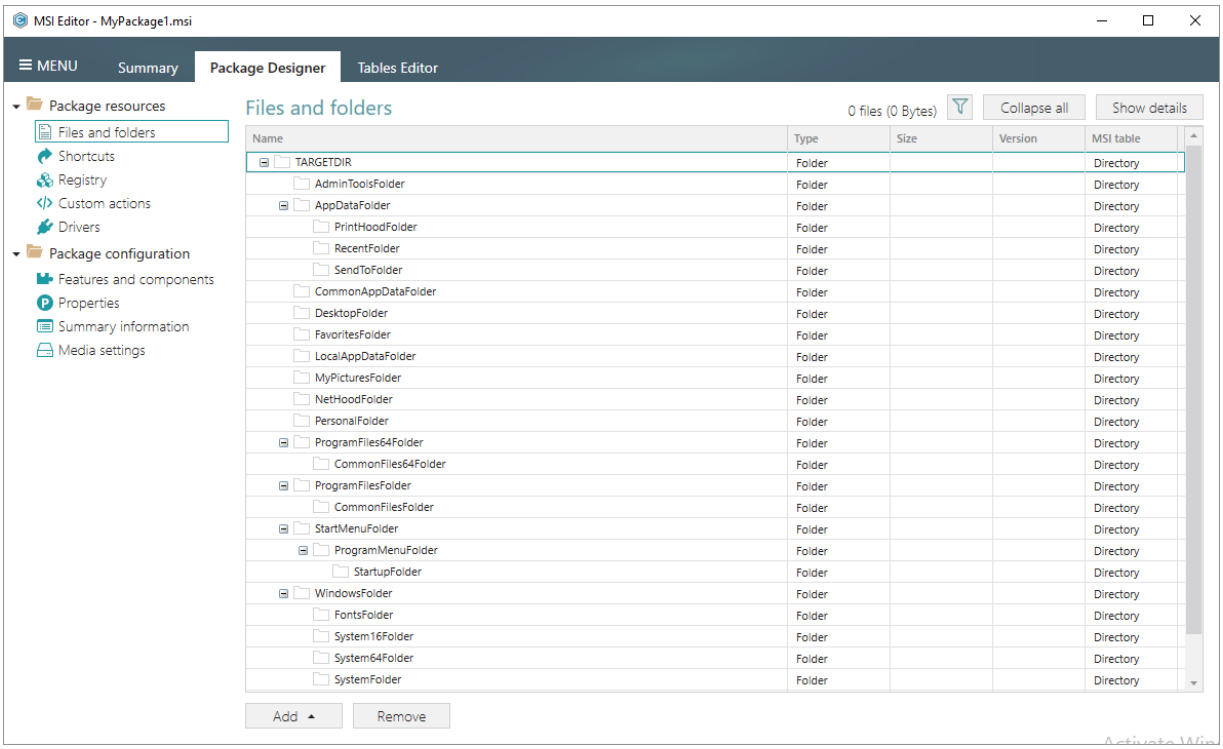



- Specify a name and a destination location for the package and click Save.

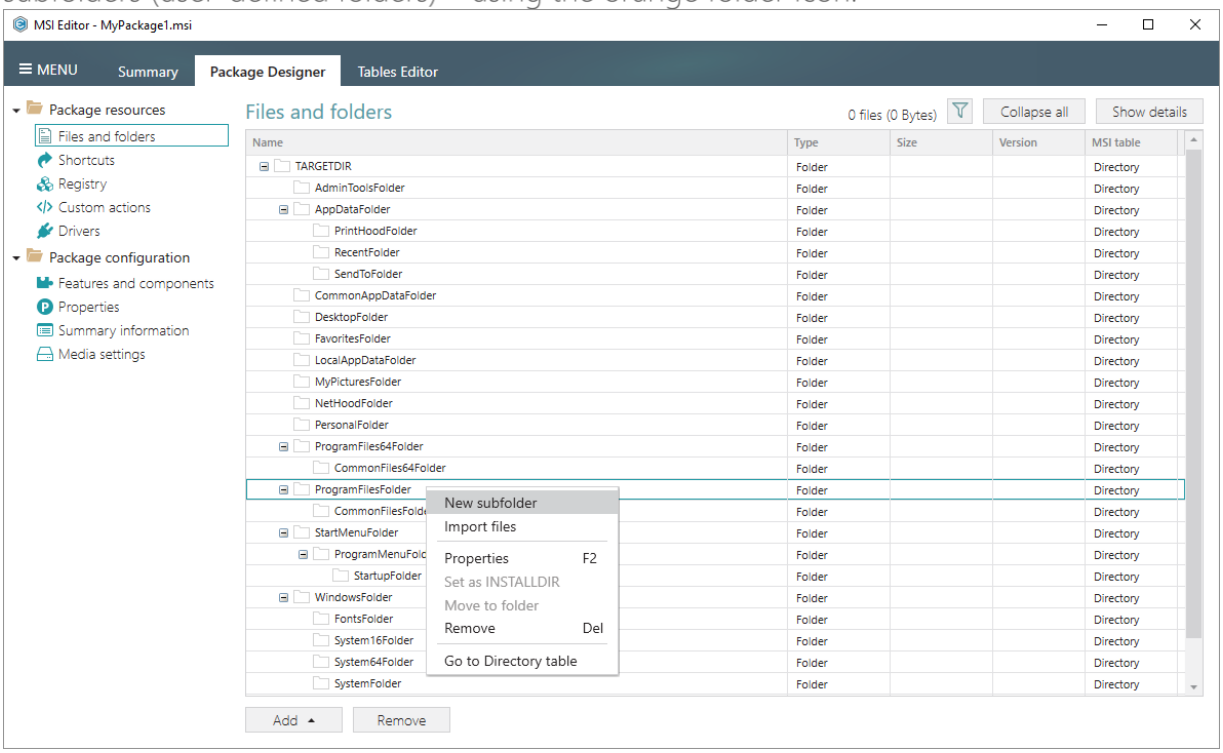



- Once the newly created package is opened automatically, you can start adding the necessary resources using MSI Editor. The steps from [5] to [14] describe how to create a new folder and import files to this folder; the steps from [15] to [23] describe how to create a new registry key, value, and import registry entries from the REG files; the steps from [24] to [32] describe how to create a new shortcut and import exsting ones from the file system; and the last two steps [33] and [34] describe how to update the application details and save the MSI package.

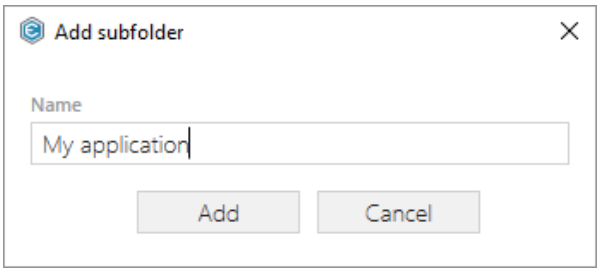
- In order to create a new folder and import files to this folder, go to the **Package Designer** –> **Files and folders** tab.



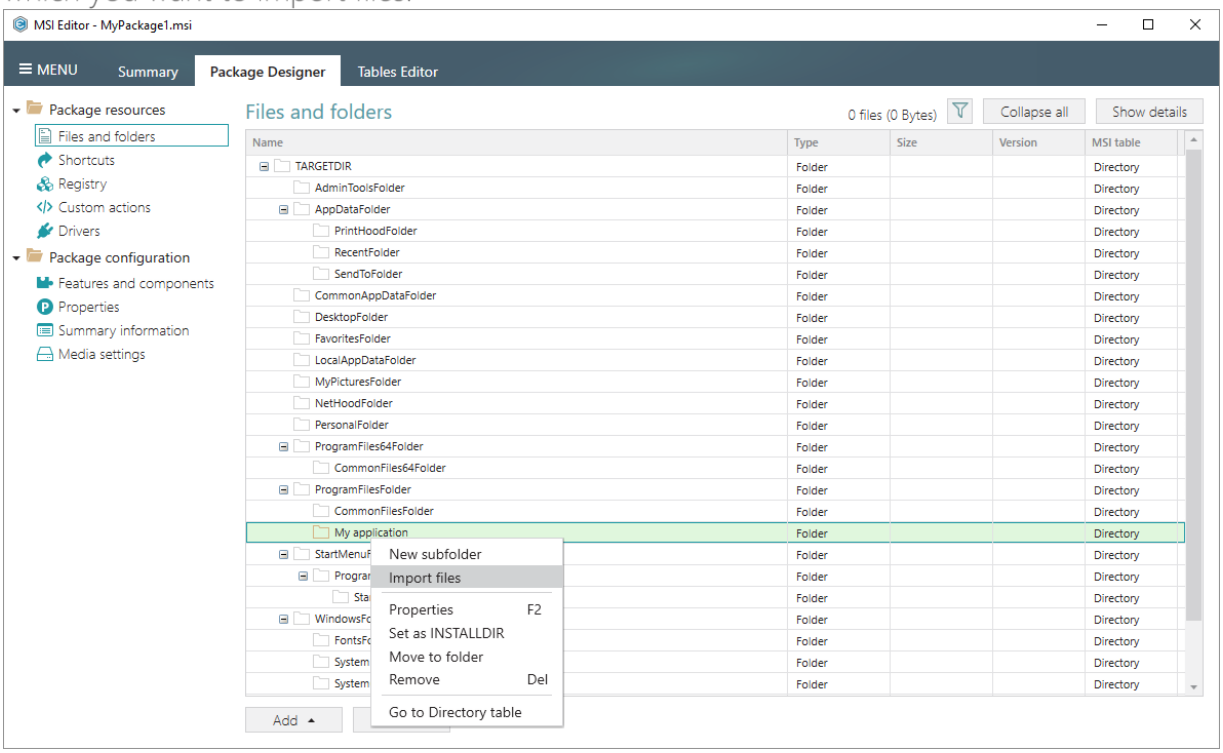

- Select **New subfolder** from the context menu of a folder, where you want to create a new subfolder. Note that system folders are displayed using the grey folder icon and the subfolders (user-defined folders) – using the orange folder icon.



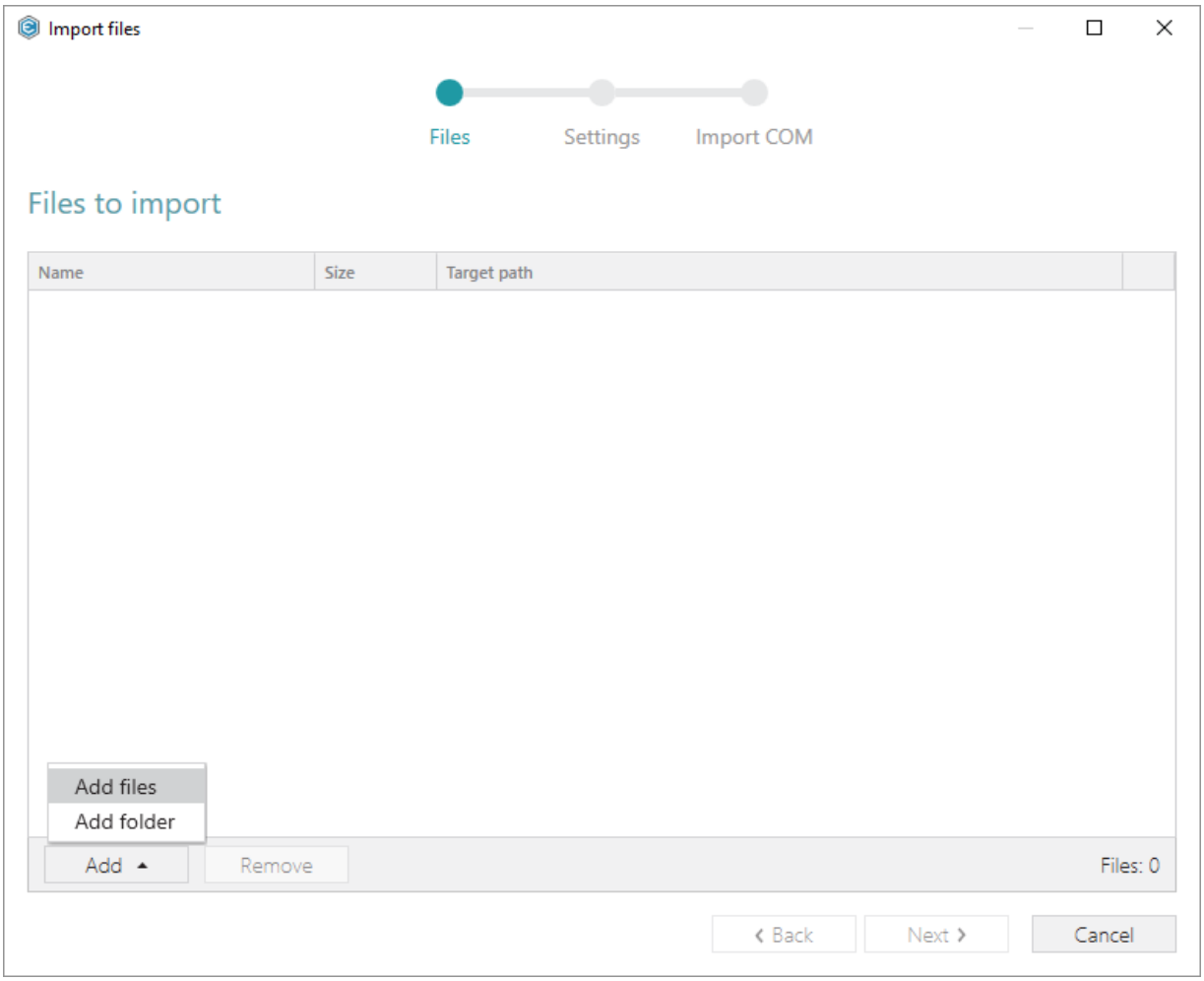
- Enter a folder name and click **Add**.



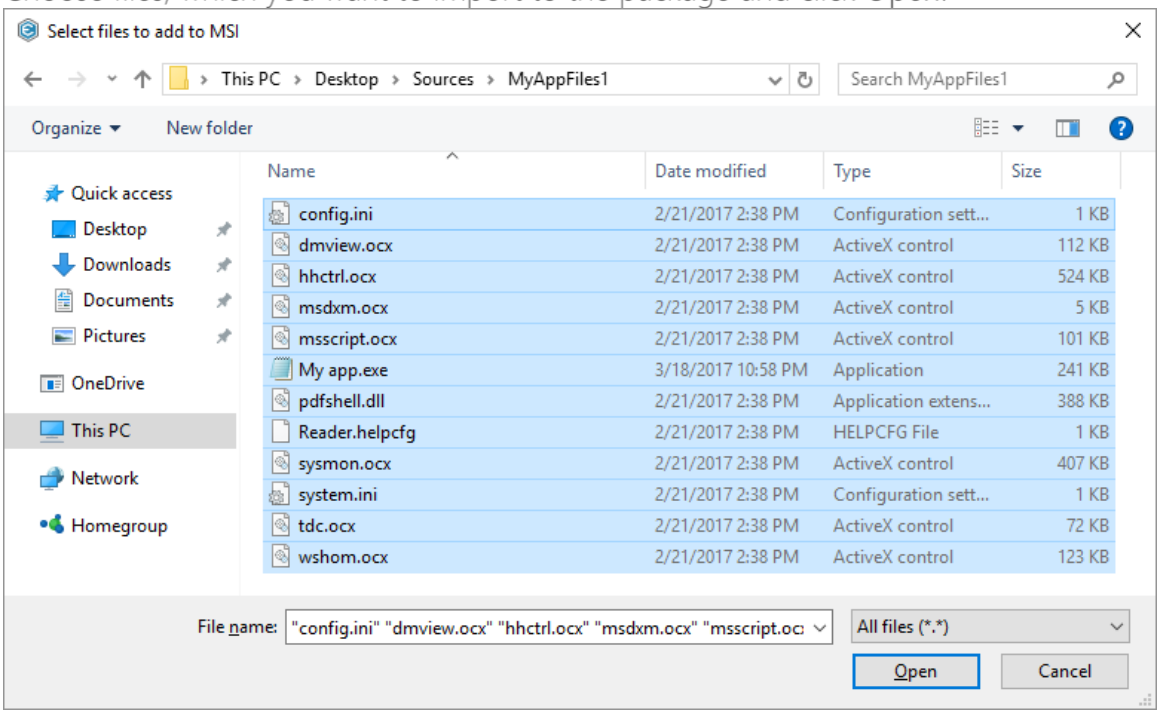

- For adding files to your package, select **Import files** from the context menu of a folder, to which you want to import files.



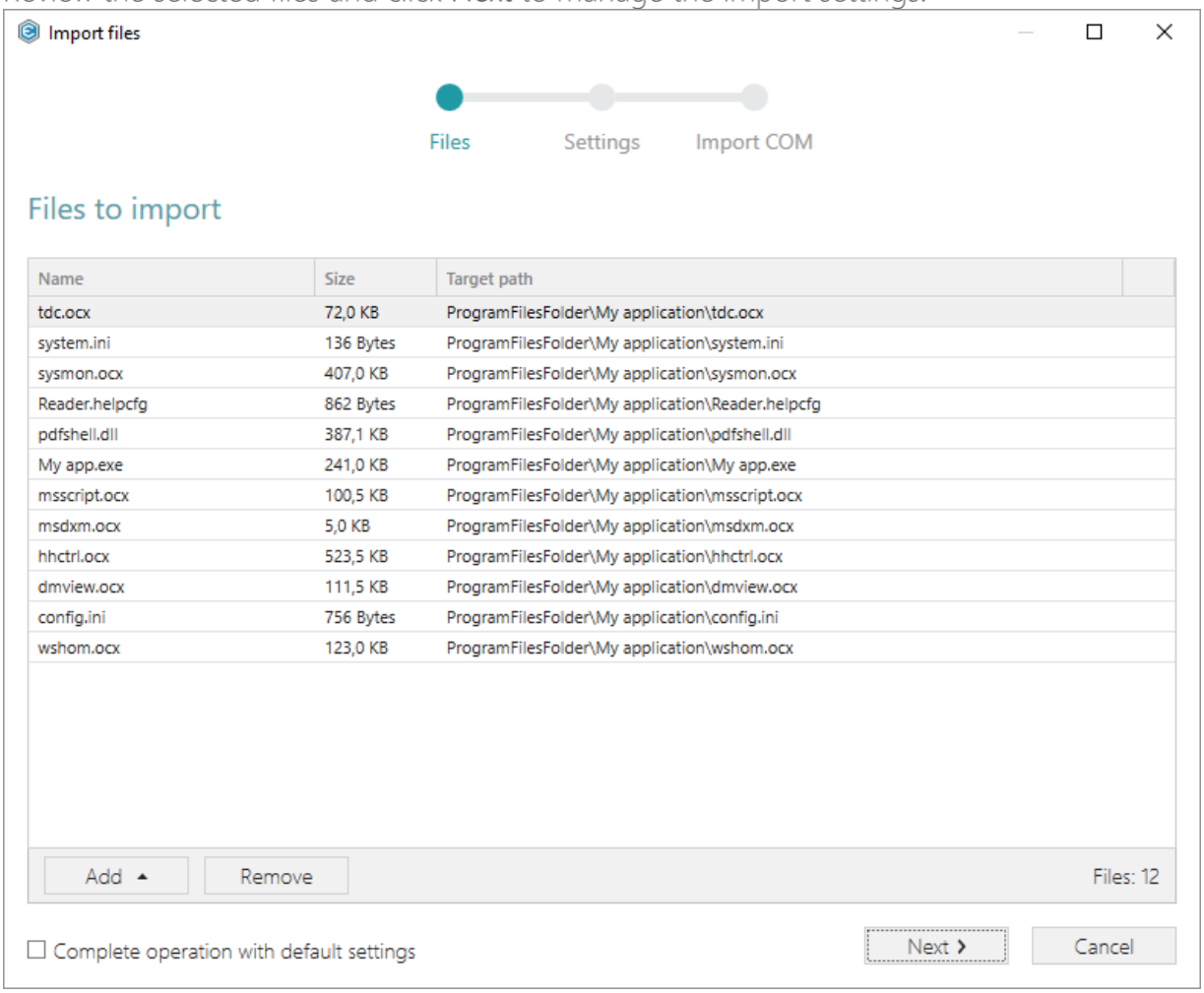
- Click **Add** -> **Add files** to choose files in a folder or **Add** -> **Add folder** to choose a folder with subfolders and files.



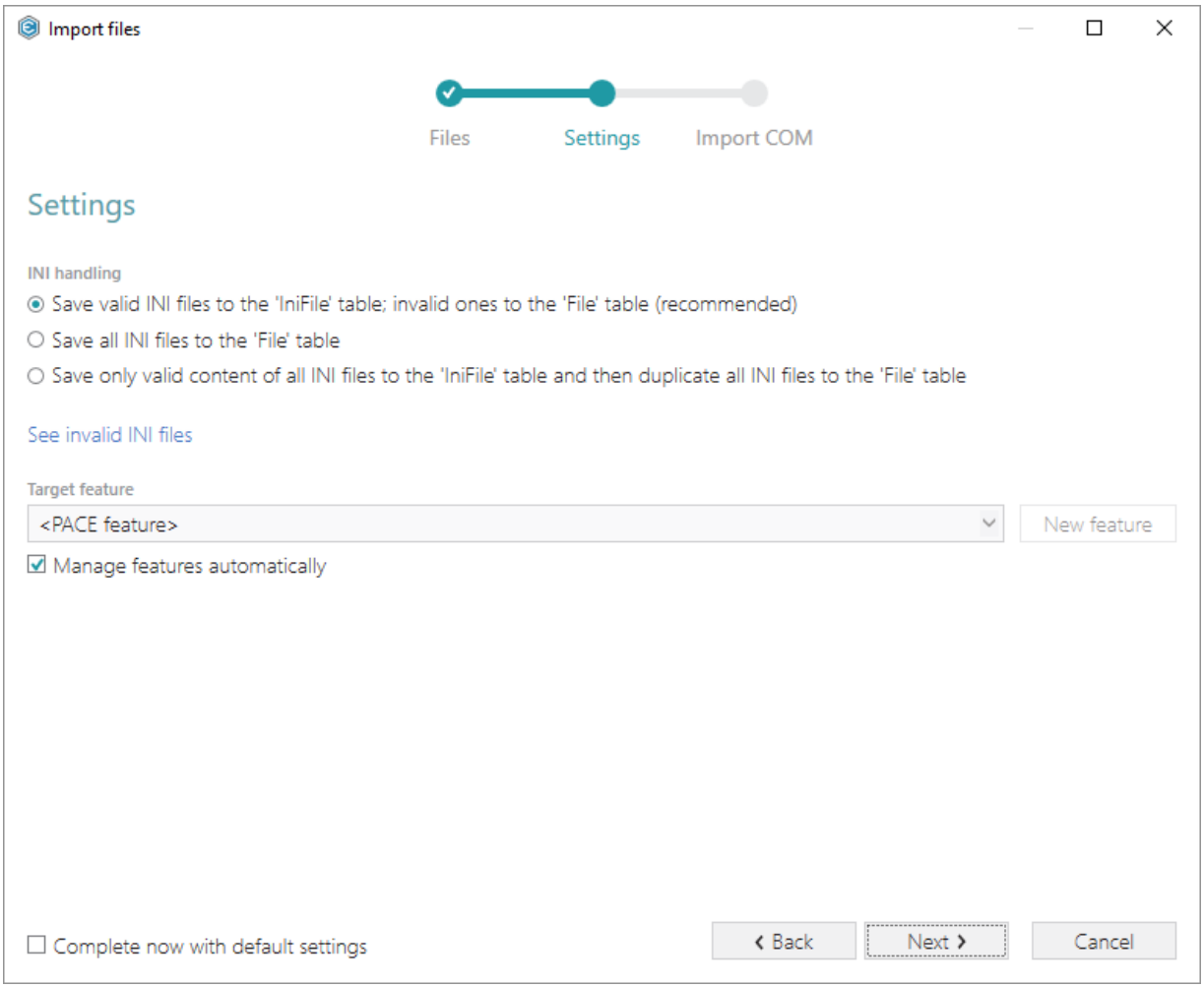

- Choose files, which you want to import to the package and click **Open**.



- Review the selected files and click Next to manage the import settings.




- At this step, you can manage how to save INI files, select an MSI Feature, to which components with files will be assigned. Below, in the tables, you will find the detailed description of these settings. Note that files compression types and other Media settings are represented on the separate Media settings tab of the Package Designer.
  Choose the desired settings and click Next to continue.




## INI handling

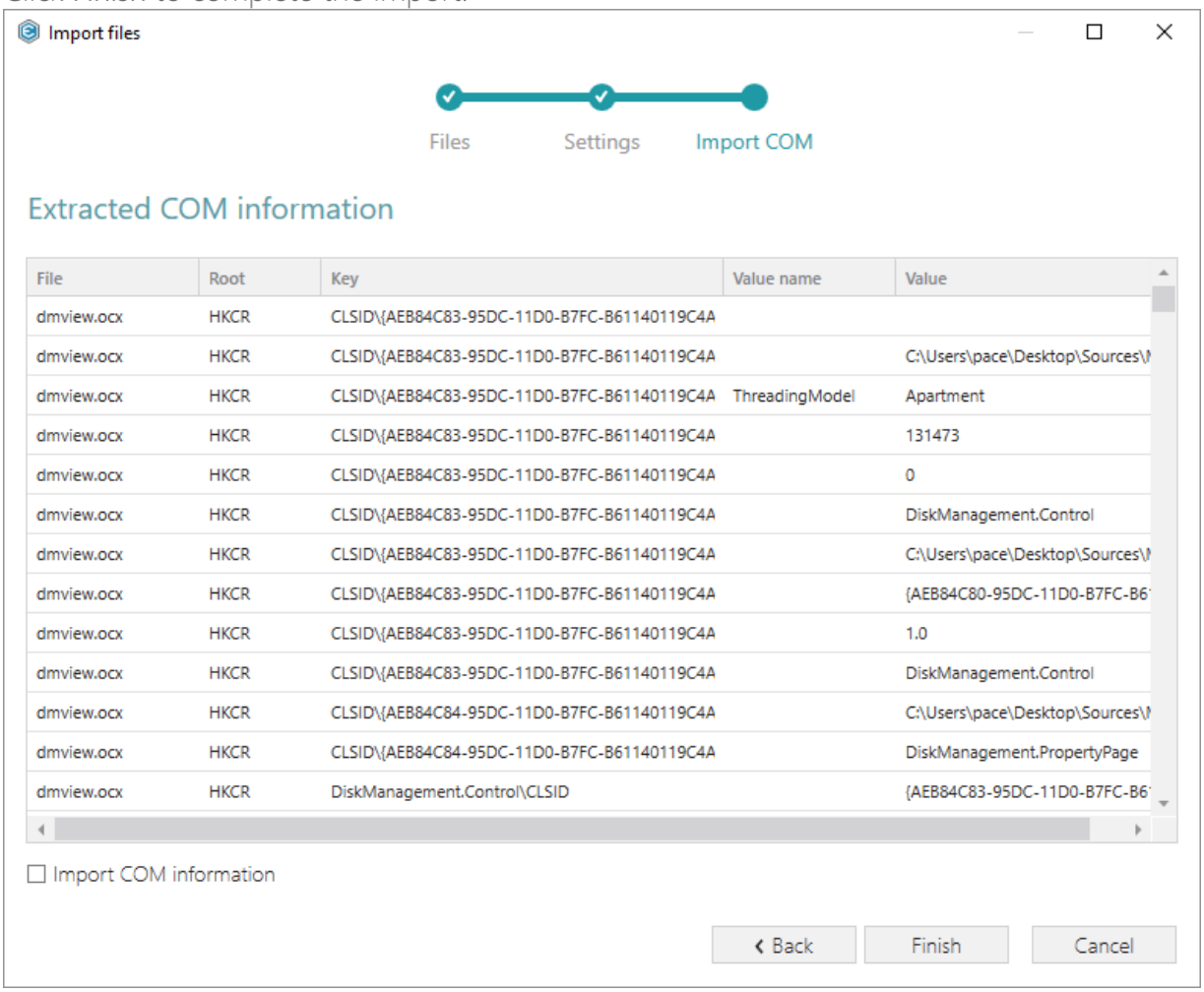
| | |
|---|---|
| Save valid INI files to the 'IniFile' table; invalid ones to the 'File' table (recommended) | INI files, which comply with the INI file format, will be saved to the 'IniFile' table.<br>All other INI files, which contain unsupported data, will be saved to the 'File' table as binary files. |
| 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.<br>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. |

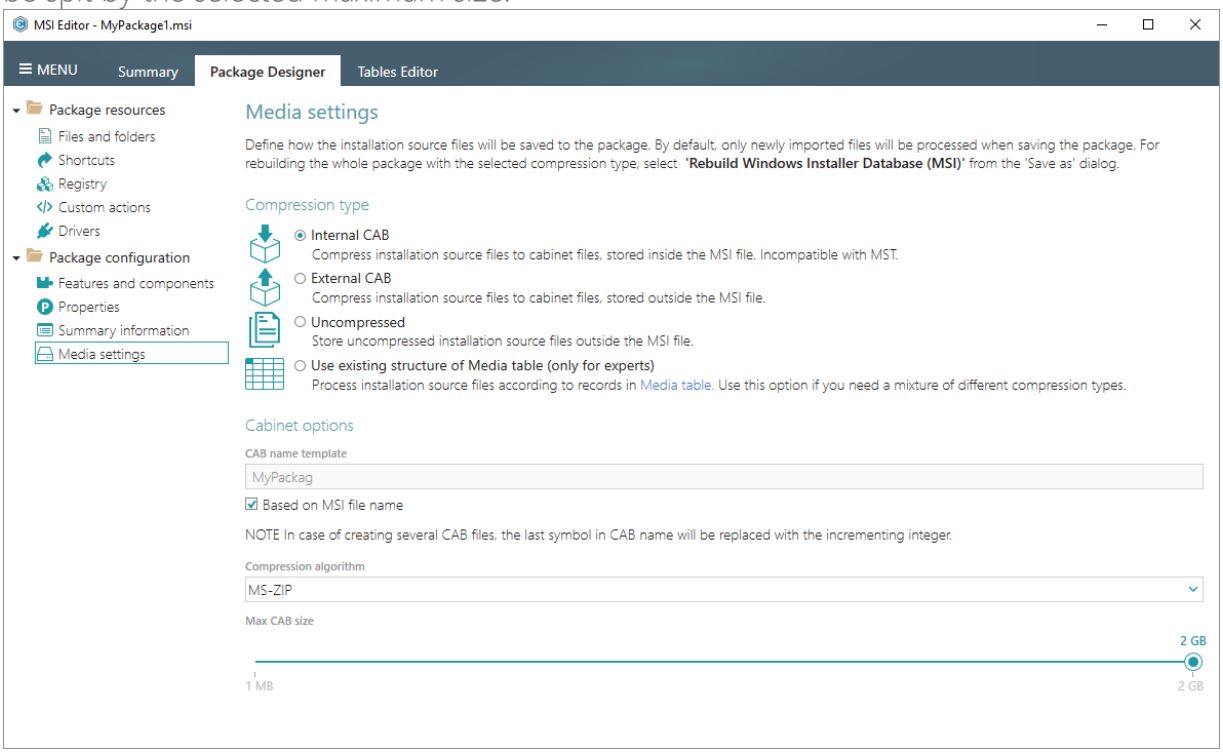
| | |
|---|---|
| <a feature, selected from the list> | Components with files will be assigned to a feature, selected from the list. |

- On the last step you can review COM information, extracted from the selected files. For importing this COM information along with files, select the Import COM information option. Click Finish to complete the import.

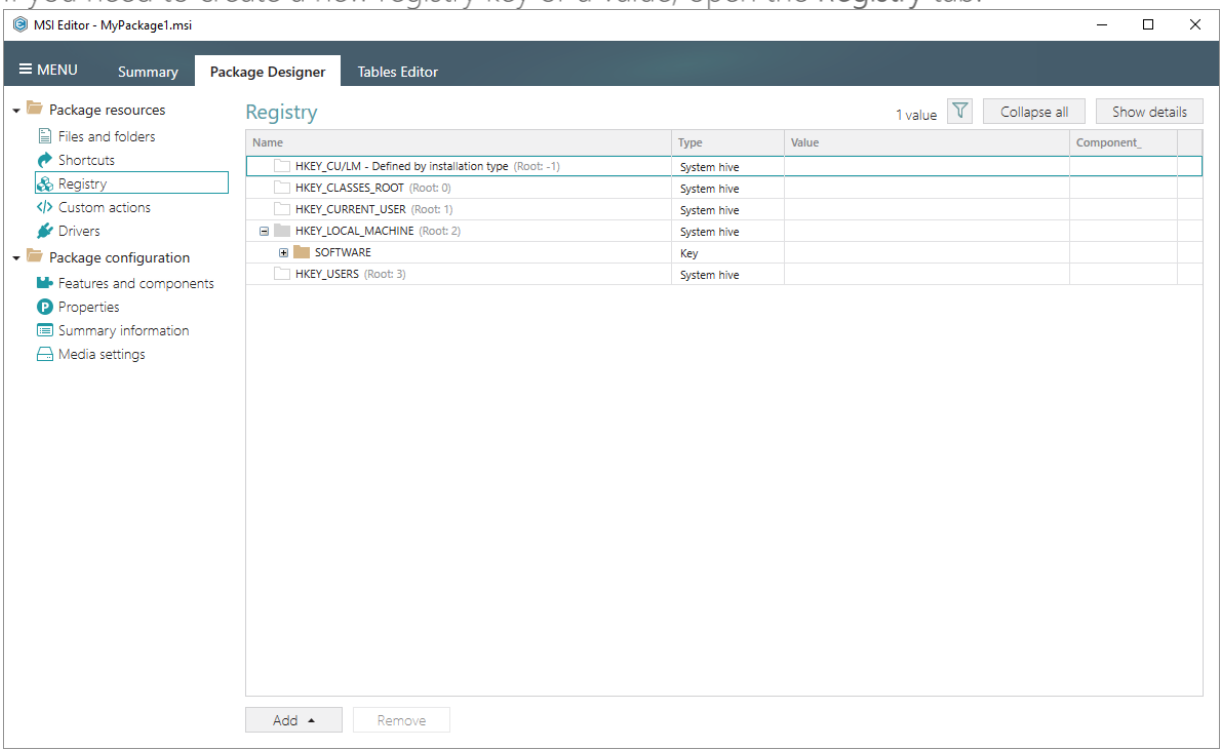



- Go to the Media settings tab of the Package Designer and choose the necessary compression type for the imported files. For the compressed files you can manage cabinet options such as CAB name, compression algorithm and maximum CAB size. Cabinets will

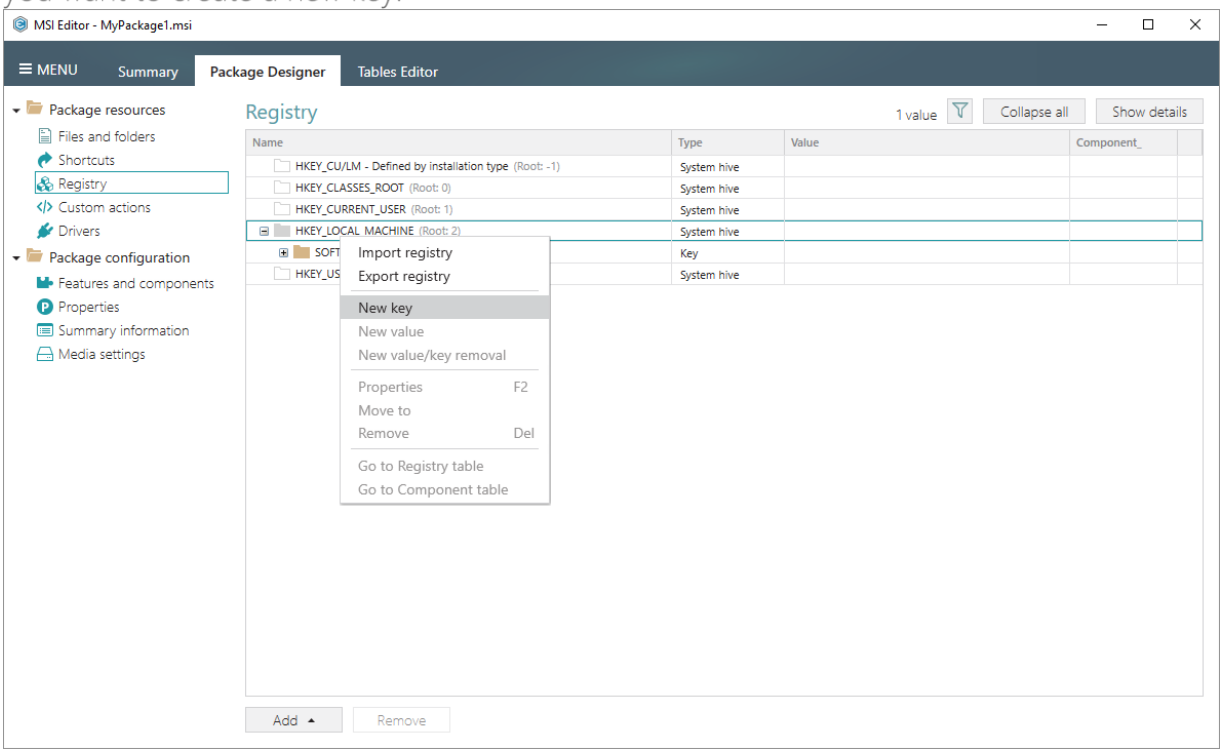
be spit by the selected maximum size.



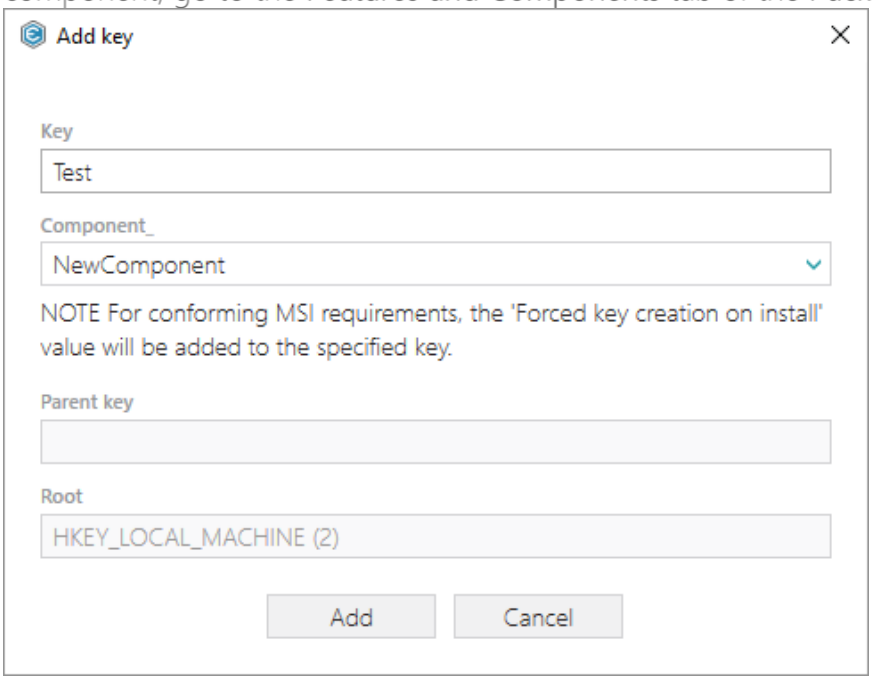

- If you need to create a new registry key or a value, open the **Registry** tab.



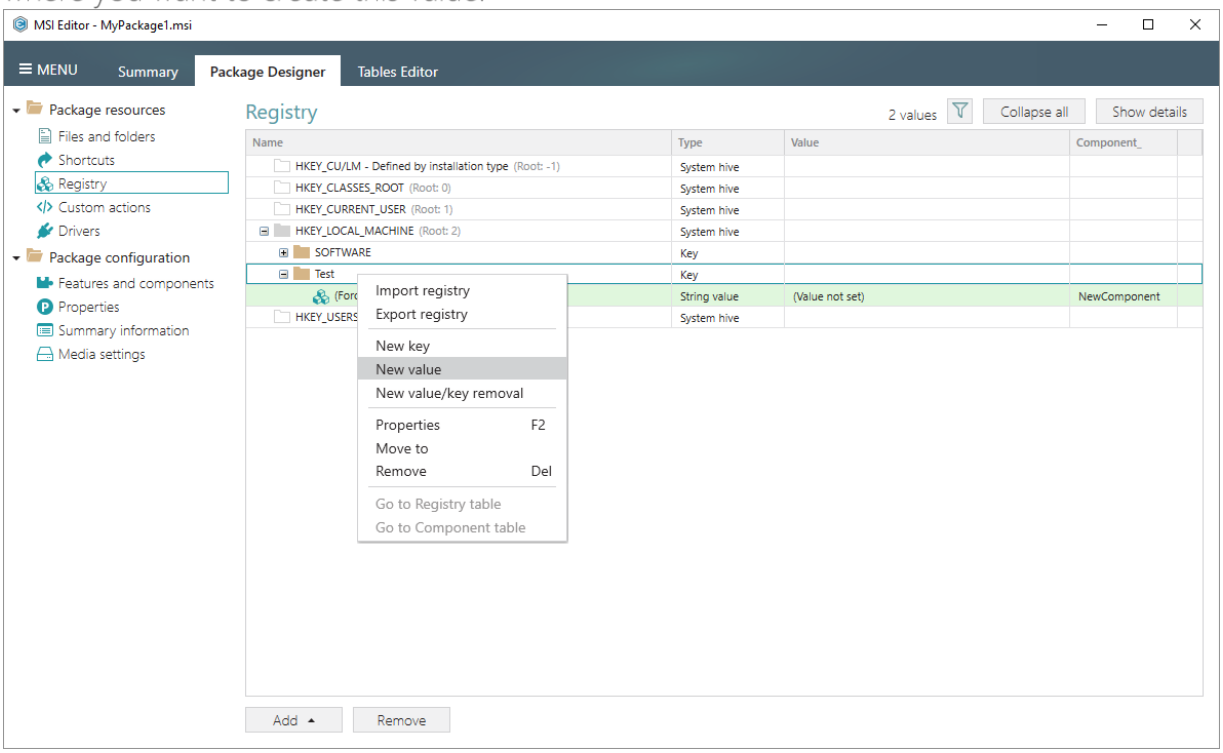
- Select **New key** from the context menu of a root hive (e.g. HKEY_LOCAL_MACHINE), where you want to create a new key.




- 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**.



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

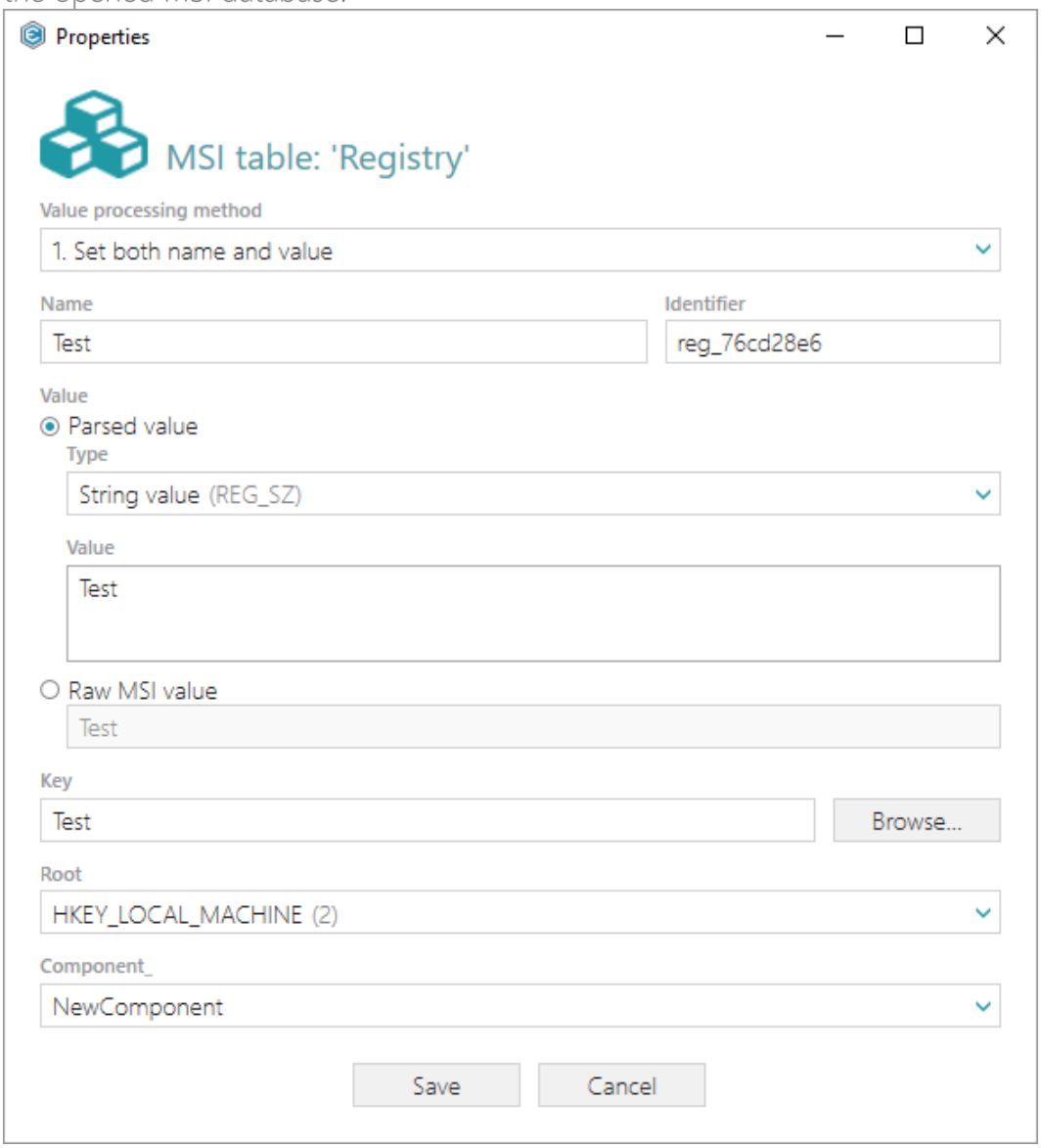



- For creating an ordinary registry value, specify a name and its value, choose a component from the list, to which this value will be assigned, and click **Save**.

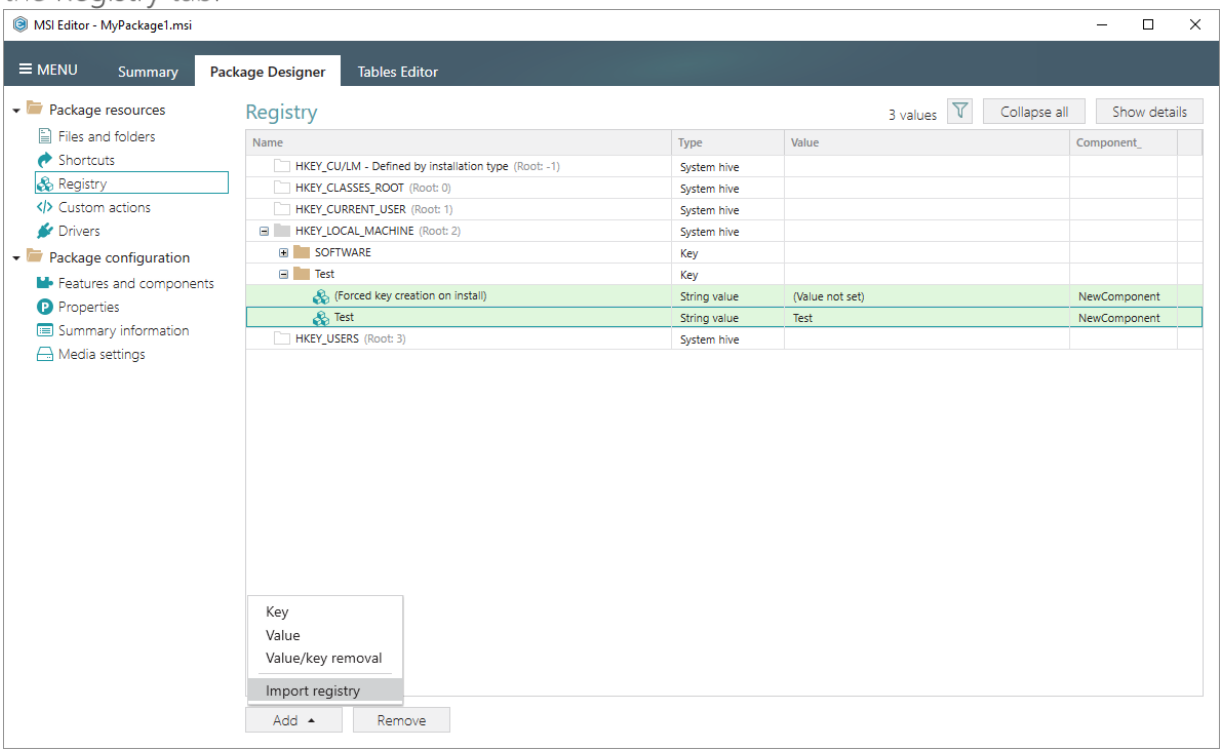
  For creating a Default key value, or securing a key creation and/or removal, select a respective option from the **Value processing method** field.

  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

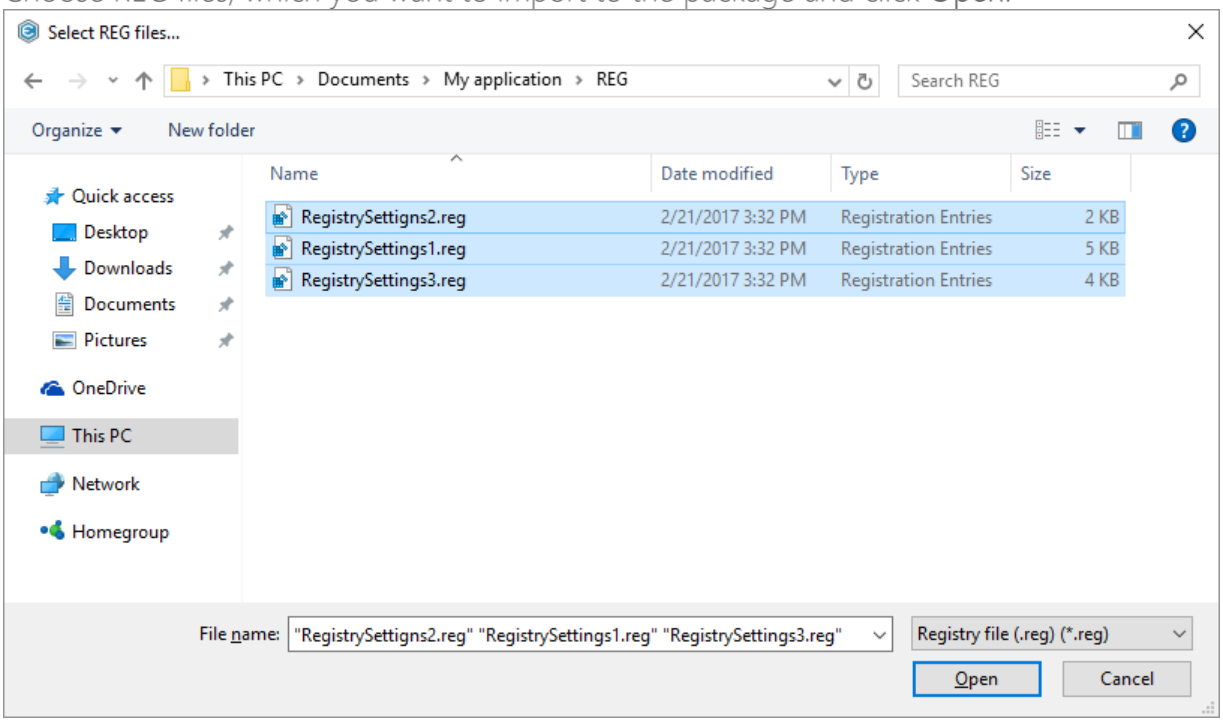
the opened MSI database.



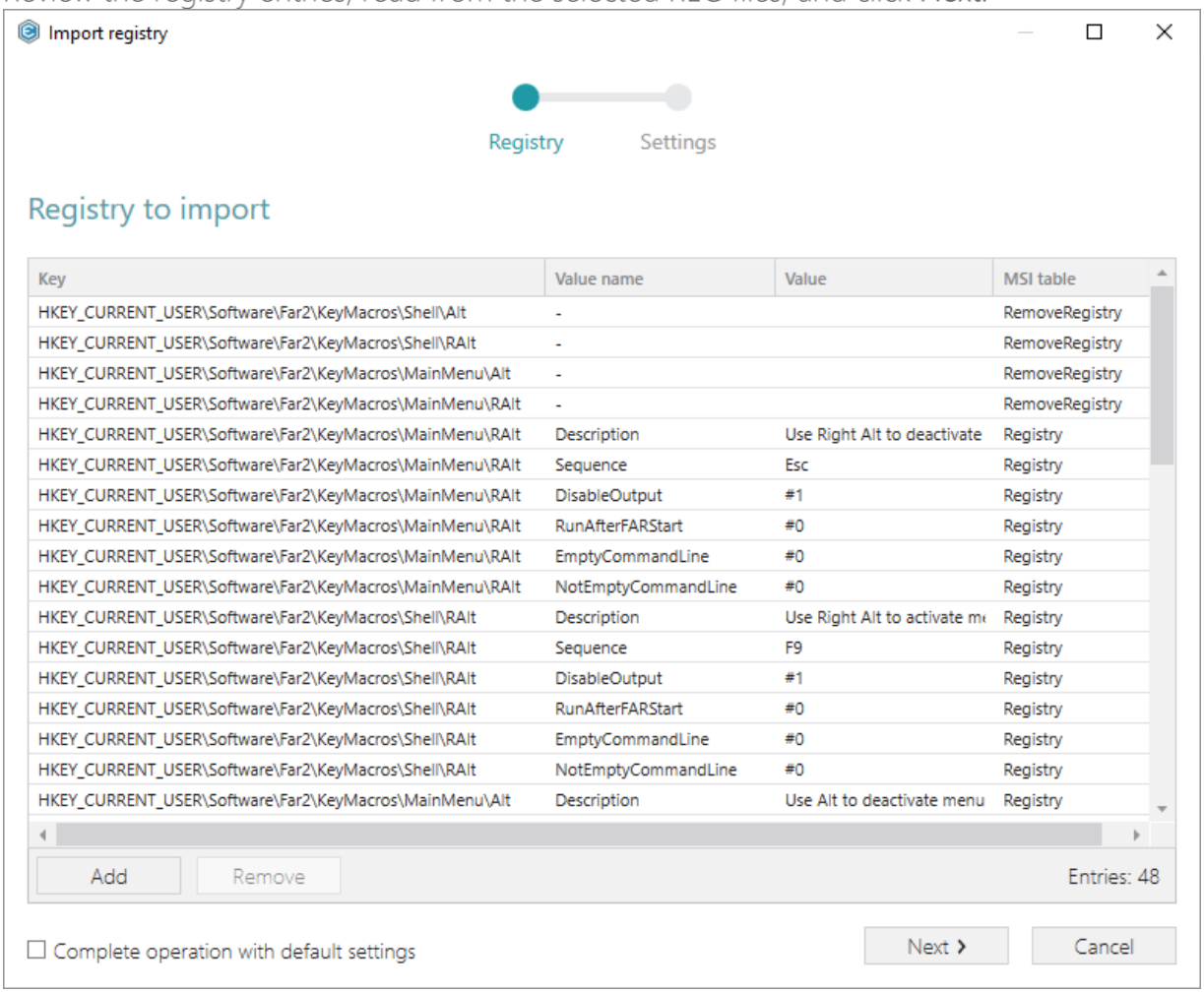
- For importing registry from the REG file, select **Add** -> **Import registry** from the toolbox of the **Registry** tab.



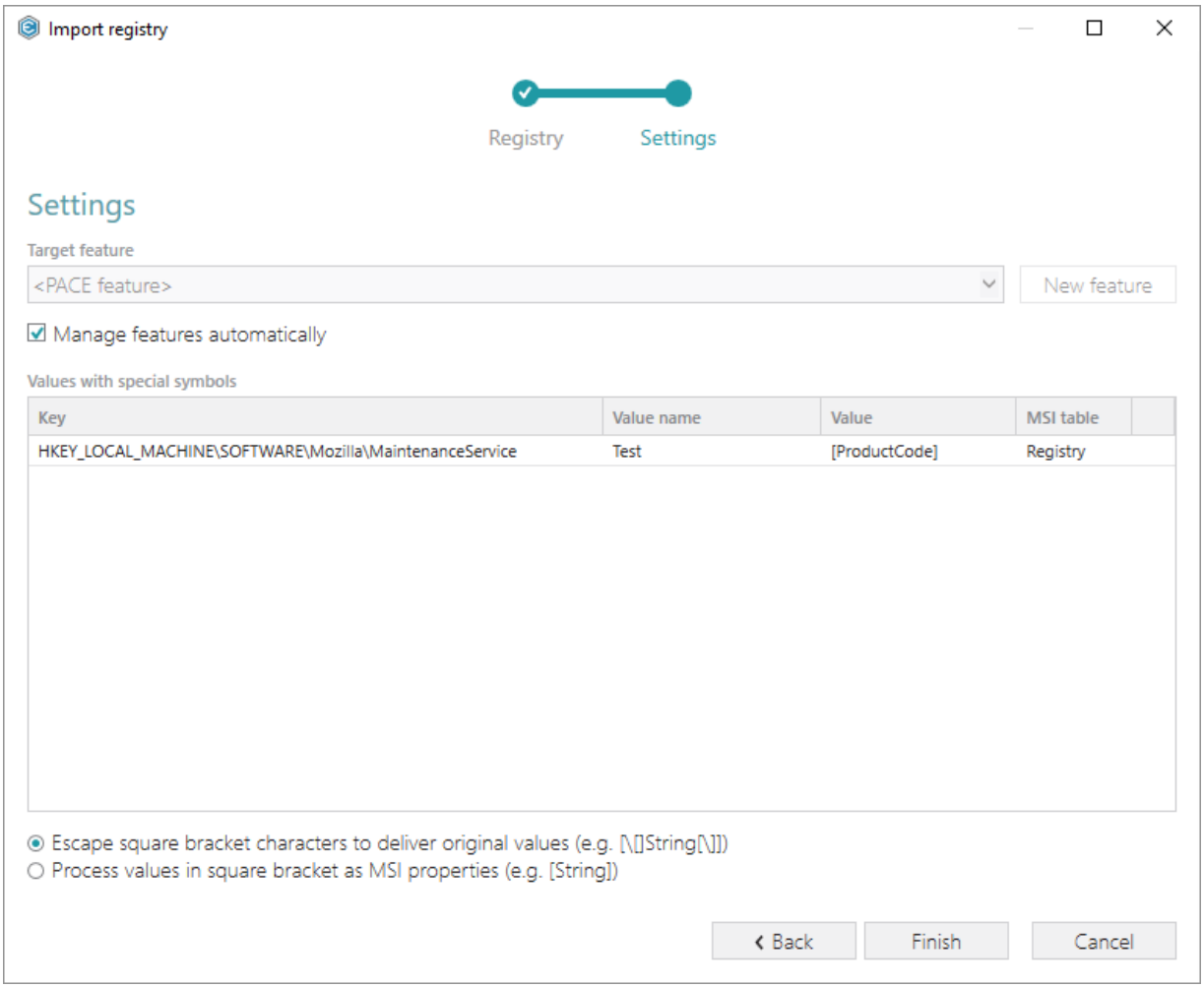

- Choose REG files, which you want to import to the package and click **Open**.



- Review the registry entries, read from the selected REG files, 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.



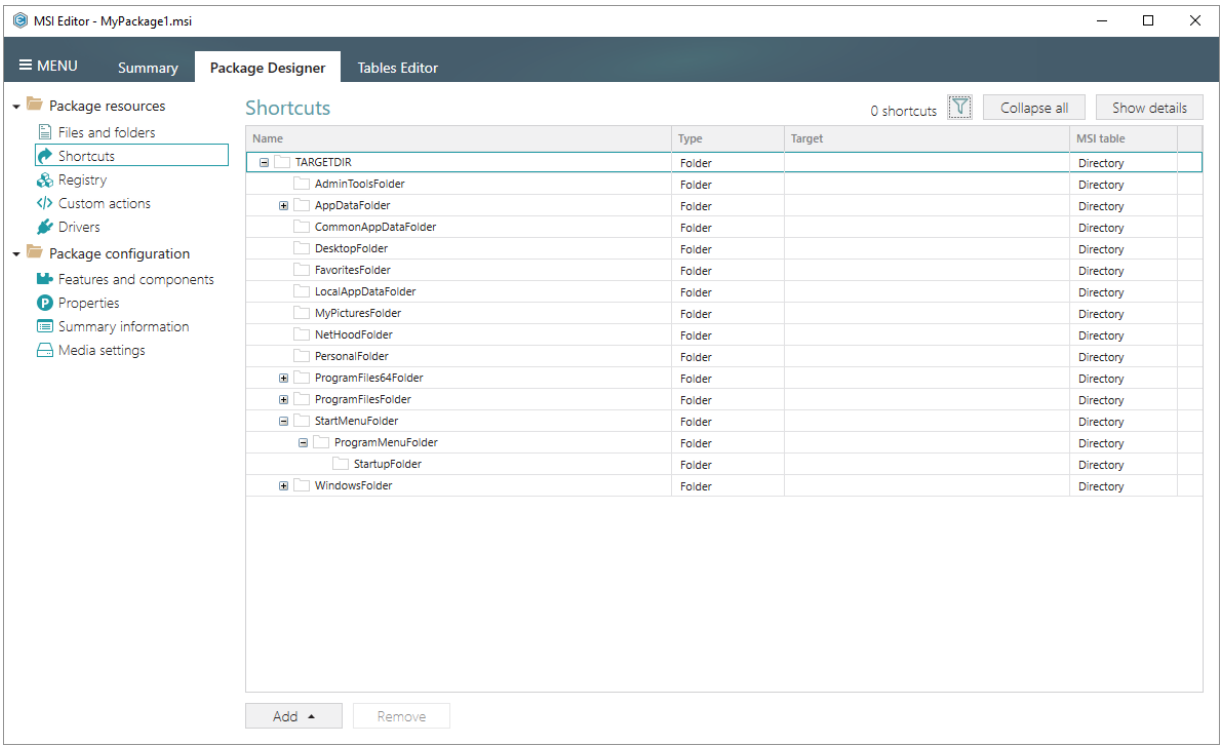

## Feature settings

| | |
|---|---|
| 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. |

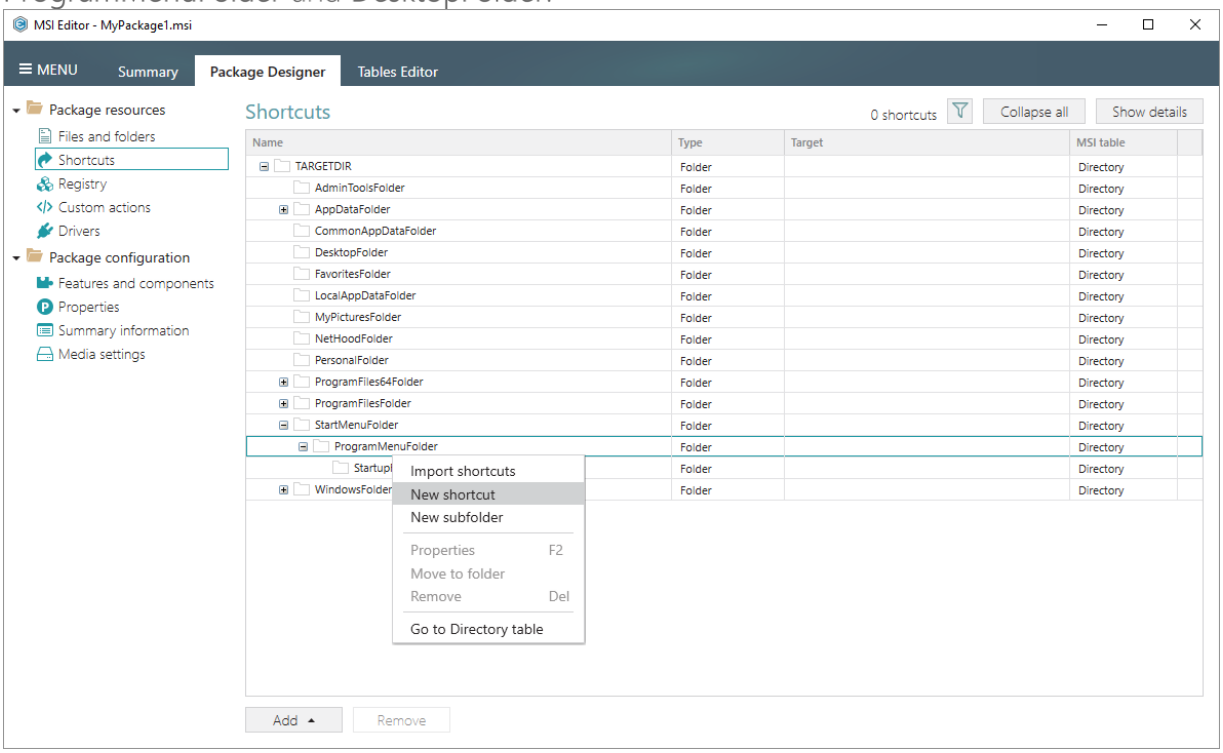
## Values with special symbols

| | |
|---|---|
| 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. |

- If you need to create a new or import the existing shortcuts to the package, go to the Shortcuts tab.

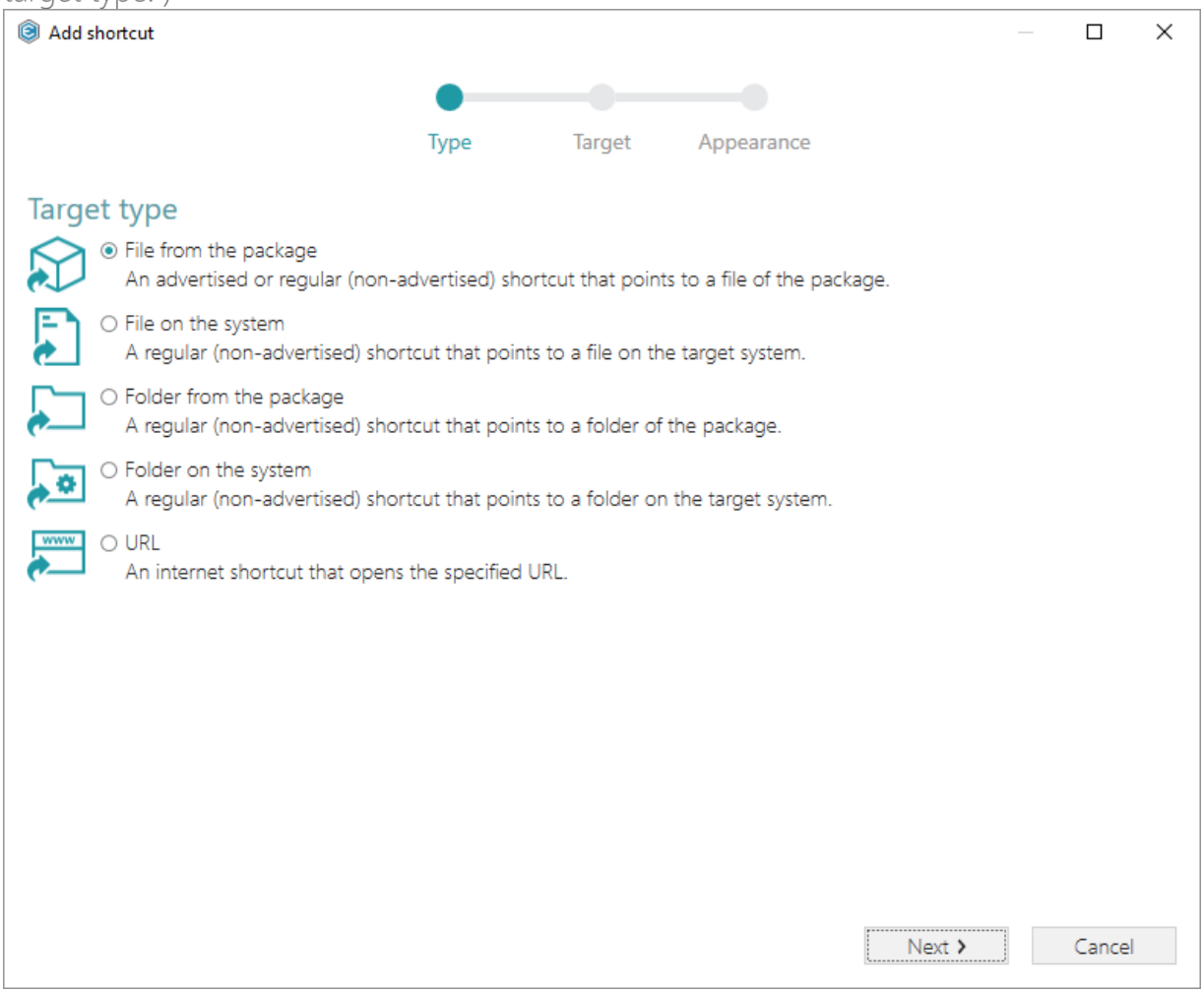



- In order to add a new shortcut, select New shortcut from the context menu of a folder, where you want to create this shortcut. Usually, application shortcuts are installed to the ProgramMenuFolder and DesktopFolder.

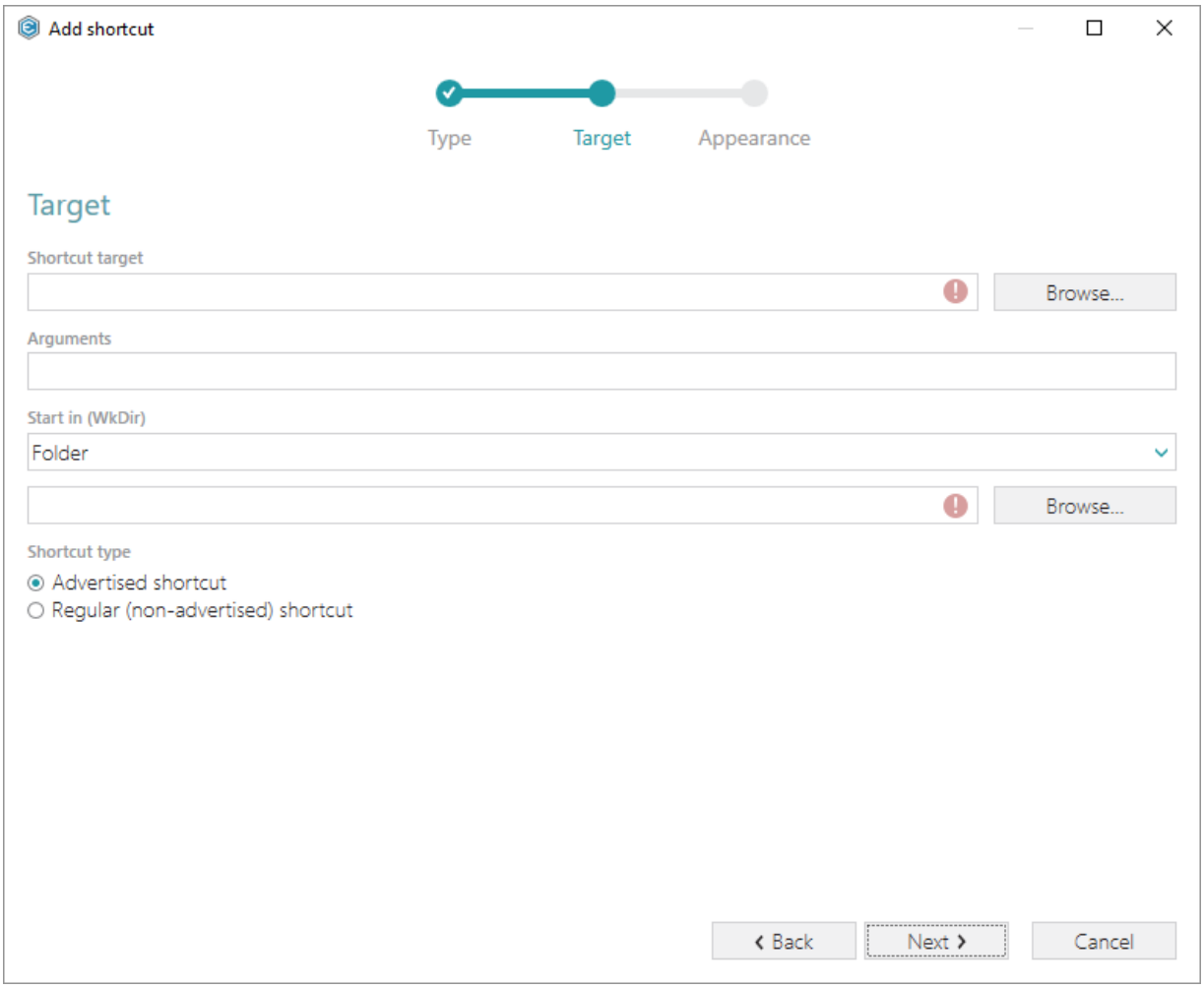



- Select a type of the shortcut target and click Next.
(The following instructions describe creation of the shortcut of the 'File from the package'

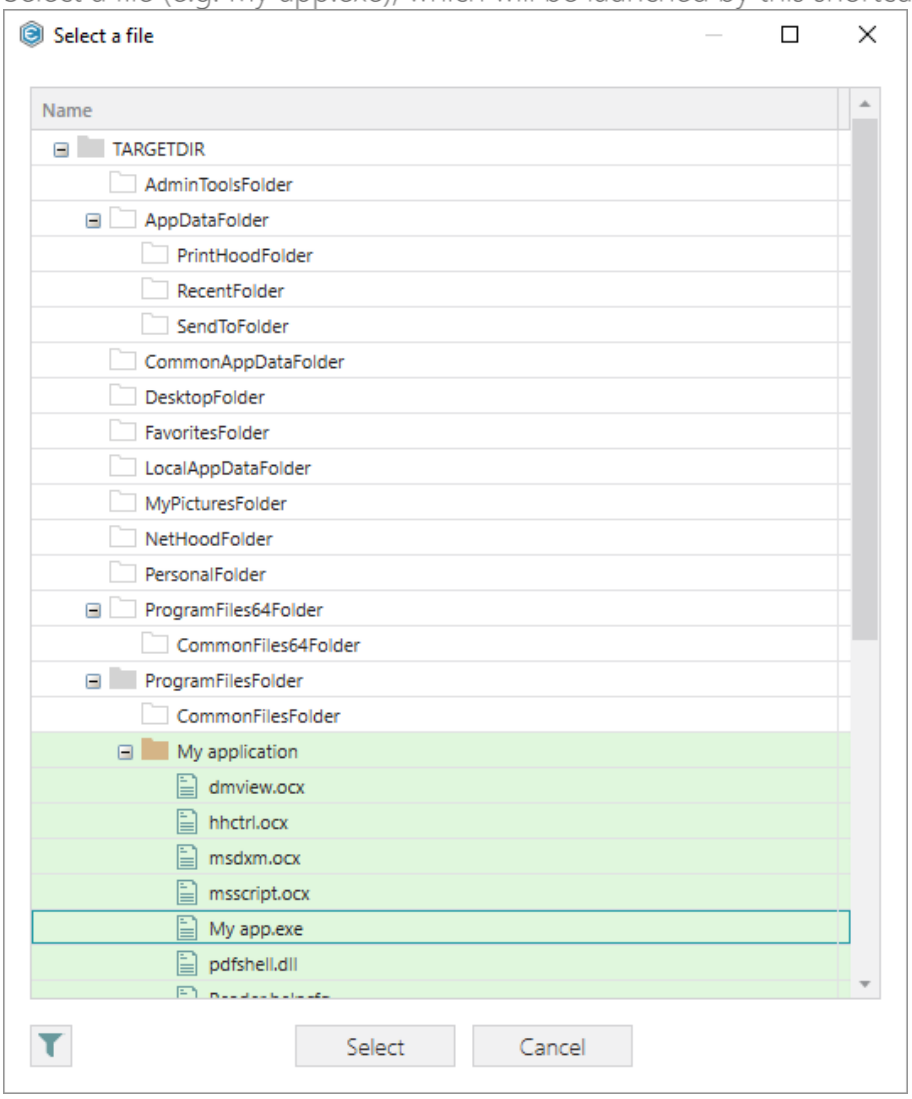
target type. )


Add shortcut
Type
Target
Appearance
Target type
File from the package
An advertised or regular (non-advertised) shortcut that points to a file of the package.
File on the system
A regular (non-advertised) shortcut that points to a file on the target system.
Folder from the package
A regular (non-advertised) shortcut that points to a folder of the package.
Folder on the system
A regular (non-advertised) shortcut that points to a folder on the target system.
URL
An internet shortcut that opens the specified URL.
Next ›
Cancel

- On the second step, click **Browse**, located next to the **Shorcut target** field, to set a target of the shortcut.

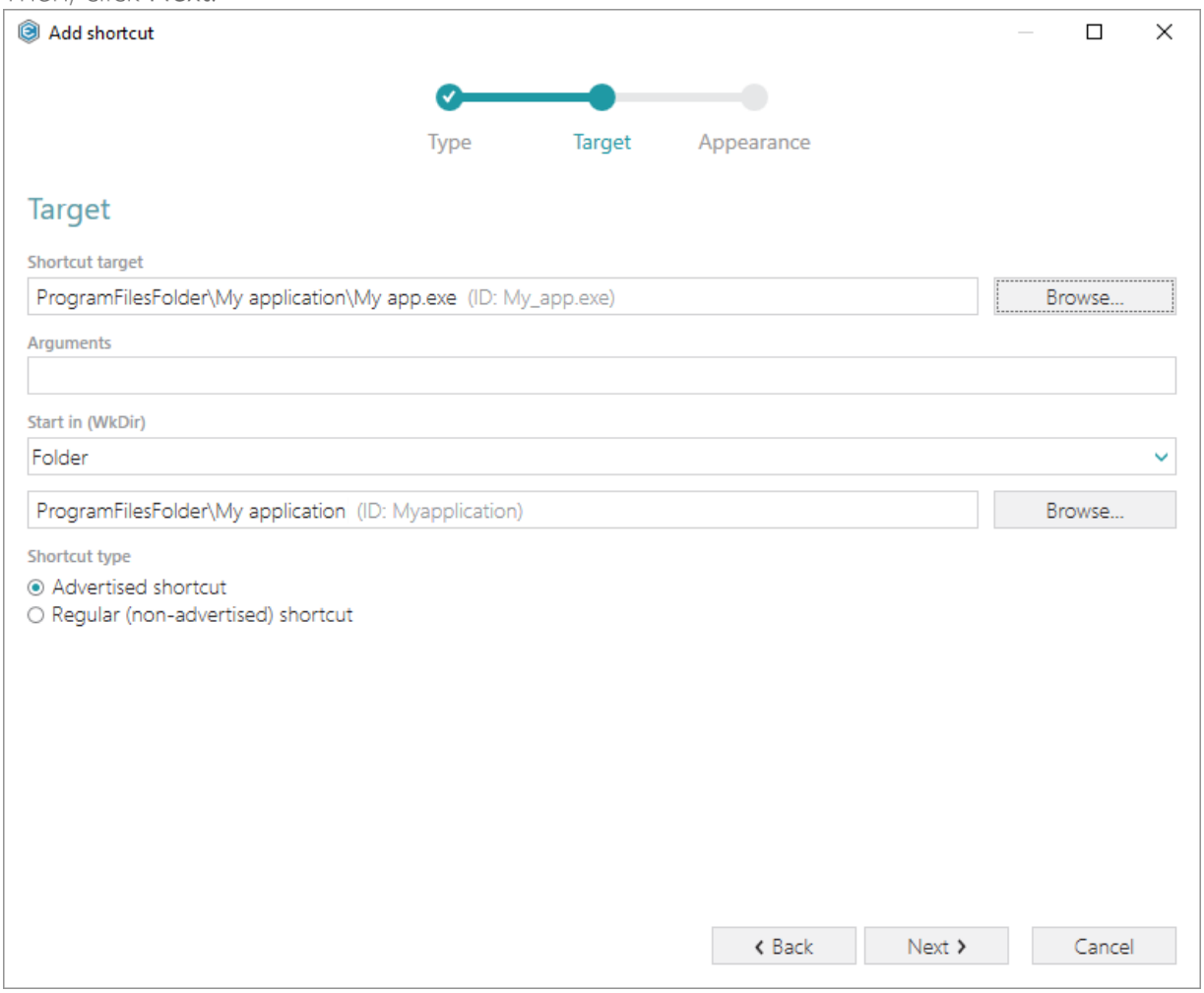


- Select a file (e.g. My app.exe), which will be launched by this shortcut, and click **Select**.

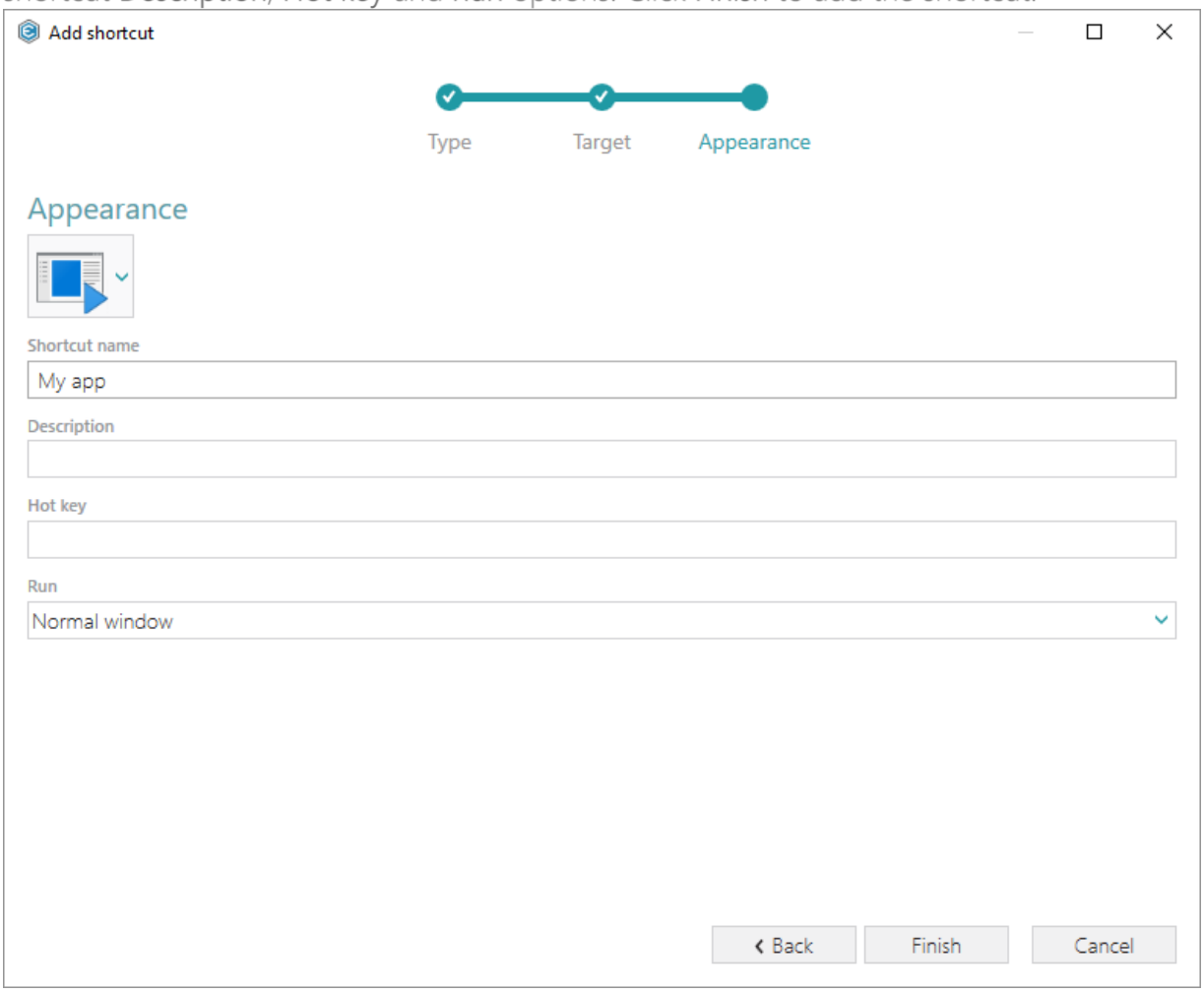



- If needed, enter a shortcut **Arguments**, change the **Start in** path (select a folder or a property), and select a **Shortcut type**. Note that only the Advertised shortcut supports advertisement and repair. This means that if the executable to which the shortcut is pointing is not there then windows installer will repair the application and replace the missing files.

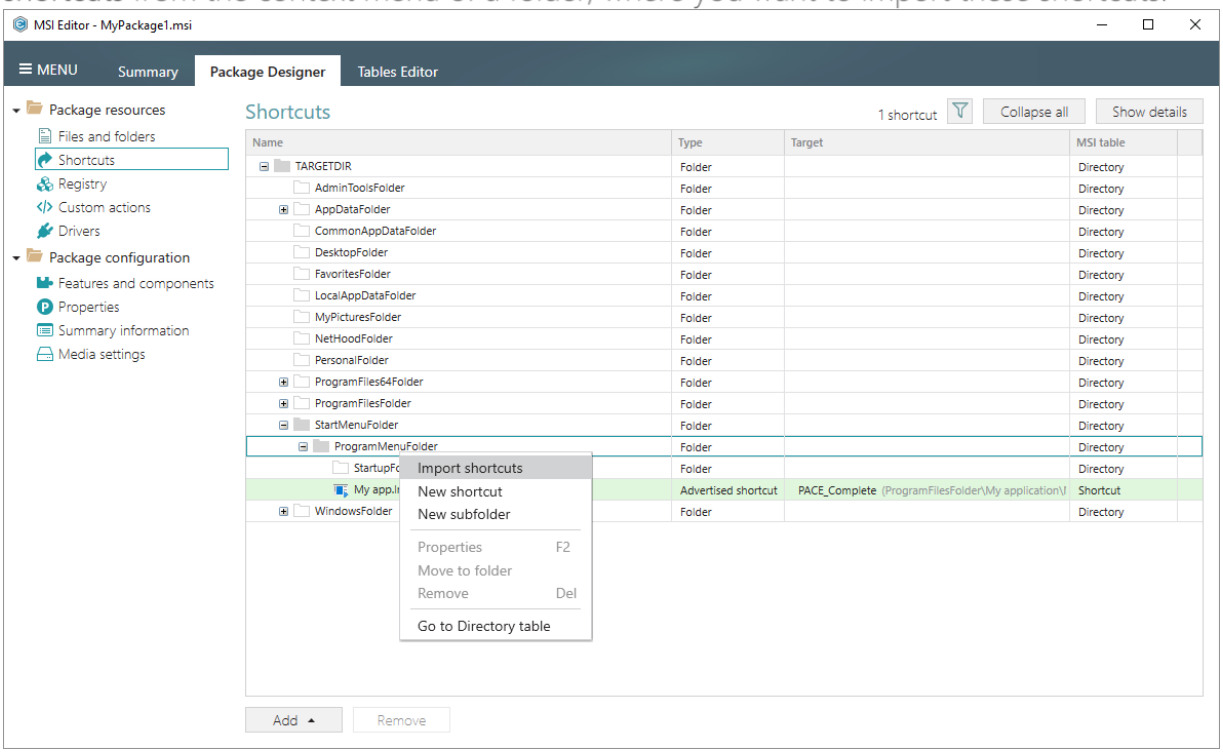
Then, click **Next**.



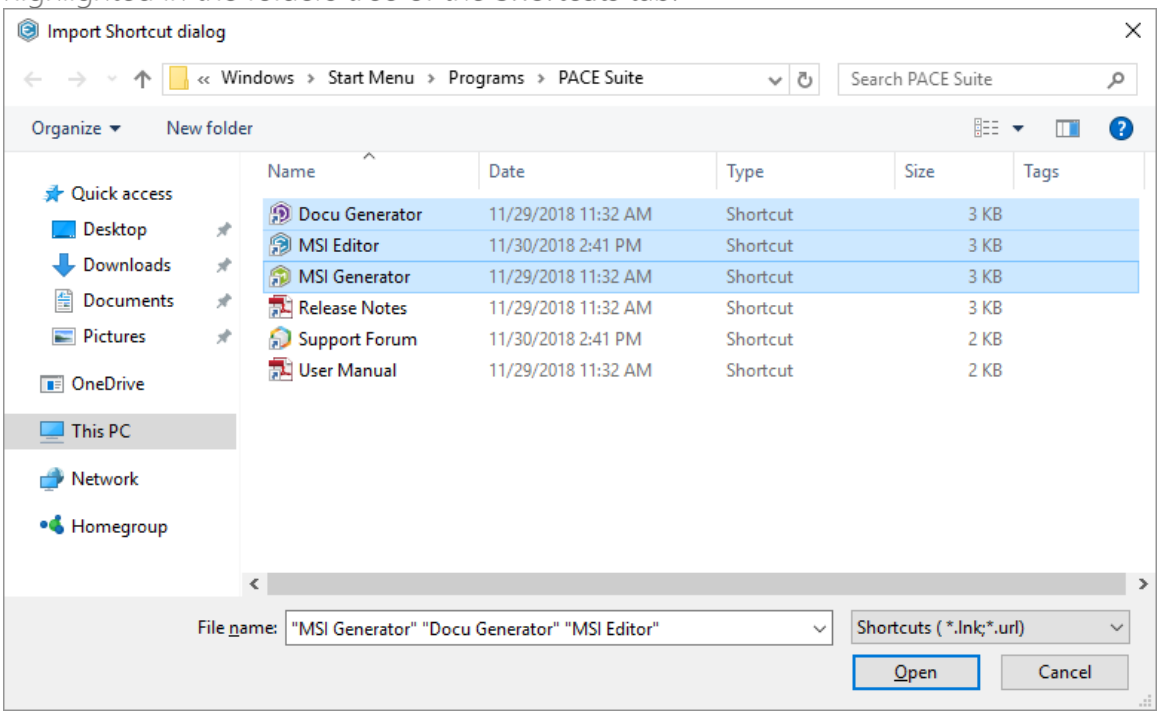
- On the last step, enter a **Shortcut name**, choose a shortcut icon and, if needed, specify the shortcut **Description**, **Hot key** and **Run** options. Click **Finish** to add the shortcut.



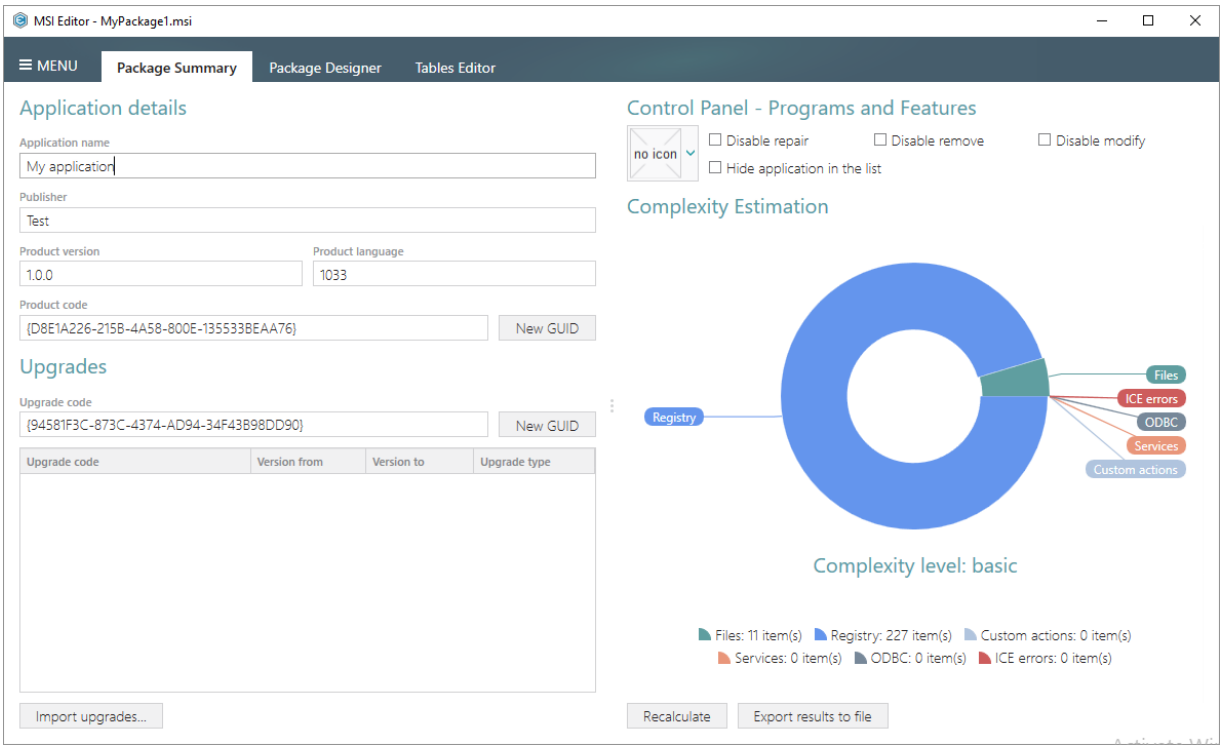
- In order to import existing shortcuts from the file system to the package, select **Import shortcuts** from the context menu of a folder, where you want to import these shortcuts.



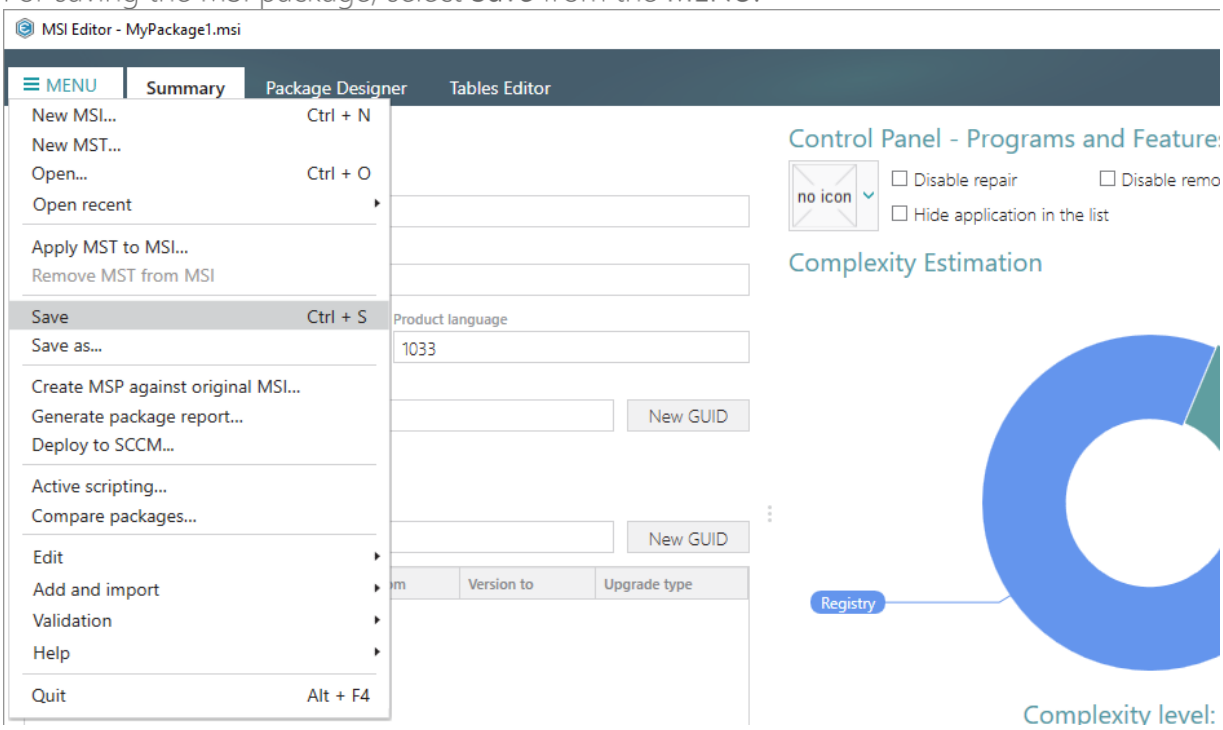

- Choose shortcuts you would like to import and click **Open**. The imported shortcuts will be highlighted in the folders tree of the **Shortcuts** tab.



- Finally, go to the Package Summary tab and enter the Application name, Publisher, and Product version.



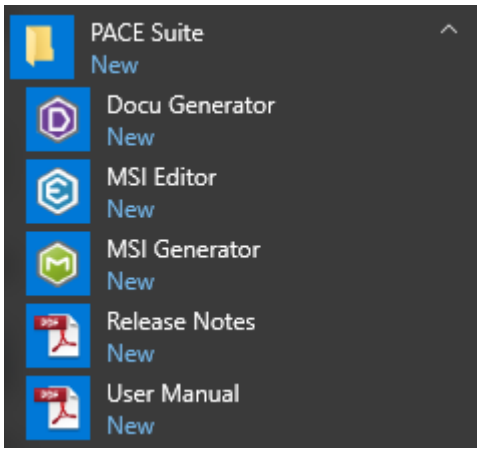

- For saving the MSI package, select Save from the MENU.



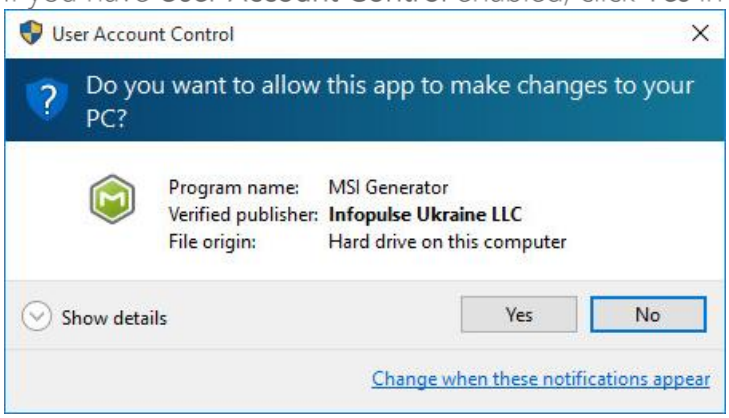

## 3.1.2 Generate MSI from Scratch

Create a new empty project in MSI Generator, add folders, files, registry entries and shortcuts to this project and then generate the MSI package from this project.

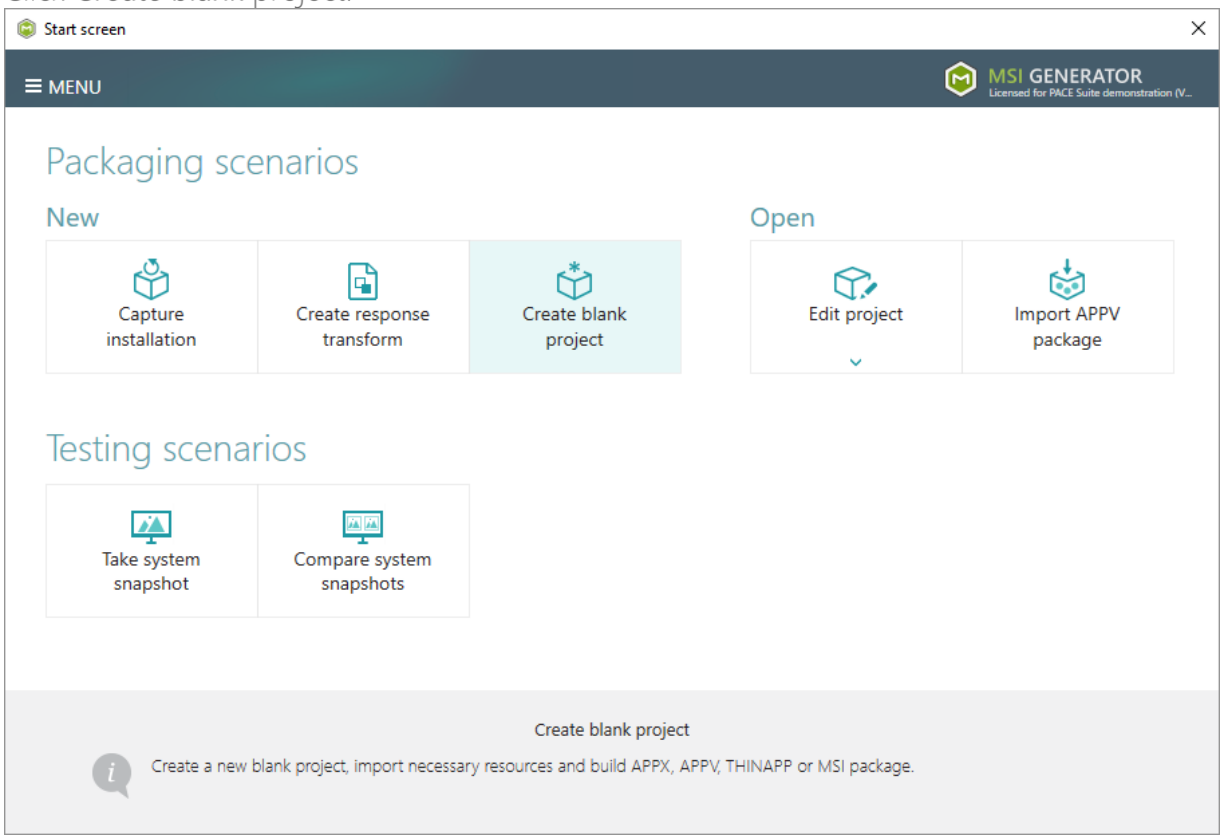
[1]. Launch MSI Generator from the desktop or the start menu shortcut.



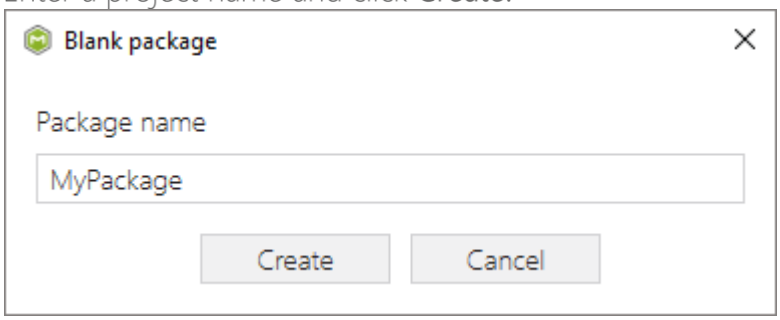

[2]. If you have User Account Control enabled, click Yes in the opened window.




[3]. Click Create blank project.

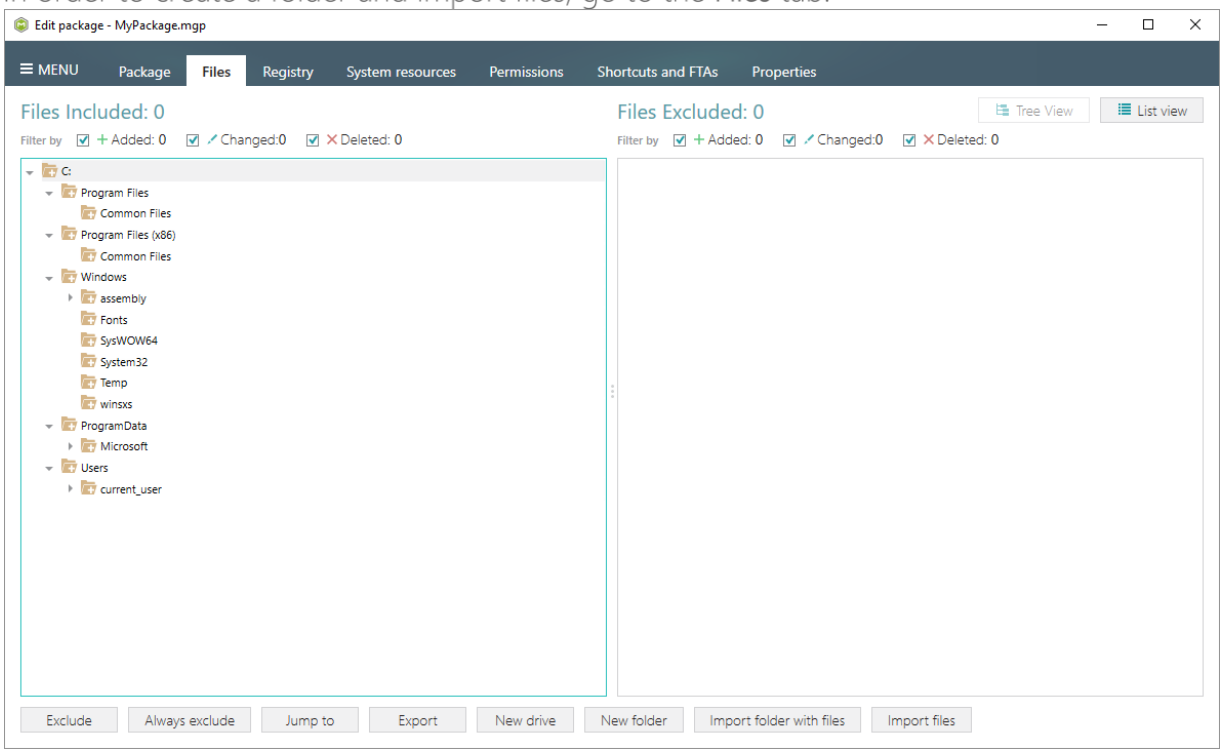


[4]. Enter a project name and click **Create**.



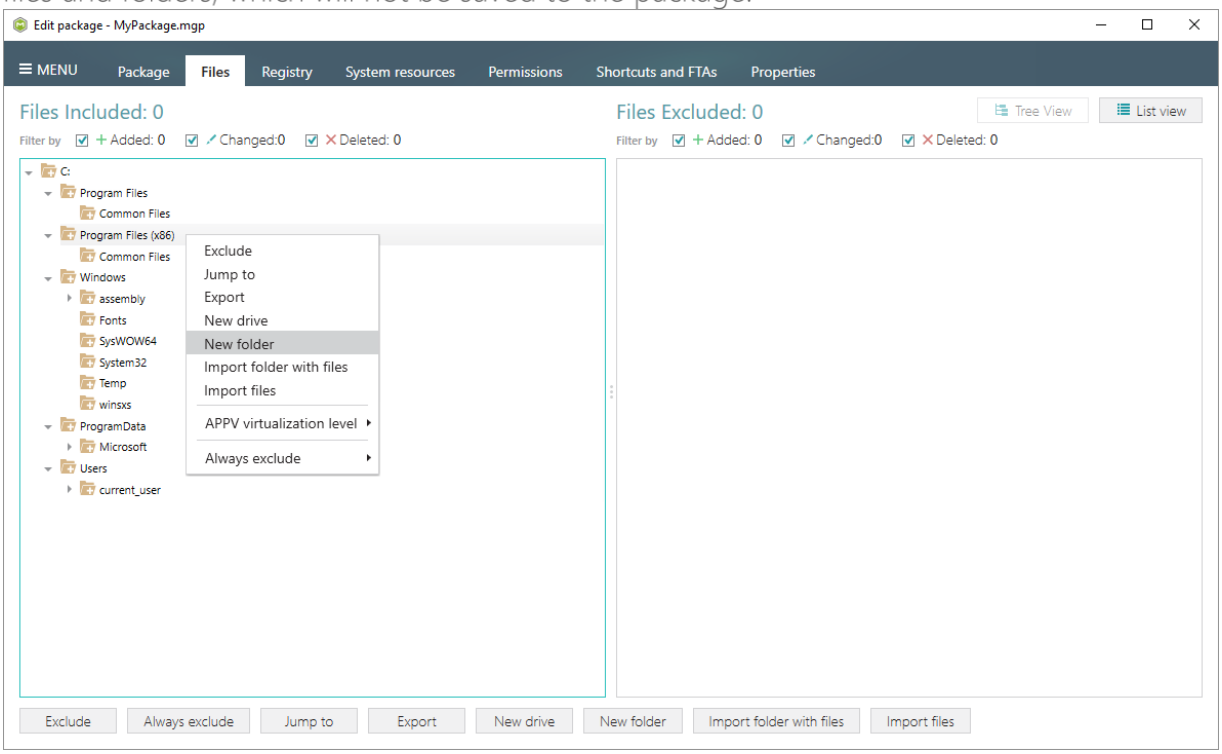

[5]. Once the newly created project is opened automatically, you can start adding the necessary resources using MSI Generator. The steps from [6] to [10] describe how to create a new folder and import files to this folder; the steps from [11] to [13] describe how to import registry entries from the REG file; the steps from [14] to [17] describe how to import shortcuts to the project; and two last steps [18] and [19] describe how to build MSI package from the prepared project.

[6]. In order to create a folder and import files, go to the **Files** tab.

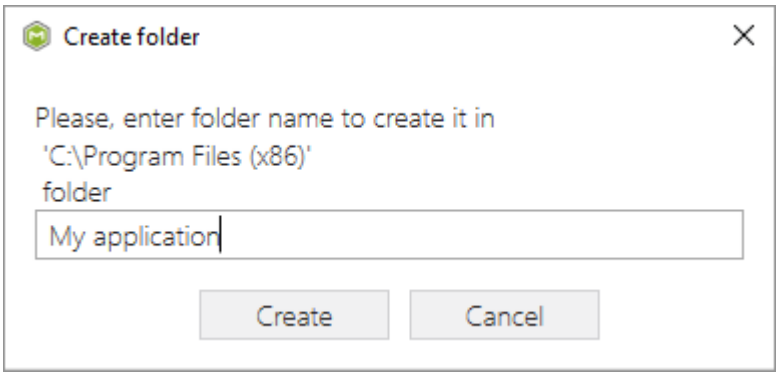



[7]. In the left pane, select **New folder** from the context menu of a folder, where you want to create a new empty folder. Note that the left '**Files Included**' pane displays files and folders, which will be a part of your package, and the right '**Files Excluded**' pane displays

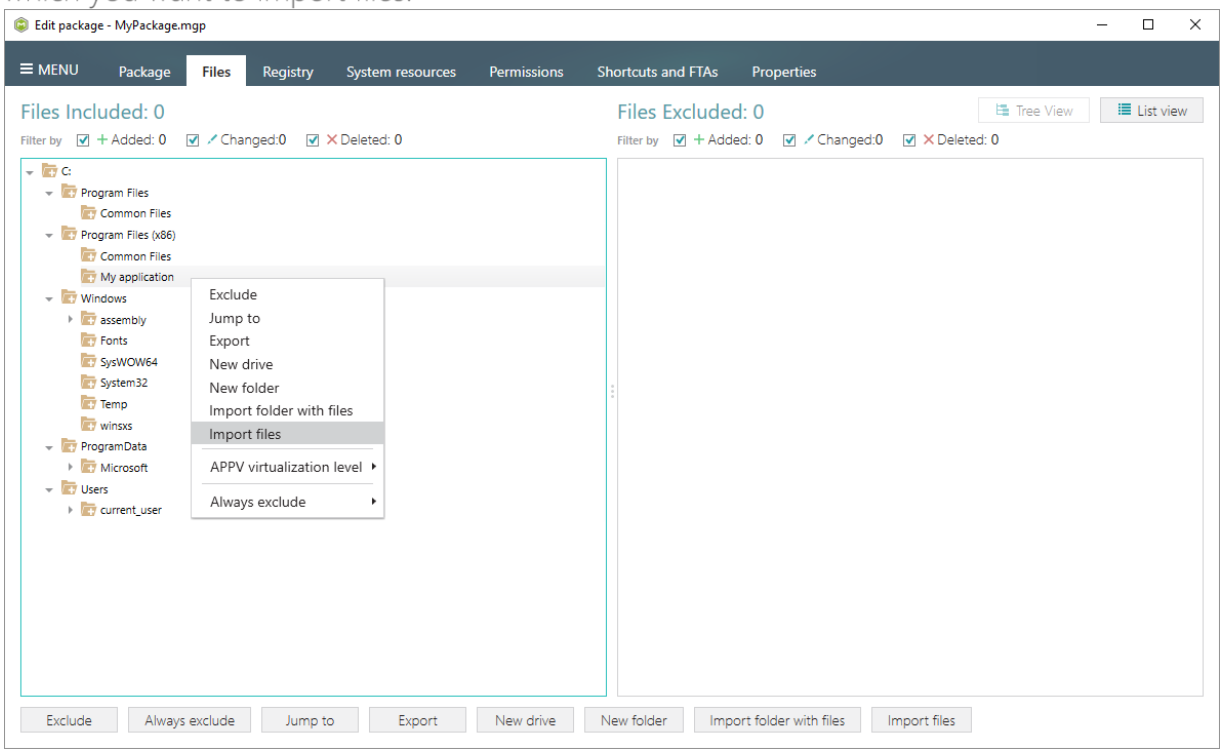
files and folders, which will not be saved to the package.



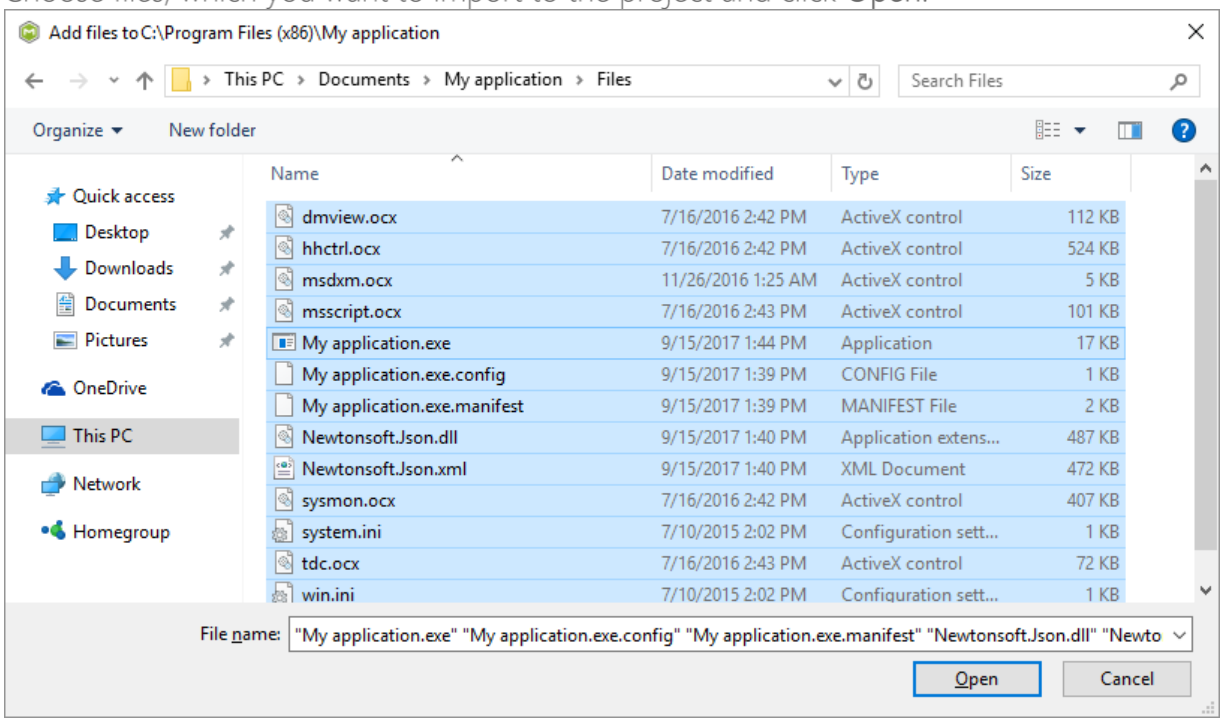

[8]. Enter a folder name and click **Create**.



[9]. For adding files to your project, select **Import files** from the context menu of a folder, to which you want to import files.

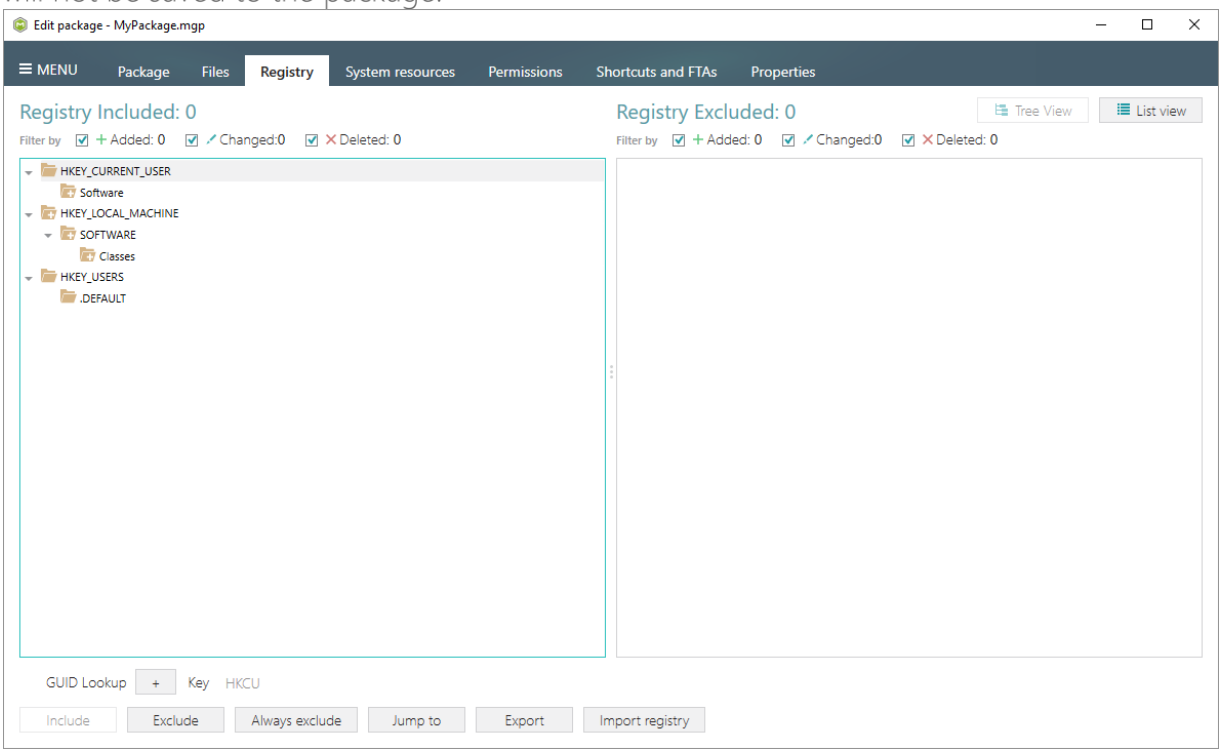



[10]. Choose files, which you want to import to the project and click **Open**.

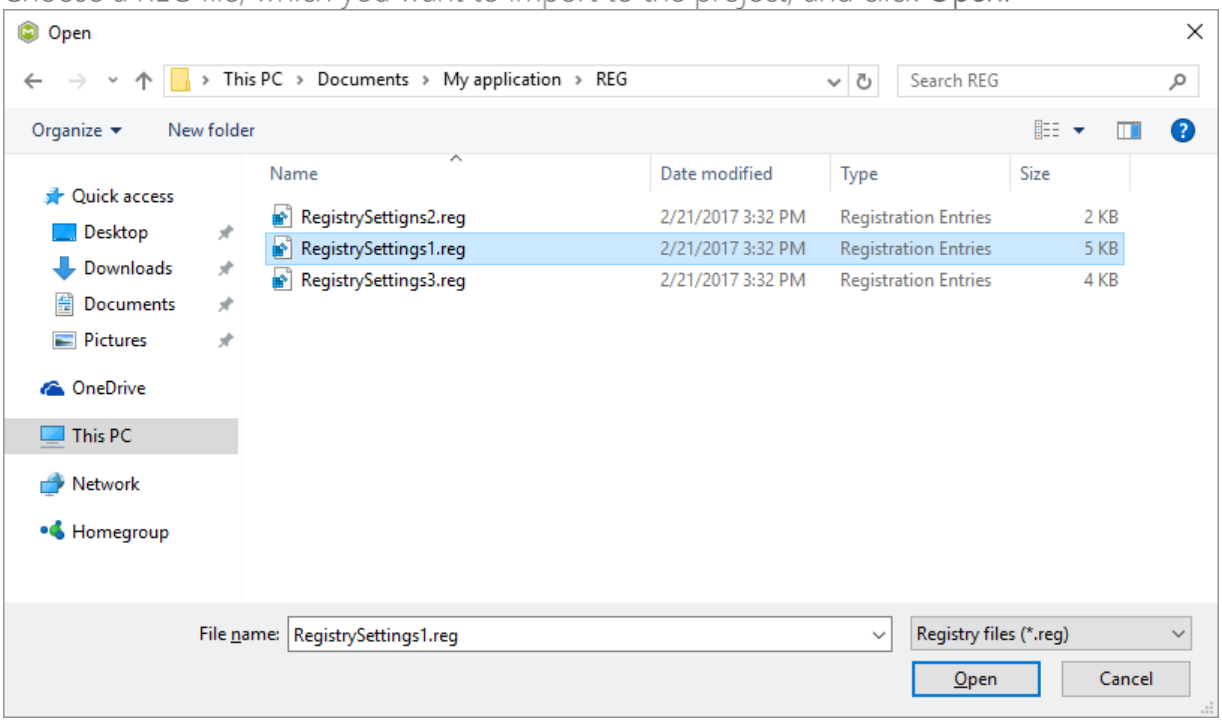



[11]. In order to import registry entries from the REG file, go to the **Registry** tab and click the **Import registry** button, located in the bottom part of the window. Note that the left '**Registry Included**' pane displays registry keys and values, which will be a part of your package, and the right '**Registry Excluded**' pane displays registry keys and values, which

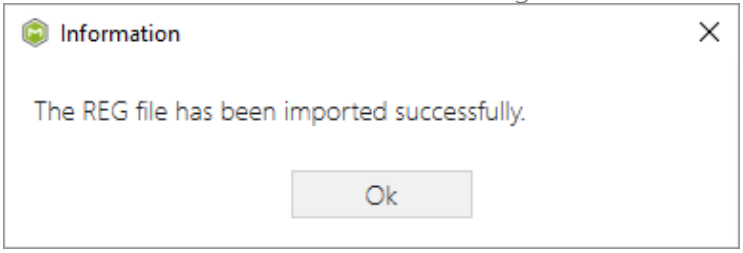
will not be saved to the package.



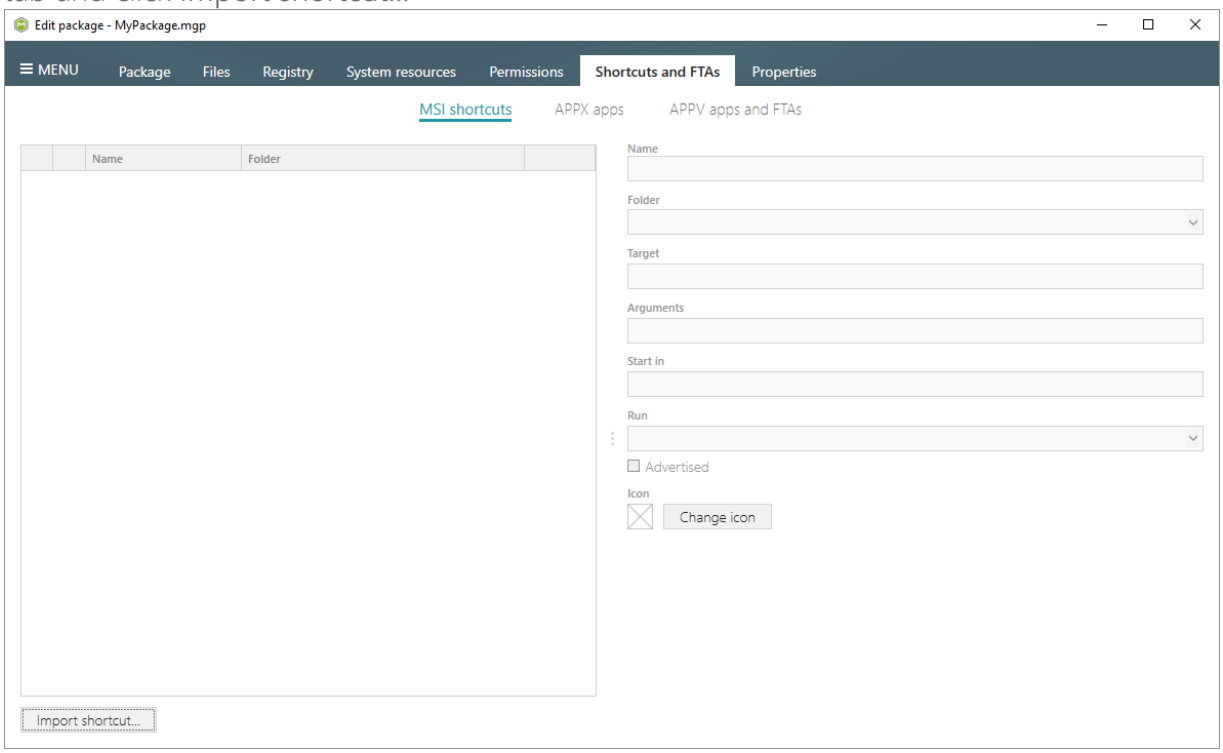

[12]. Choose a REG file, which you want to import to the project, and click **Open**.



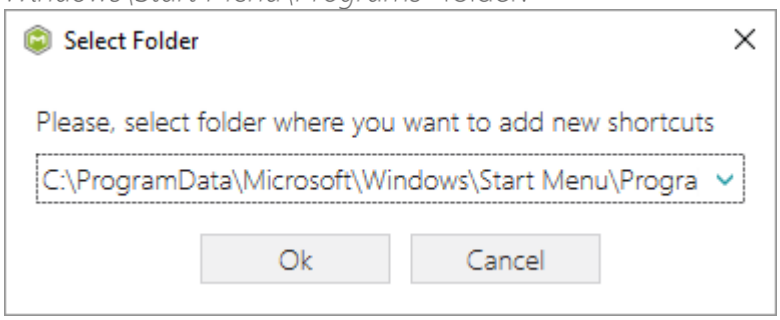

[13]. Click **Ok** to close the information dialog.



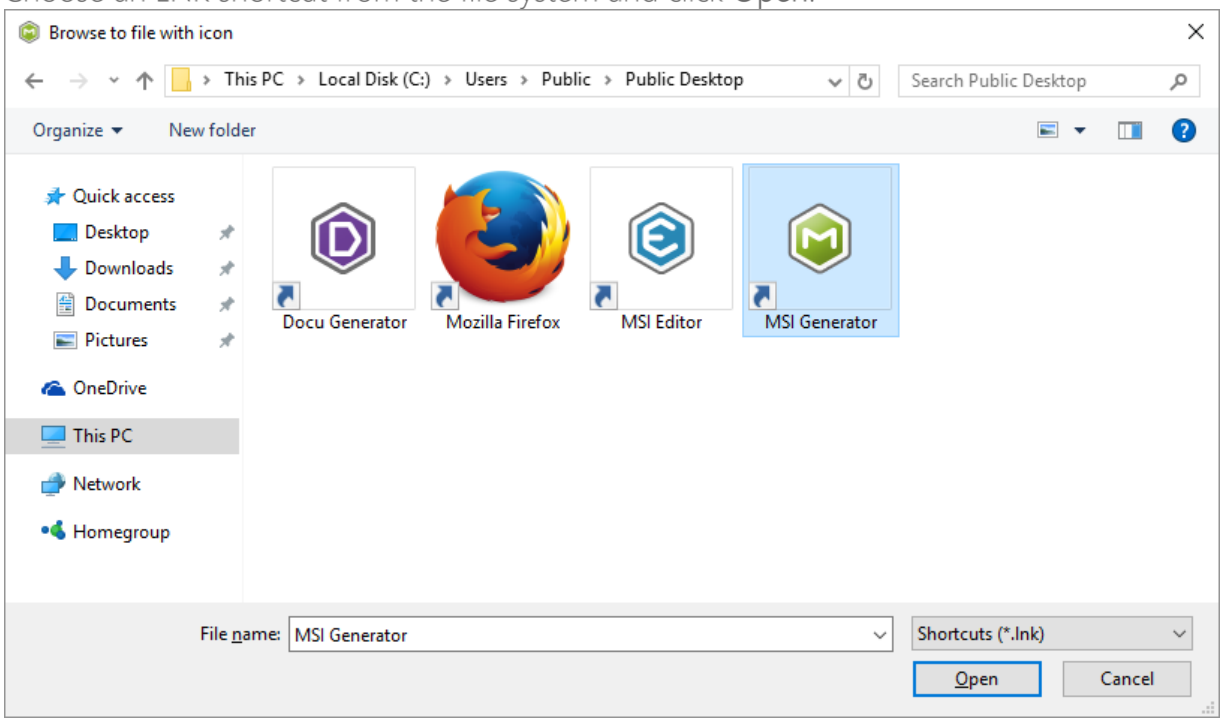
[14]. If you need to add shortcuts to the project, go to the Shortcuts and FTAs -> MSI shortcuts tab and click Import shortcut...



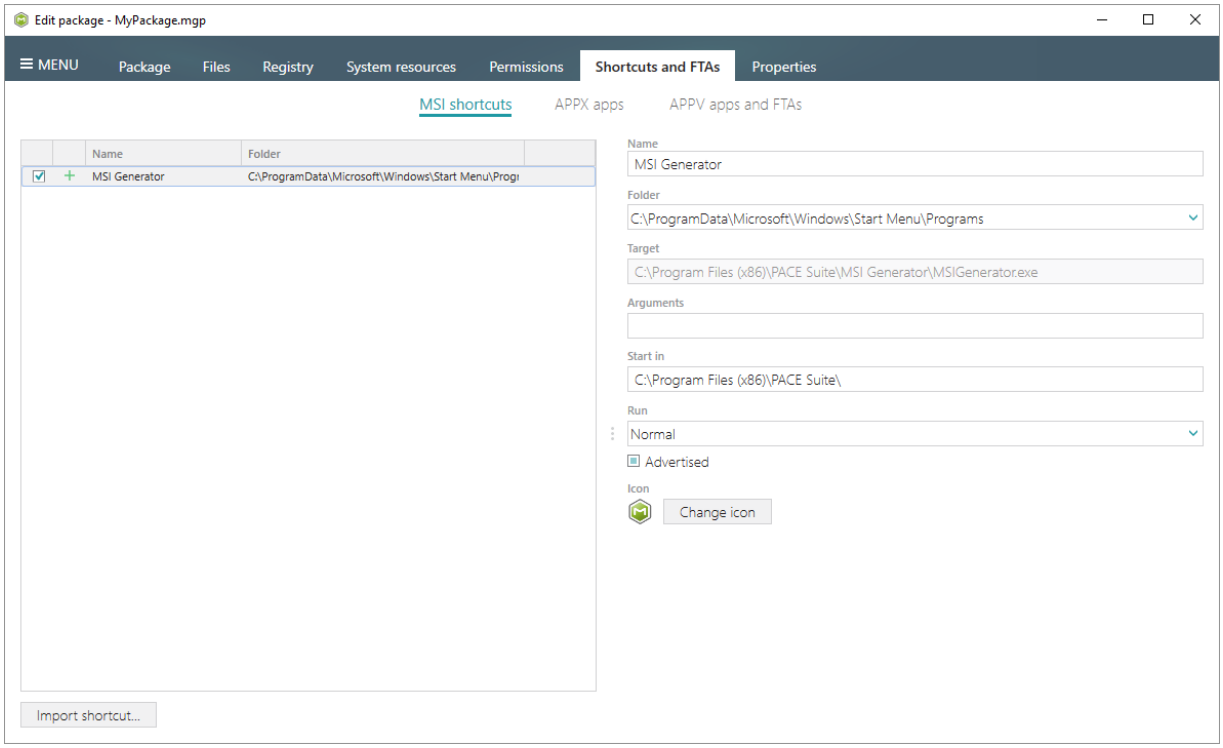

[15]. Select a folder from the list, where your newly imported shortcut will be installed, and clik Ok. Note that the Start menu shortcuts are located at "*C:\ProgramData\Microsoft\Windows\Start Menu\Programs*" folder.



[16]. Choose an LNK shortcut from the file system and click **Open**.

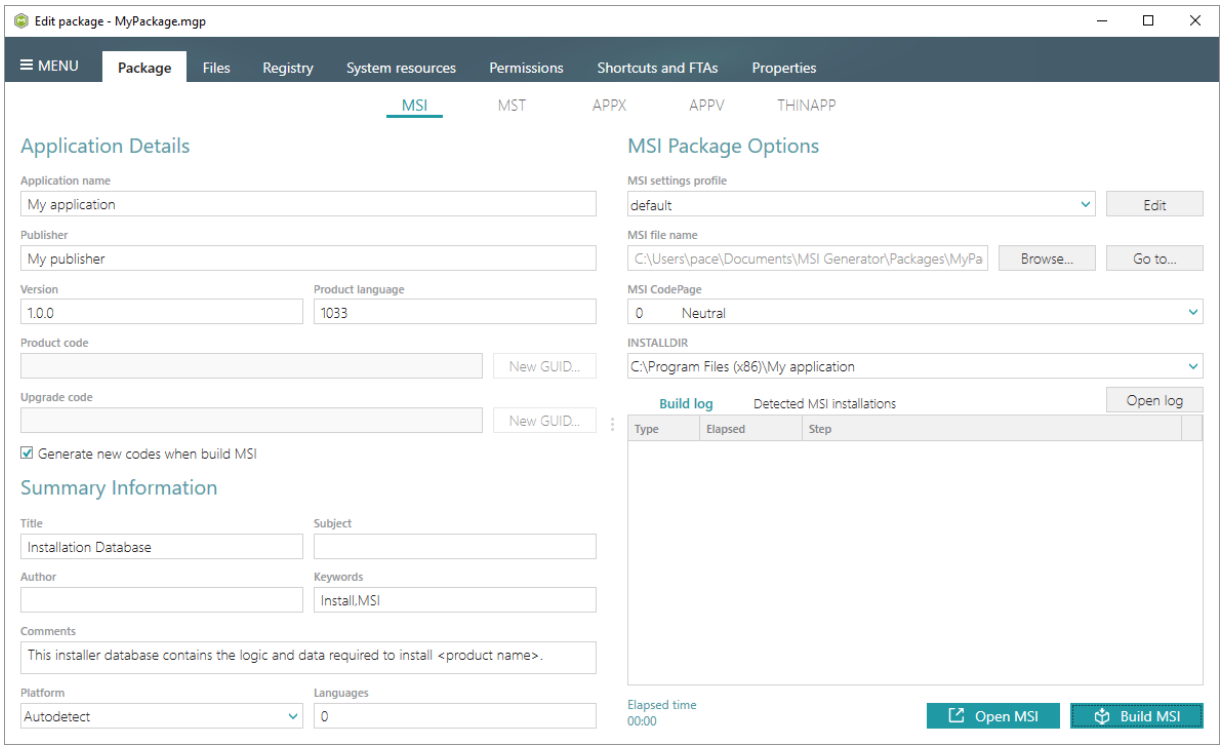



[17]. Using the details pane, you can update the shortcut **Name**, destination **Folder**, icon and other shortcut details.

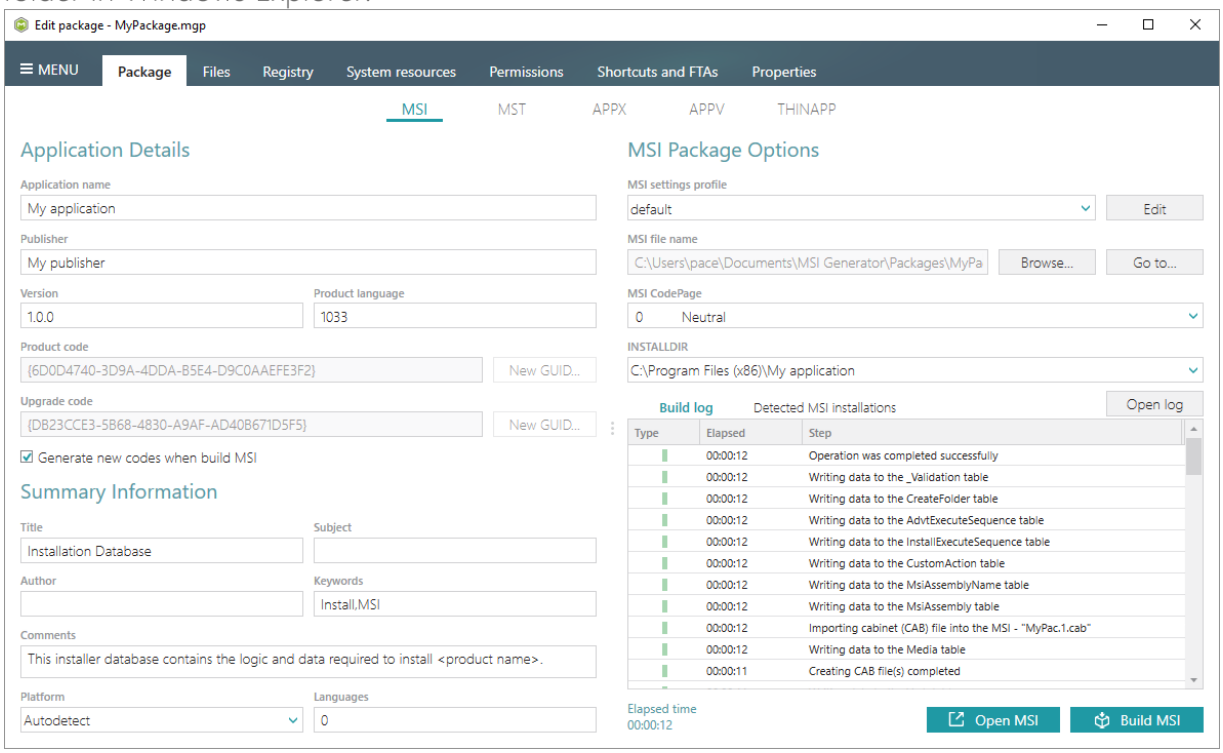



[18]. In order to build MSI package from your project, navigate to the **Package** -> **MSI** tab, update **Application Details** like name, publisher, version, select **INSTALLDIR** and click **Build**

MSI.



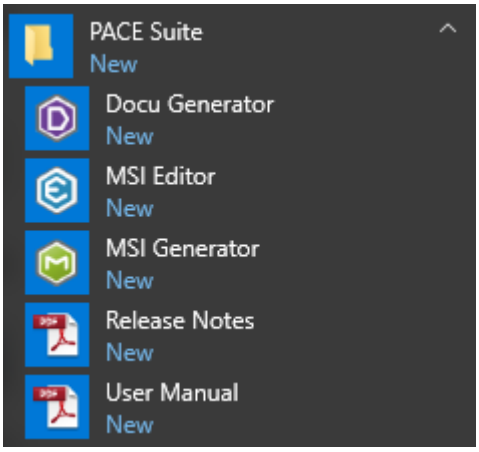

[19]. Click Go to..., located next to the MSI file name field, to open the package containing folder in Windows Explorer.



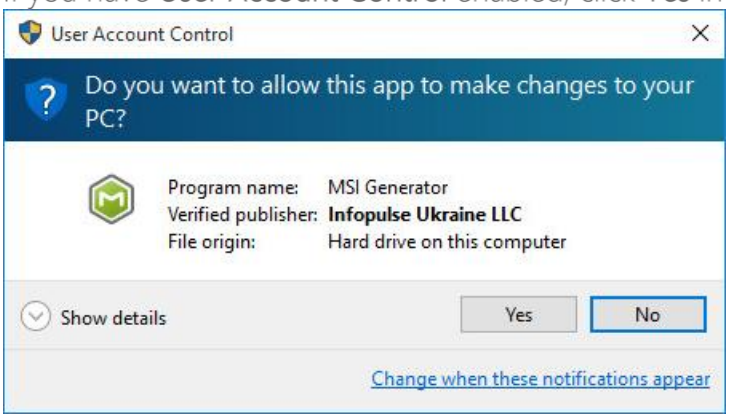
### 3.1.3 Convert (Repackage) EXE to MSI

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 file system and registry permission changes.

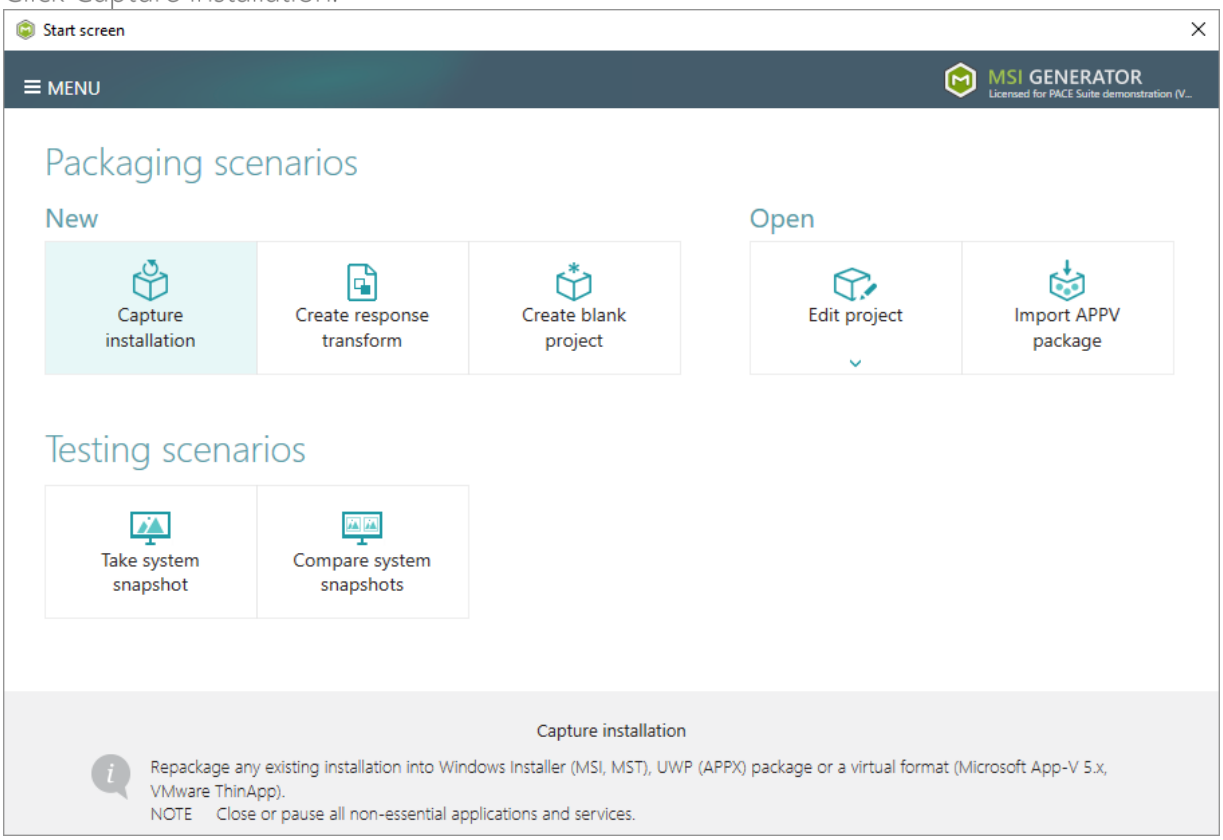
- Launch MSI Generator from the desktop or the start menu shortcut.



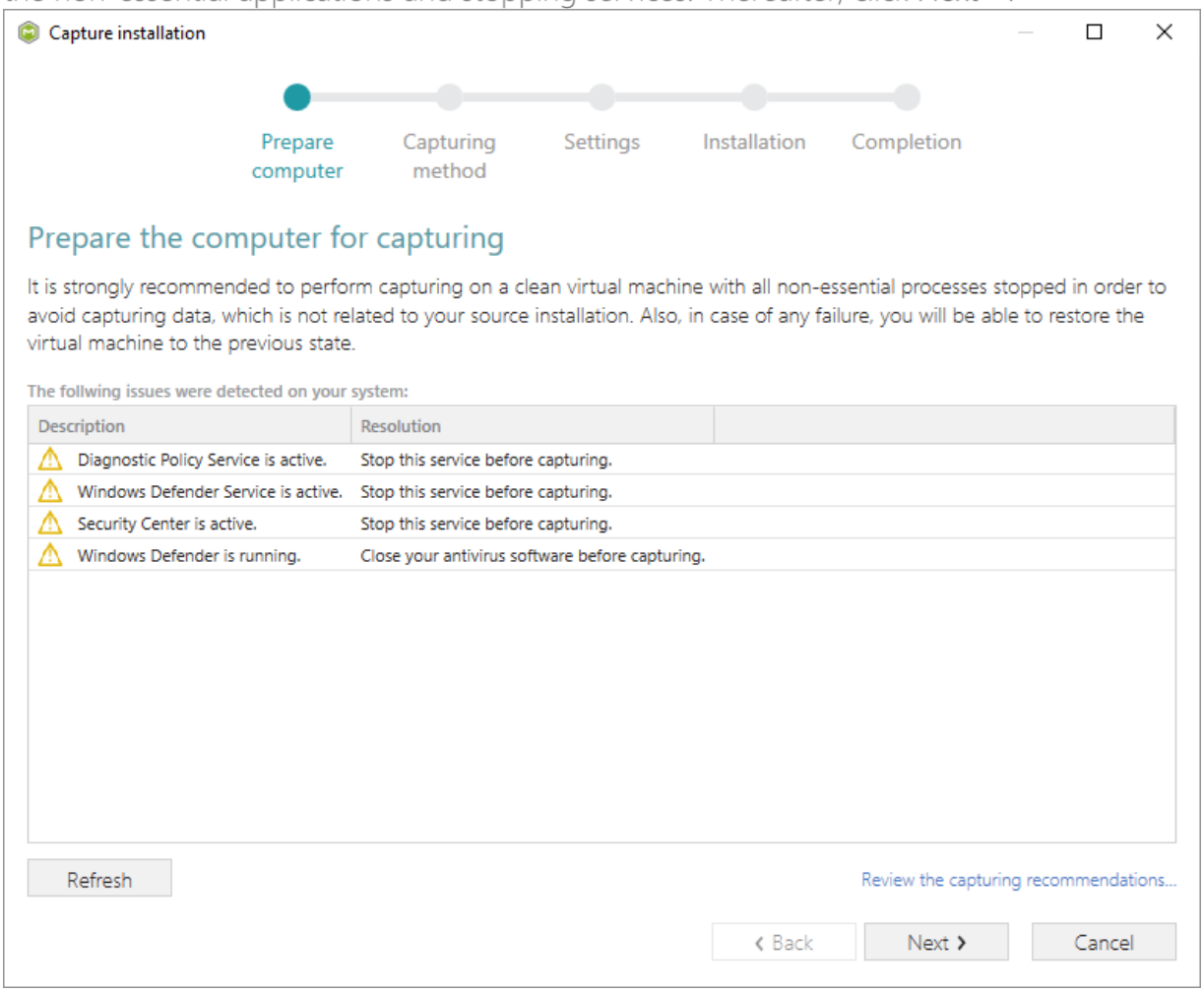

- If you have User Account Control enabled, click Yes in the opened window.



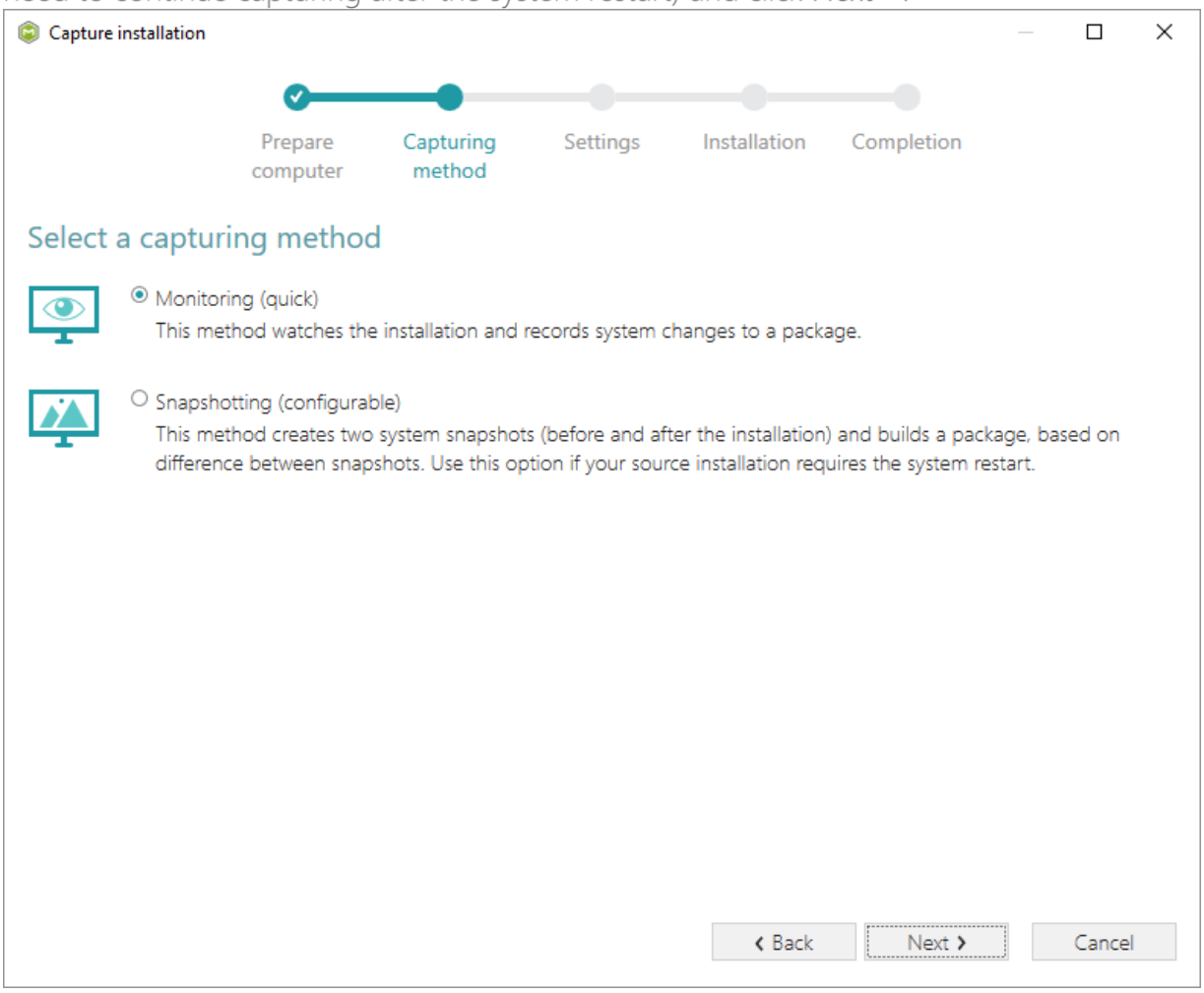
- Click Capture installation.



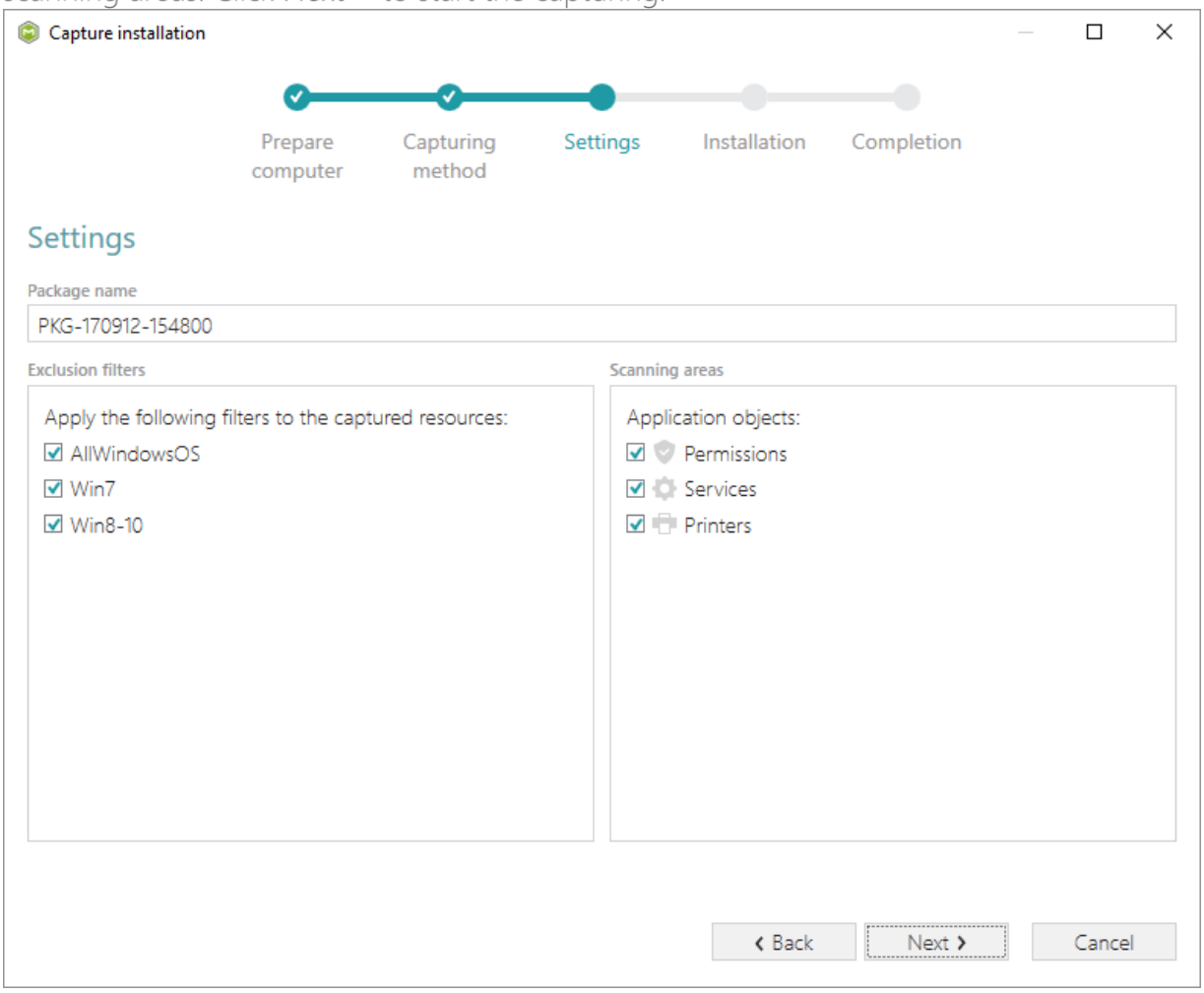
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



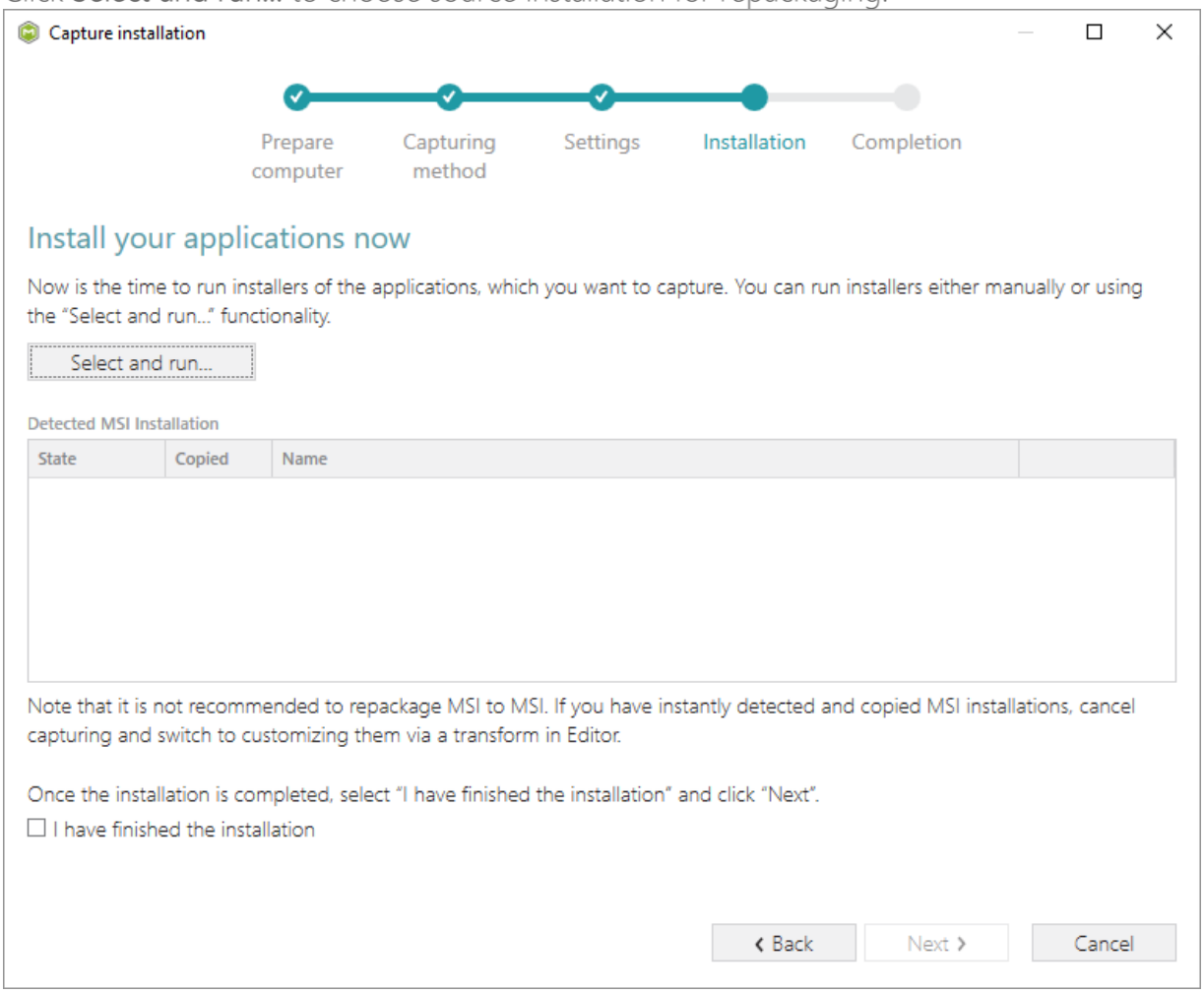
- Select the Monitoring method for the quicker capturing (or use the Snapshotting one if you need to continue capturing after the system restart) and click Next >.



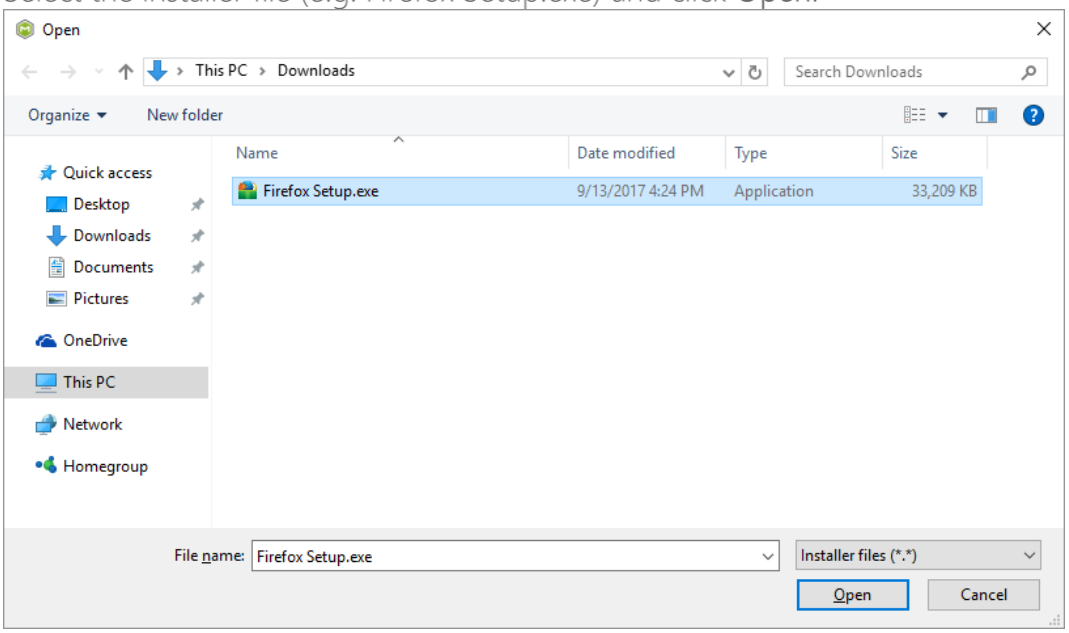
- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.



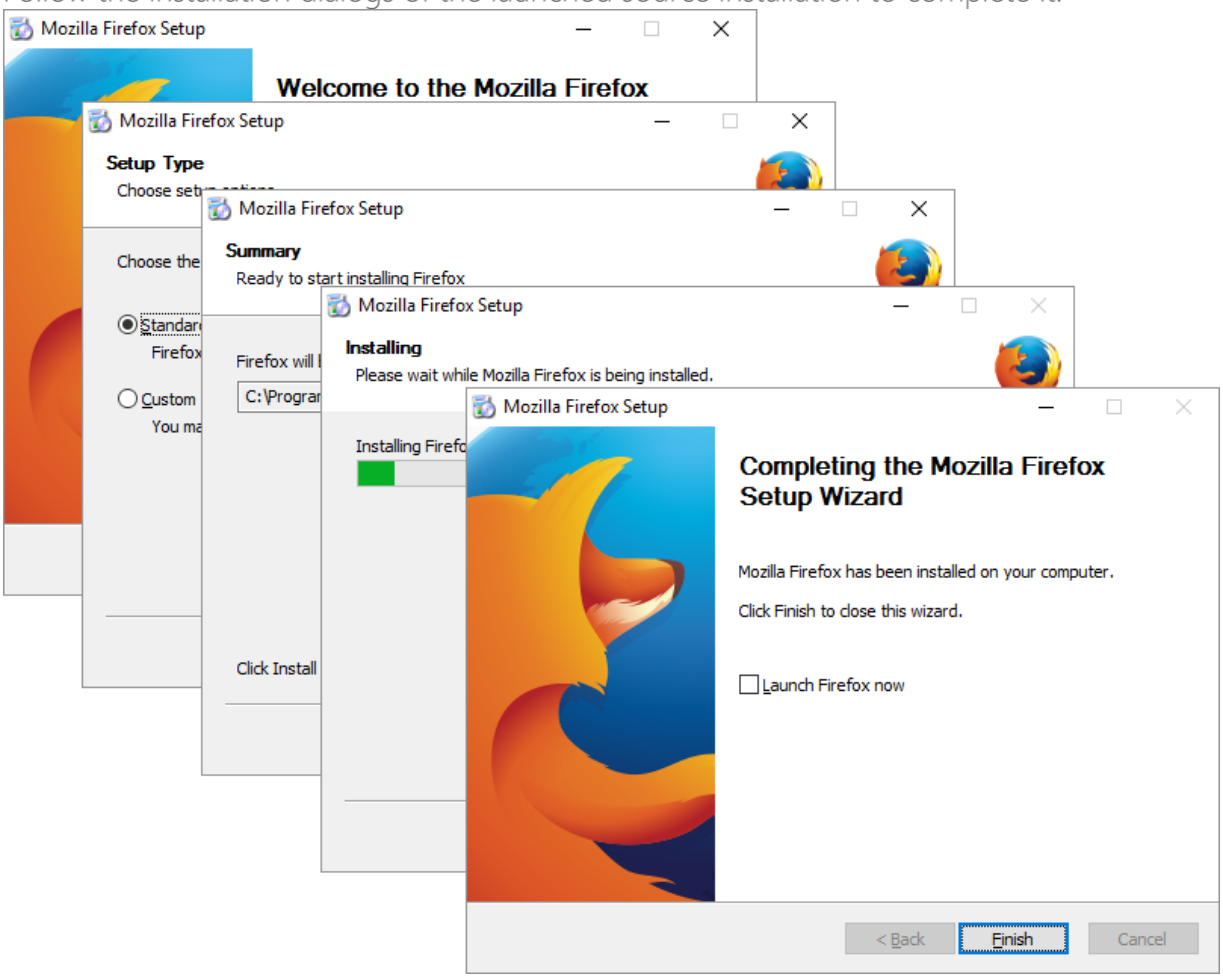
- Click Select and run... to choose source installation for repackaging.




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

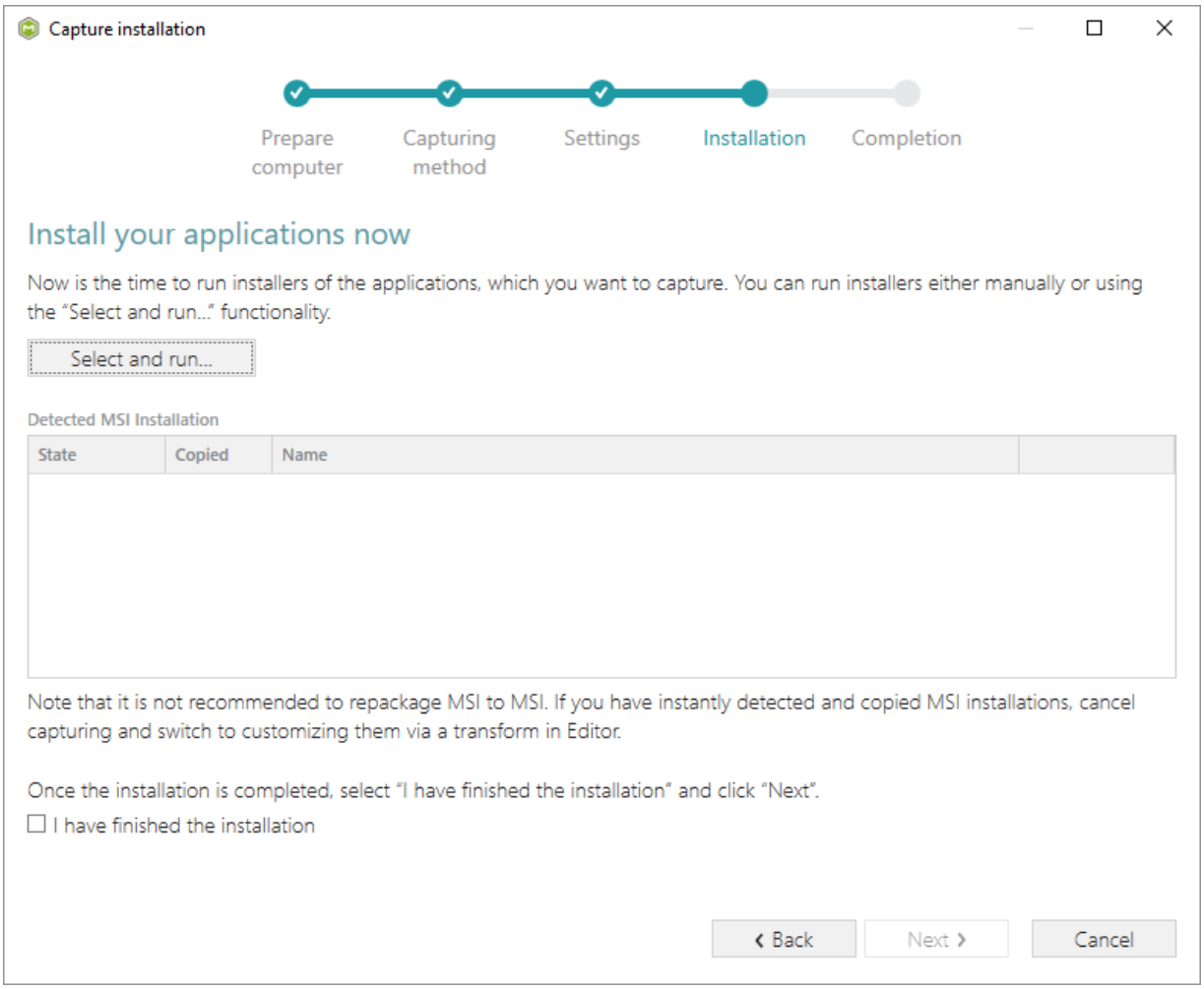


- Follow the installation dialogs of the launched source installation to complete it.



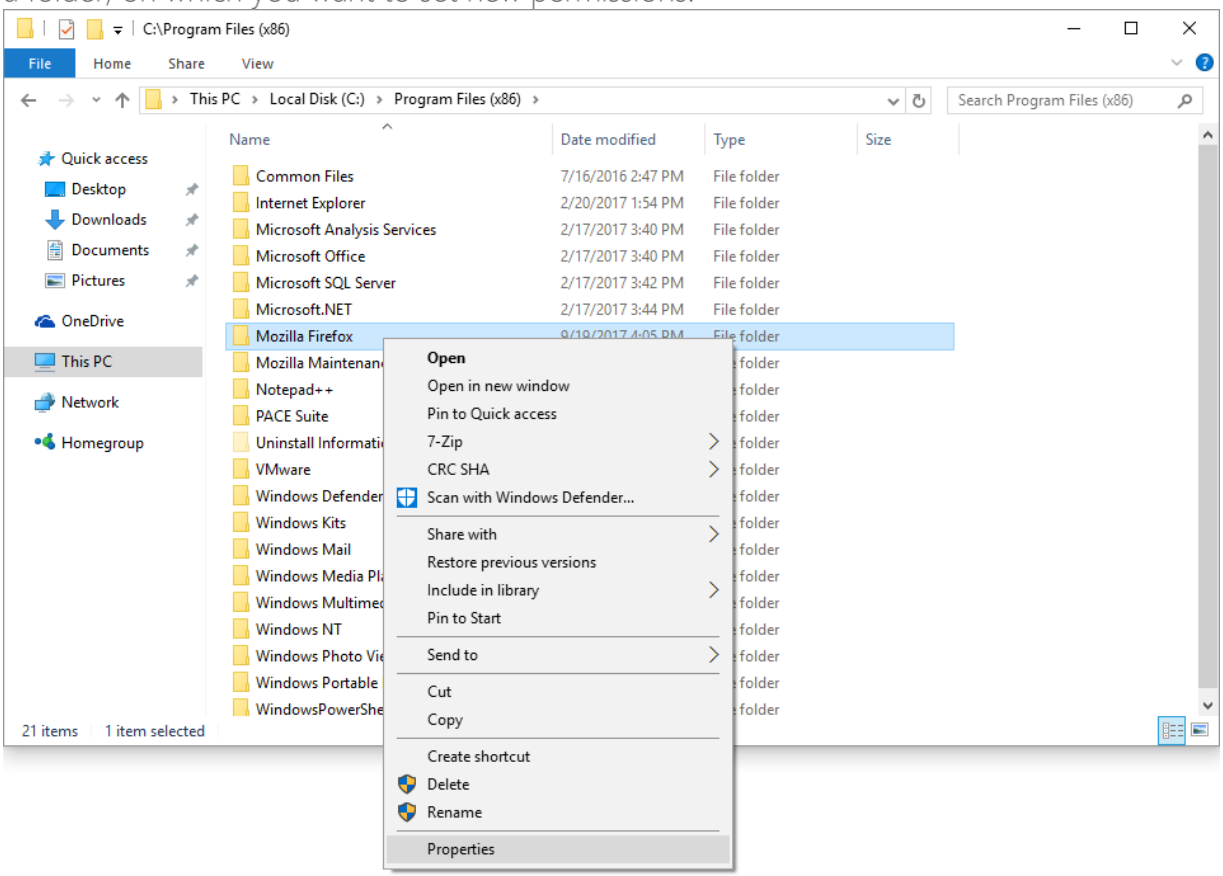

- Once the source installation is completed, we recommend to check if the Detected MSI installations tab does not contain found MSI installers. Even if your sources installation is an EXE file, it could contain embedded MSI installers, launched hiddednly. As you may know, repackaging existing vendor 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.

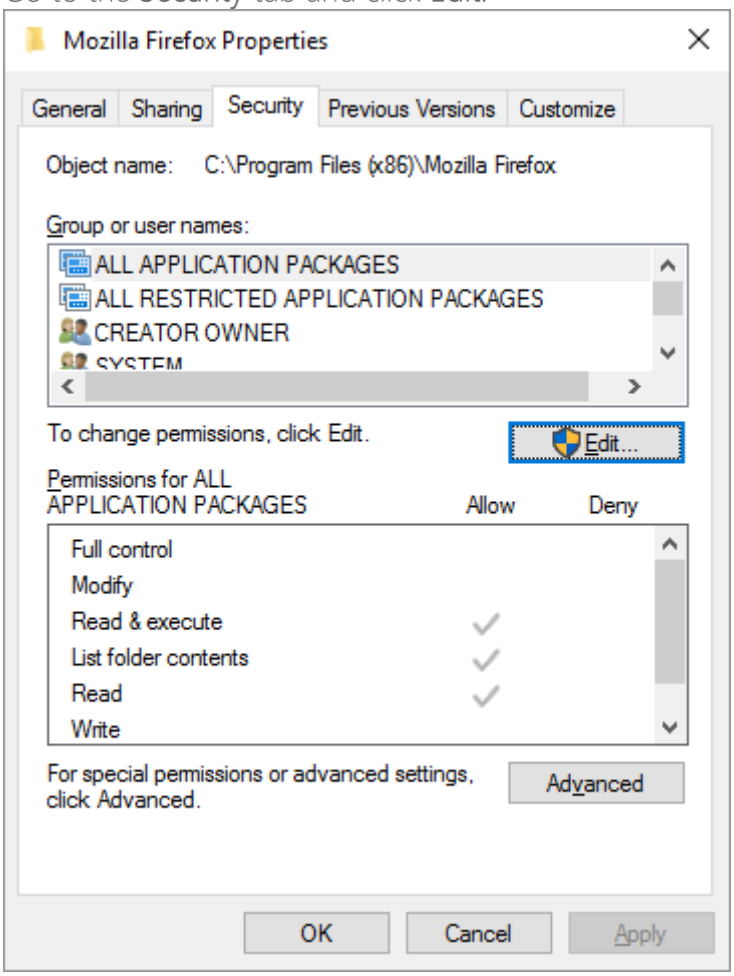



- Now you can make any changes to the file system and registry, which you want to capture and include to the package. For instance, you can create new or copy-paste existing files, import REG file to the system registry, changes permission settings, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.
- If you additionally need to change the default permission settings on a folder or a file and include them to the package, select Properties from the Windows Explorer context menu of

a folder, on which you want to set new permissions.

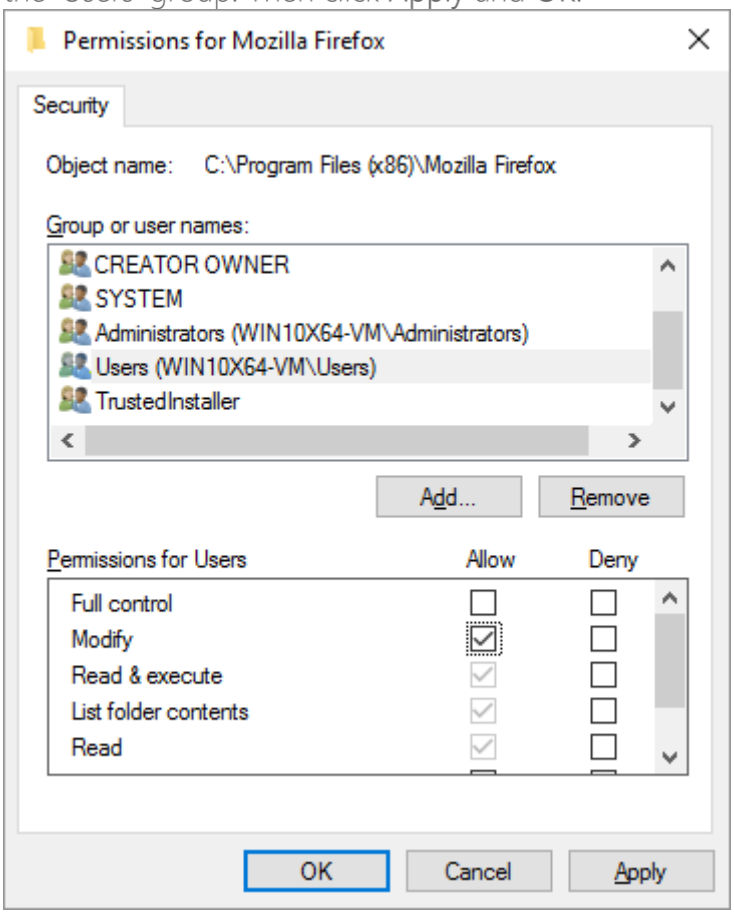


- Go to the Security tab and click Edit.

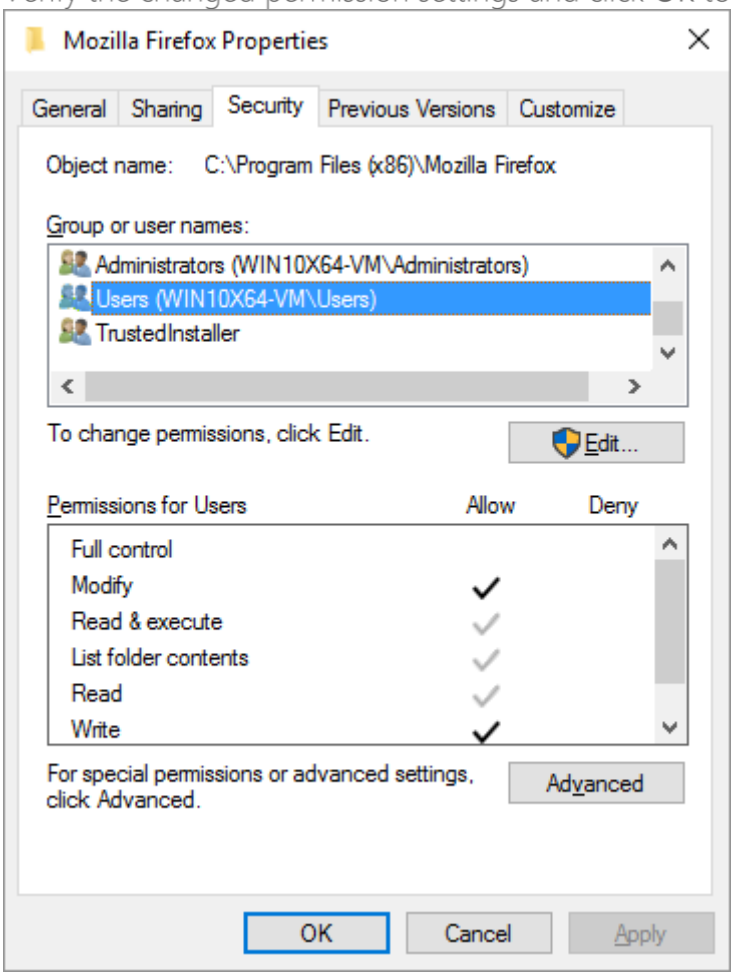



- Select a group or a user, for which you want to change the permissions and then select the necessary access rights. For example, lets set the 'Write' and the 'Modify' access rights for

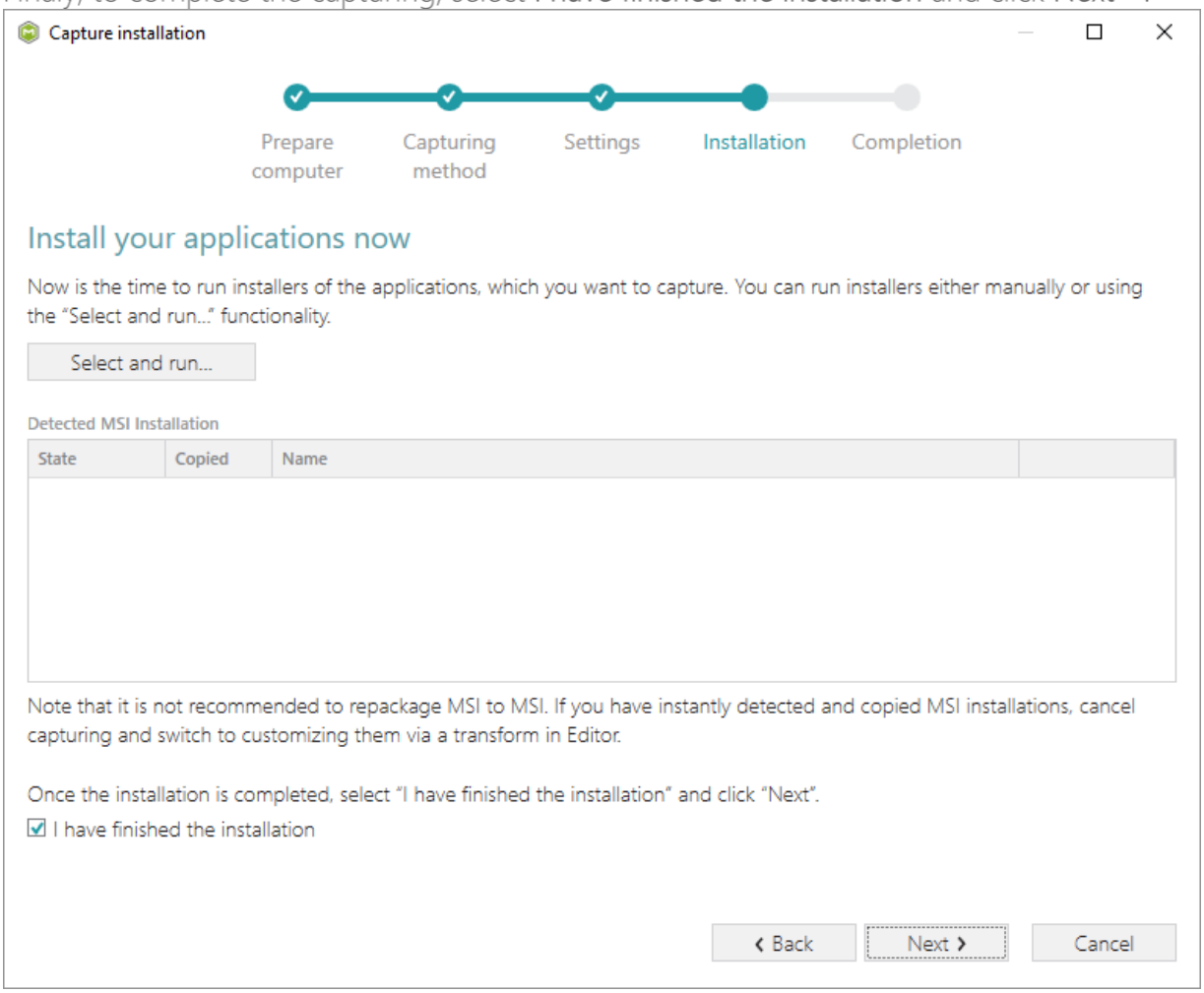
the 'Users' group. Then click Apply and OK.



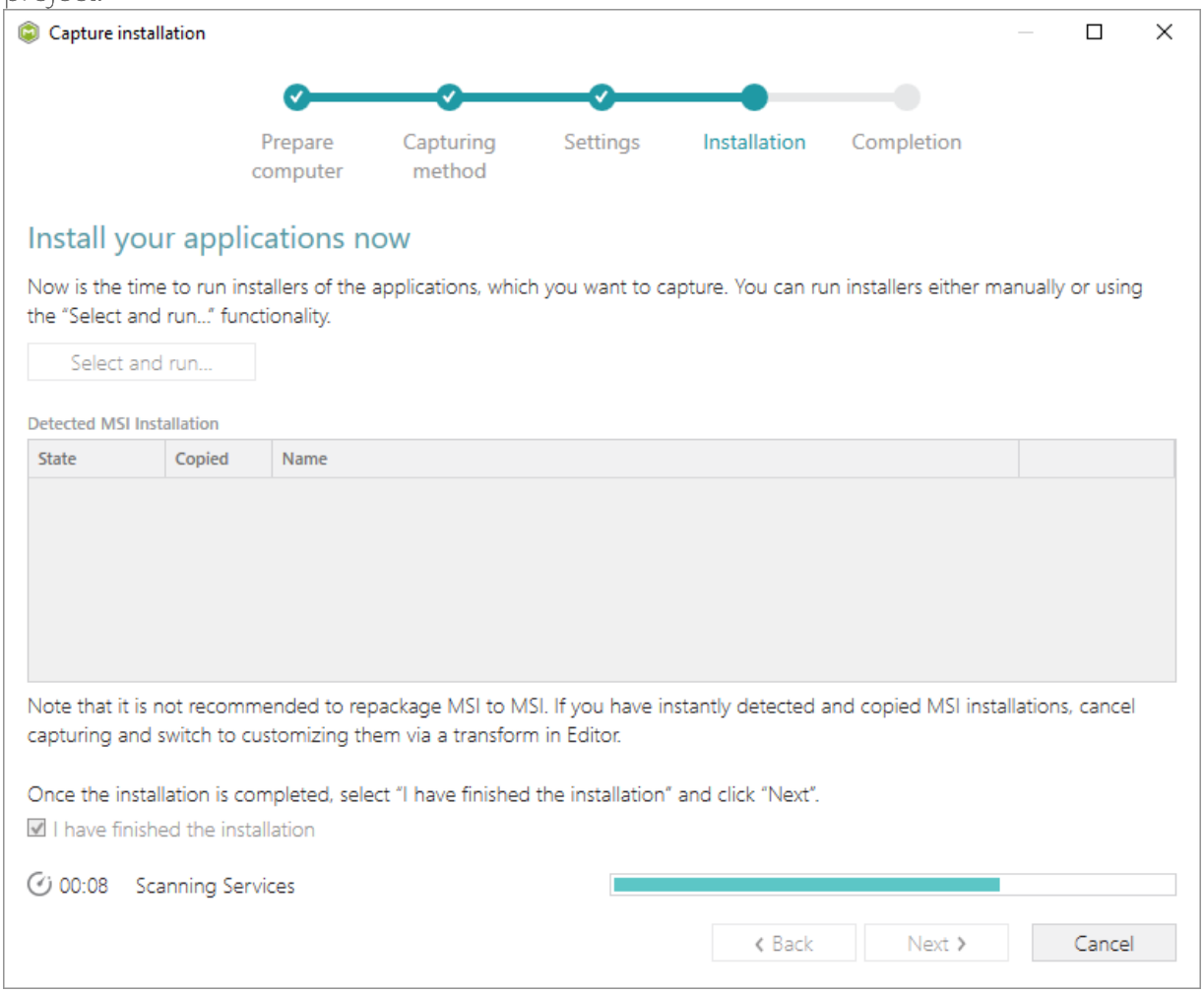
- Verify the changed permission settings and click OK to close the window.



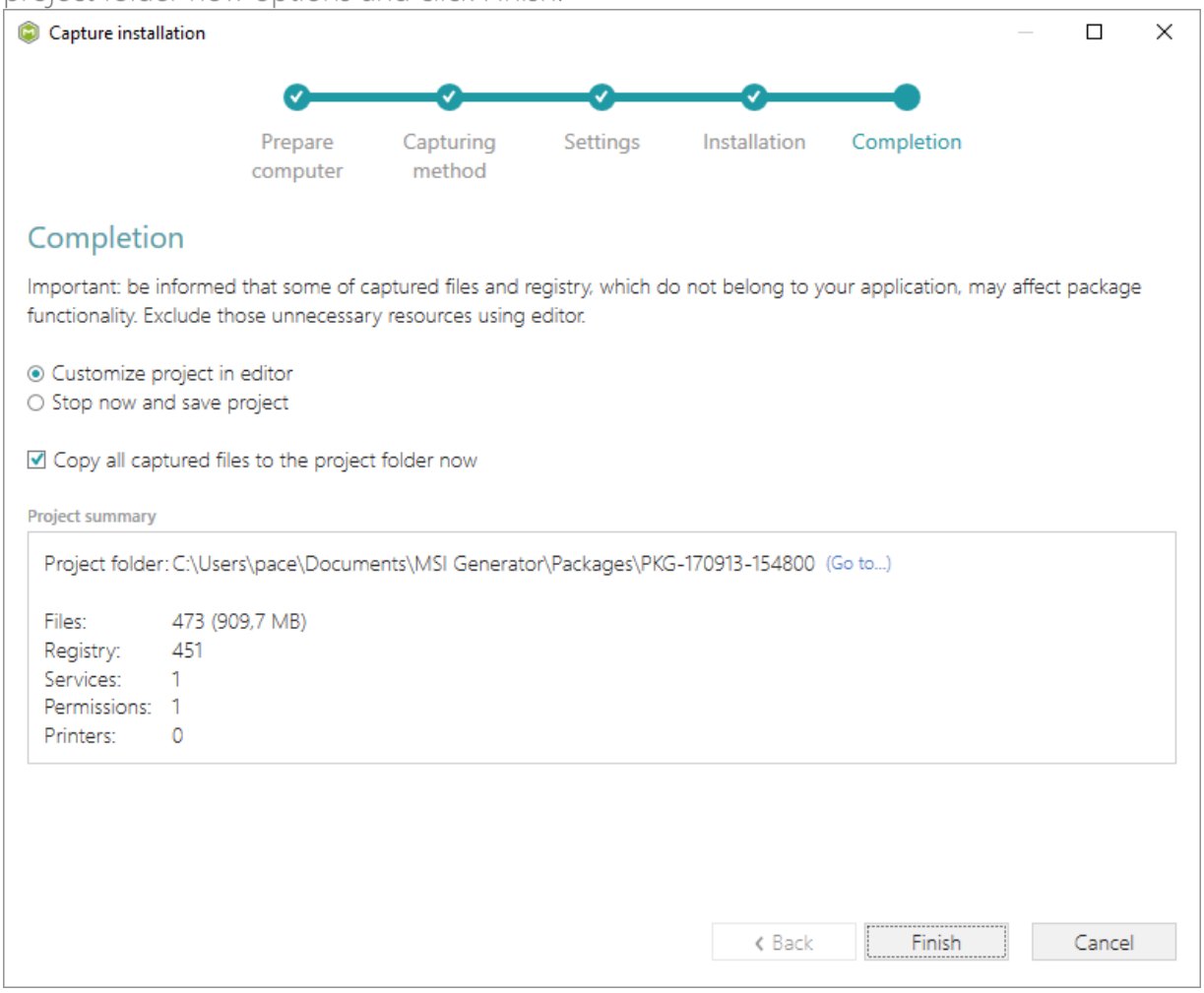
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



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

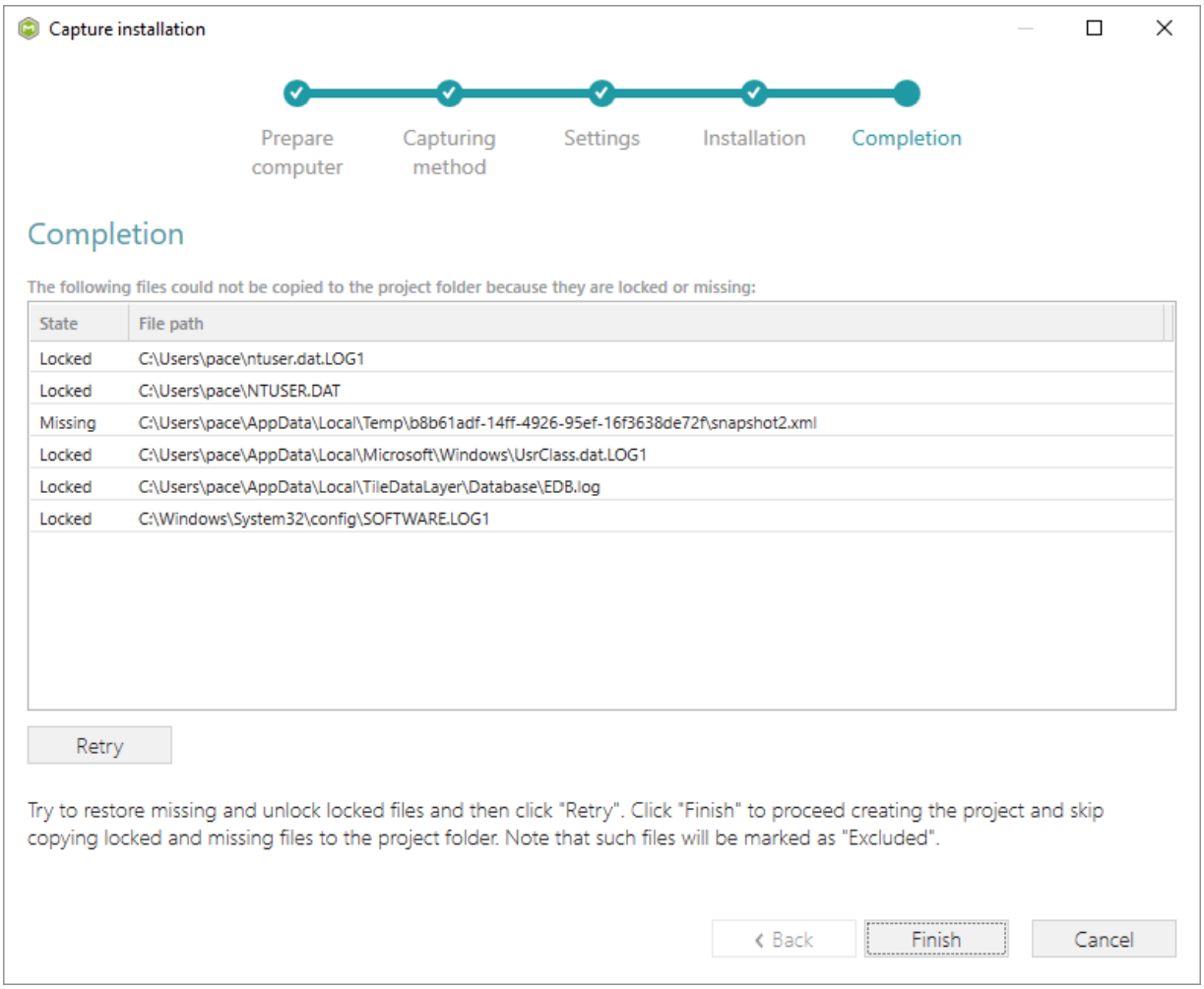


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



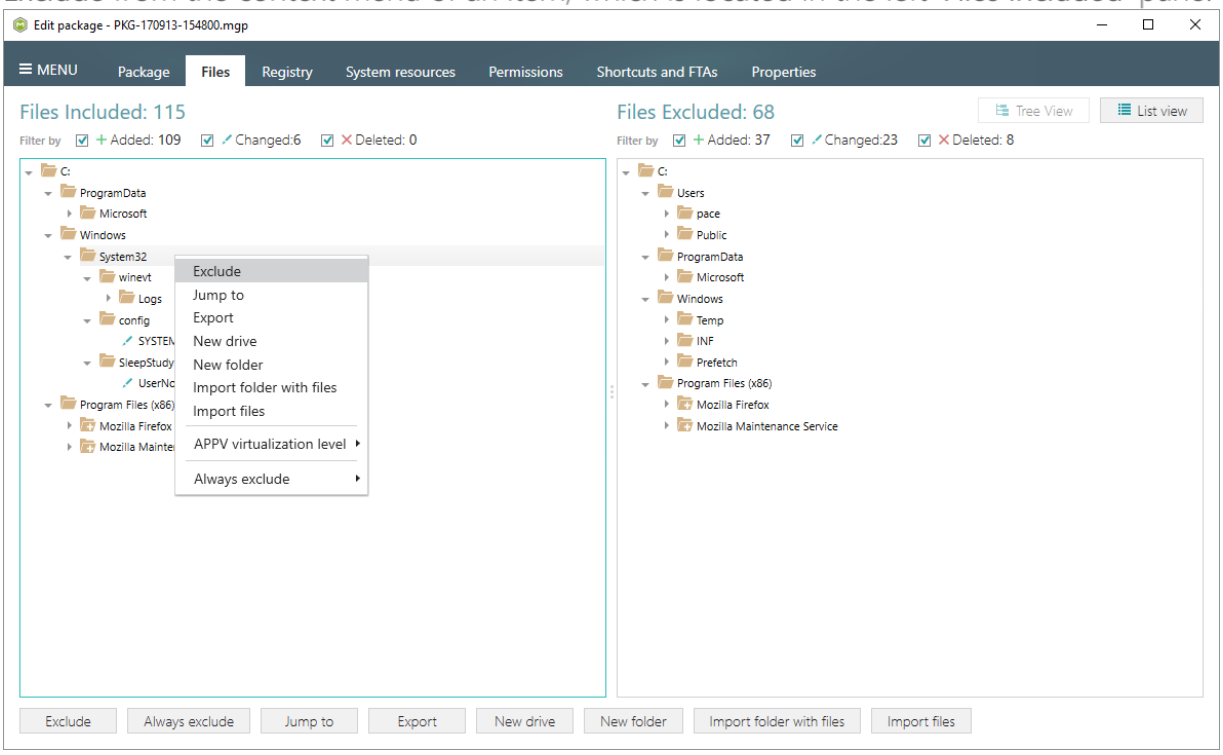

- The following dialog displays captured files, which could not be copied to the project folder because they do not exist anymore or locked by the system or by an application. Try to resolve these issues and then click Retry. Click Finish to skip copying the missing and locked

files and mark them as Excluded resources.

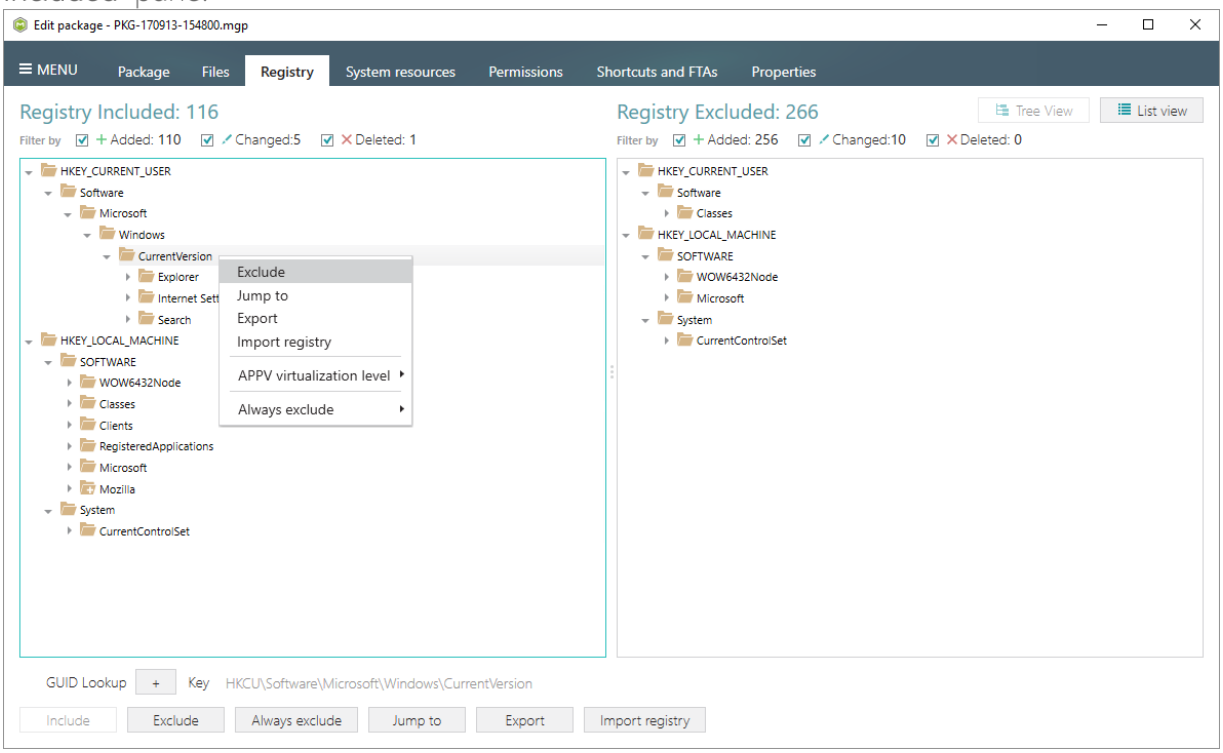



- Once the project is opened in the project editor, review the captured resources at the Files, Registry, System resources, Permissions, and Shortcuts and FTAs tabs and exclude unnecessary ones from the project. Unnecessary resources are files, registry entries, which are usually created or modified in the result of operating system work, and such resources could not be a part of your captured application. Unfortunately, there is no universal rule to discover which of captured files or registry entries should be excluded, so exclude only those ones, which almost 100% do not refer to your captured application (e.g. NOD32 antivirus files couldn't be a part of Firefox application).

- 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.

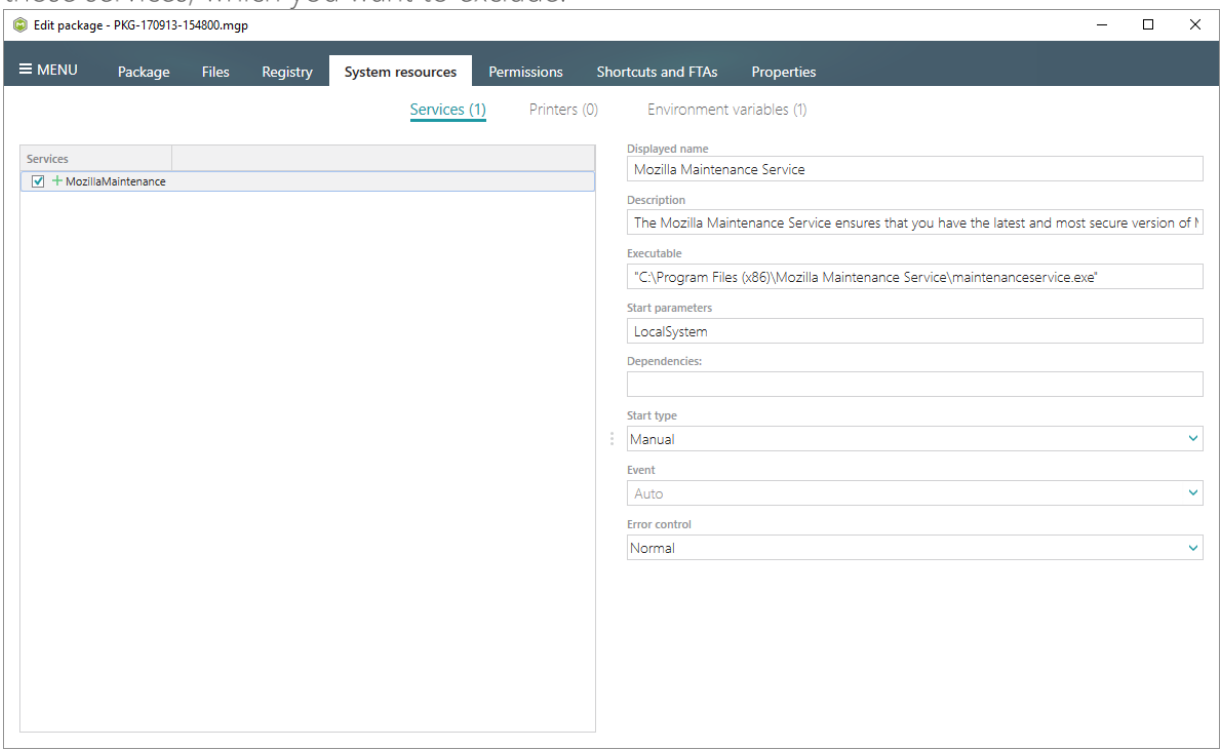



- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.

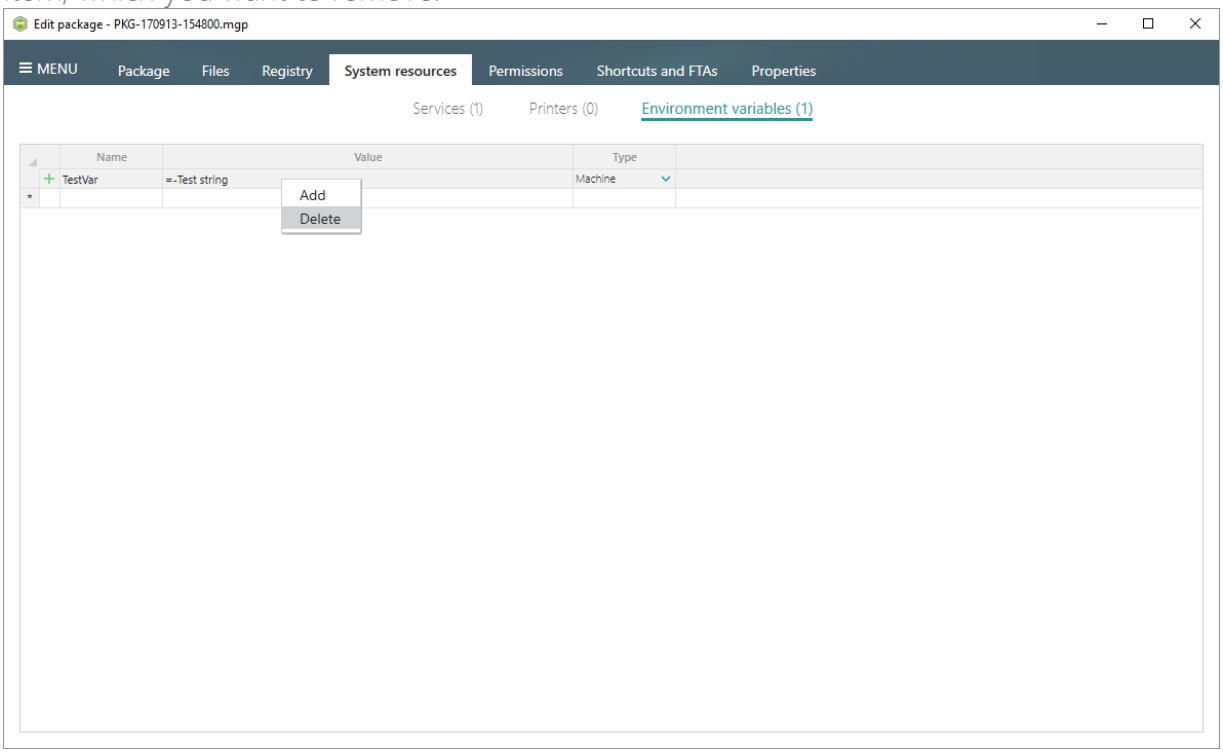



- In order to review and exclude unnecessary services, go to the **System resources** -> **Services** tab, and uncheck the checkbox, located before the service name in the list, for

those services, which you want to exclude.

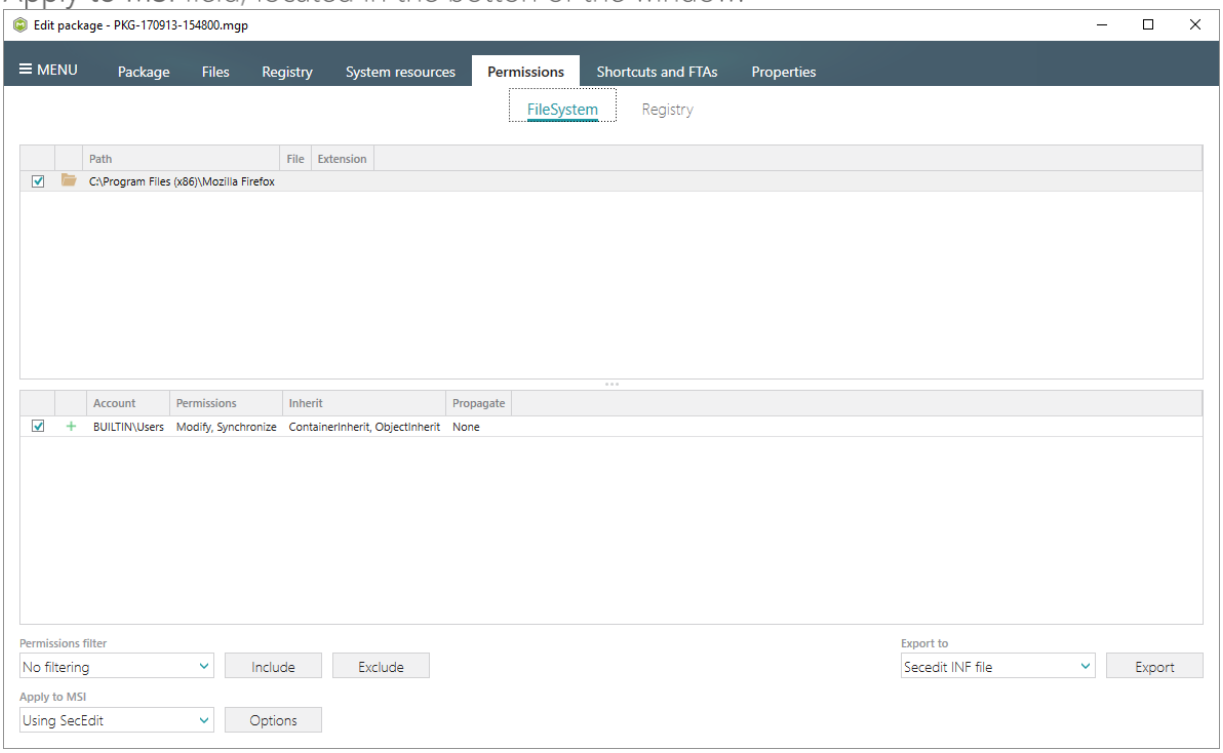



- In order to review and remove unnecessary Environment variables, go to the System resources -> Environment variables tab, and select Delete from the context menu of an item, which you want to remove.

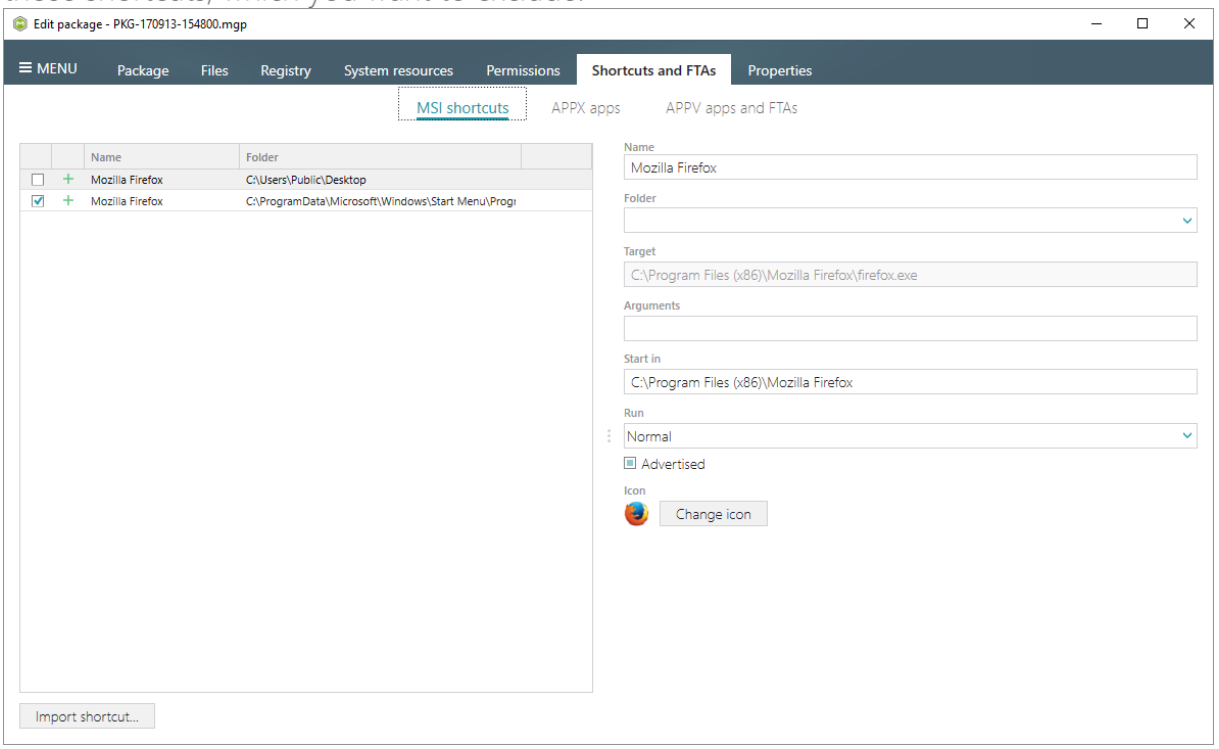



- On the Permissions tab you will find captured permissions for the file system and registry. For excluding unnecessary permission changes, either uncheck the checkbox, located before the path, for which permission changes were detected, or select Do not apply in the

Apply to MSI field, located in the botton of the window.

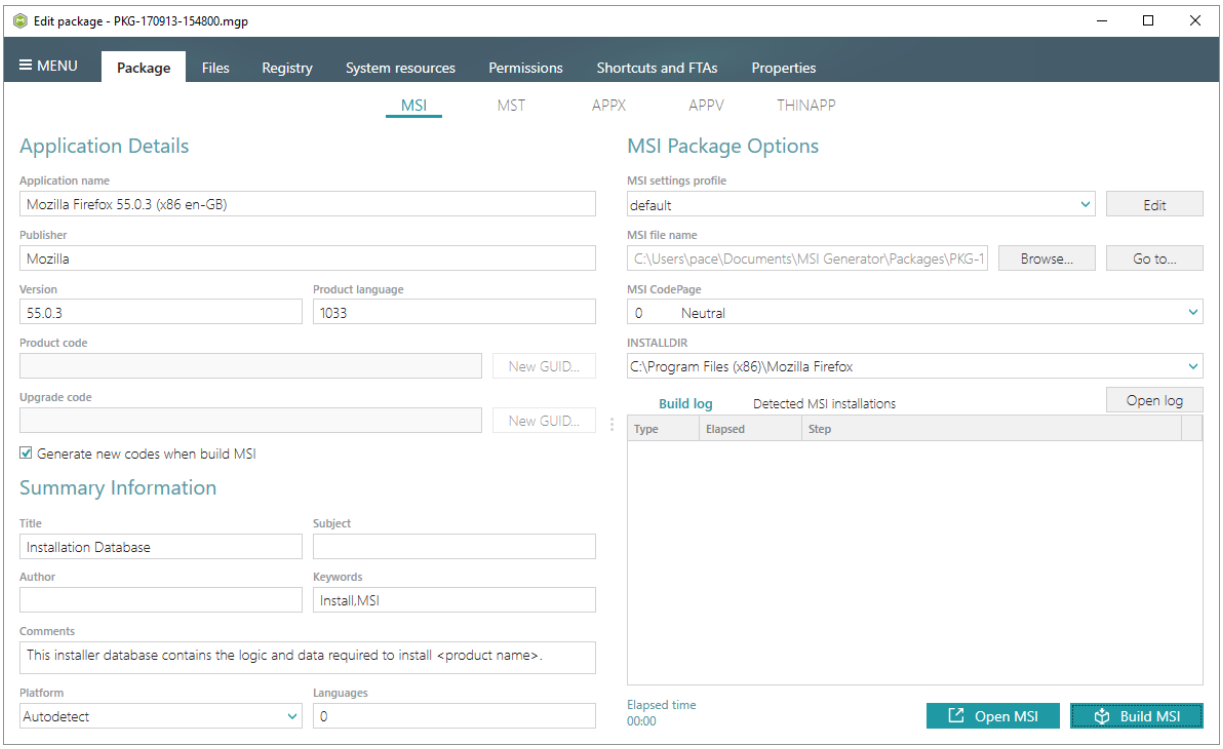



- In order to review and exclude unnecessary shortcuts, go to the Shortcuts and FTAs -> MSI shortcuts tab, and uncheck the checkbox, located before the shortcut name in the list, for those shortcuts, which you want to exclude.

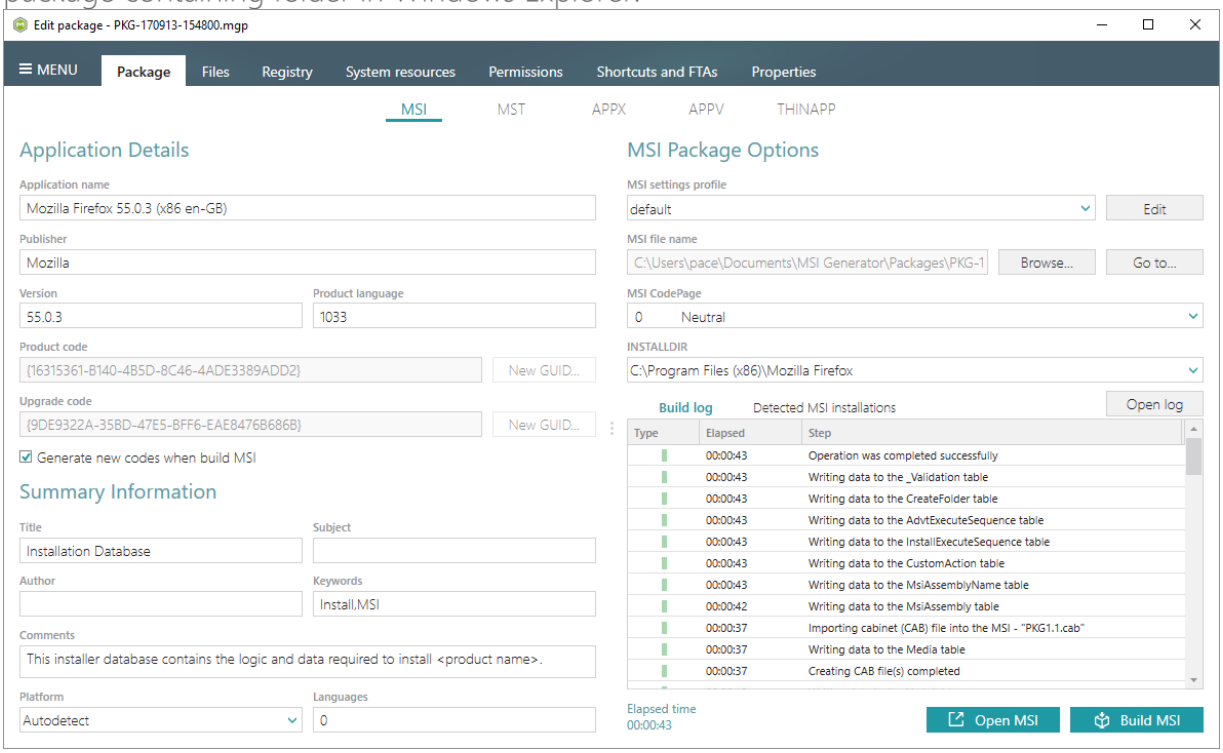



- Finaly, to build an MSI package from the project, navigate to the Package -> MSI tab, update Application details such as name, publisher, version, language, select INSTALLDIR

and click **Build MSI**.



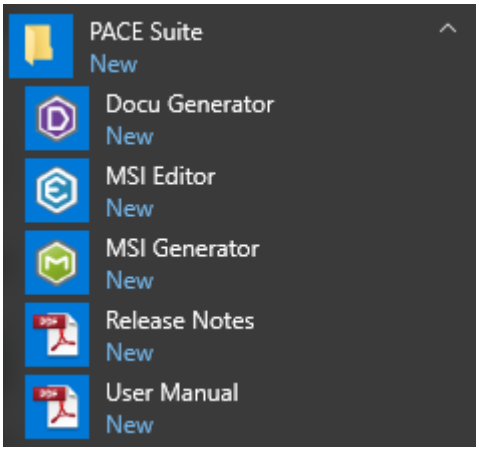

- Once the package is built, click **Go to...**, located next to the **MSI file name** field, to open the package containing folder in Windows Explorer.



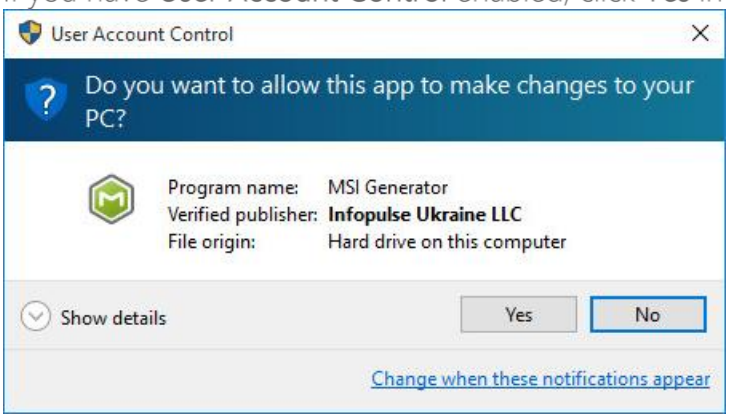
### 3.1.4 Capture System Changes to MSI

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 file system and registry permission changes.

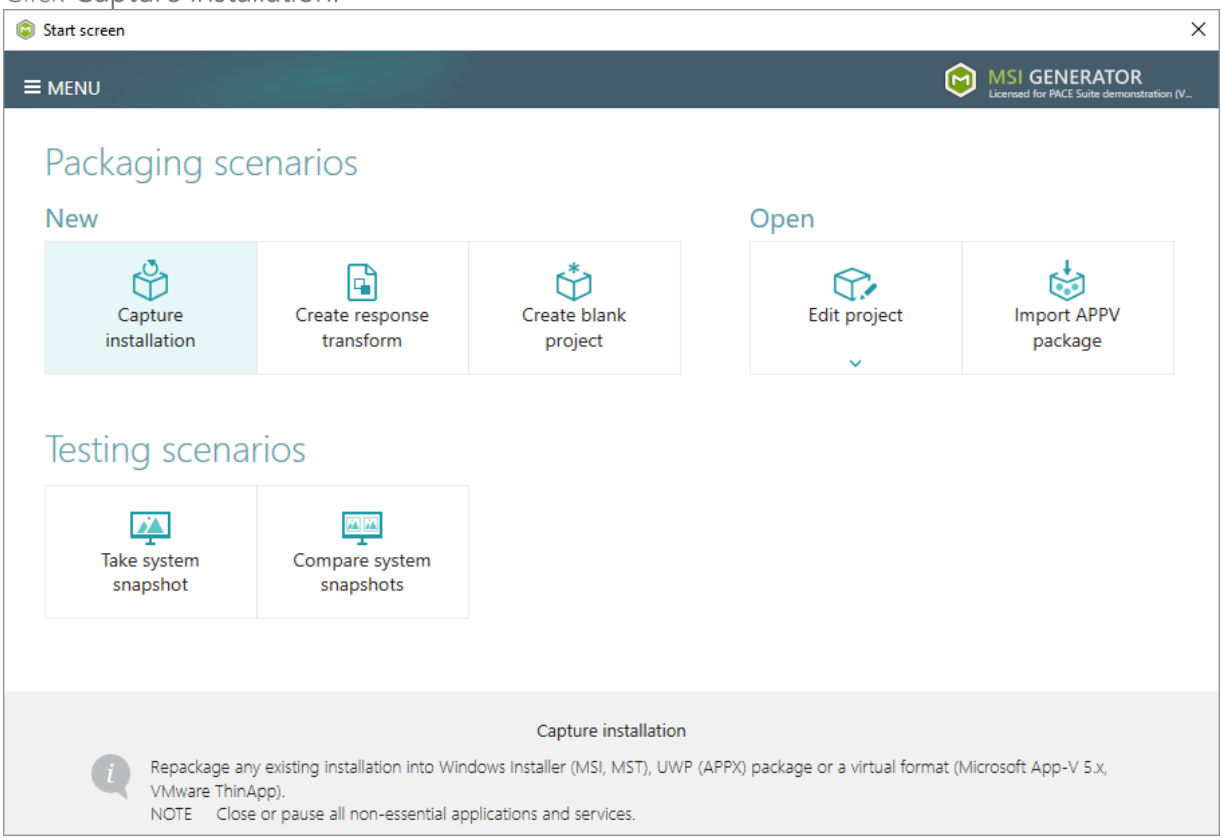
- Launch MSI Generator from the desktop or the start menu shortcut.



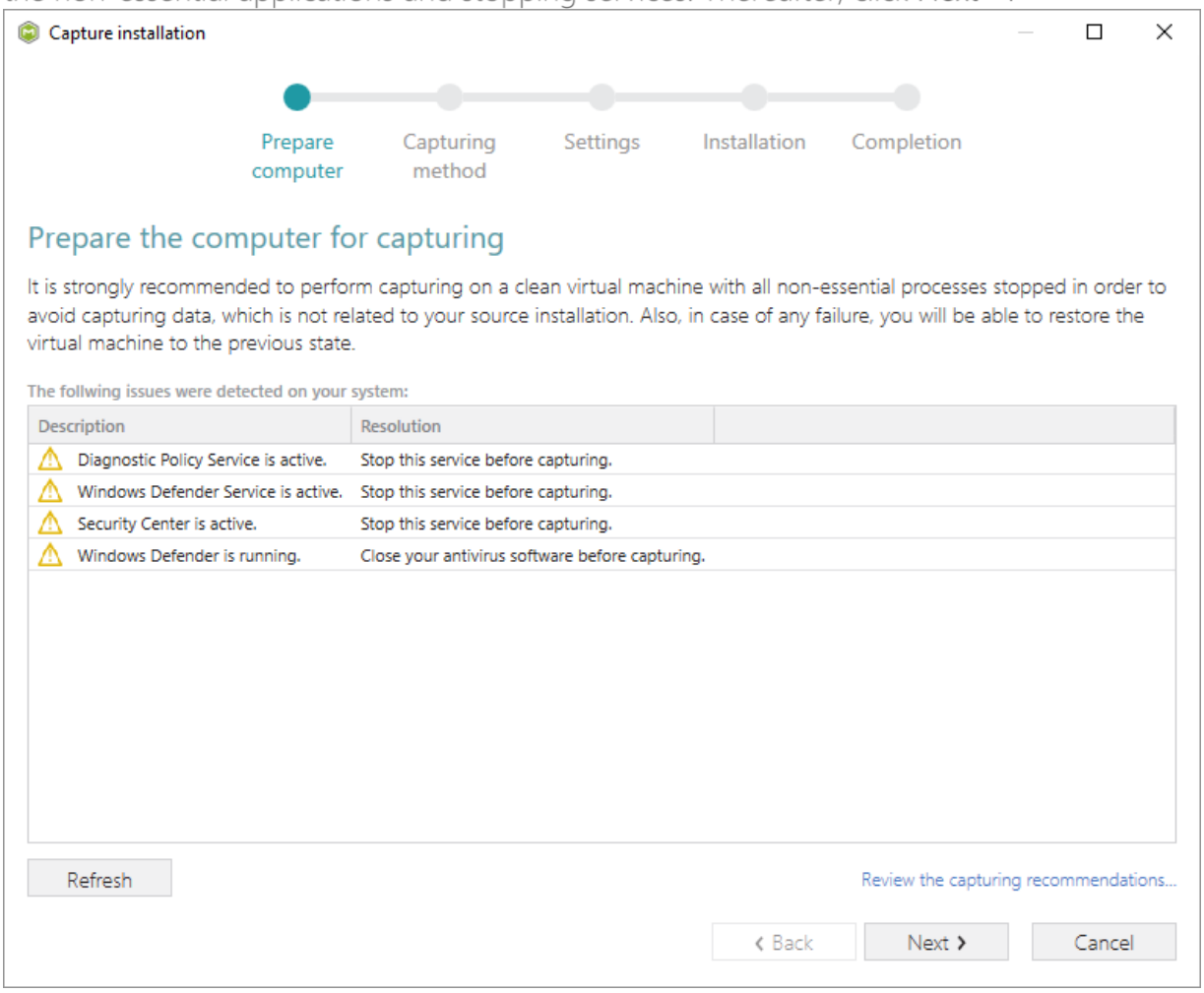

- If you have User Account Control enabled, click Yes in the opened window.



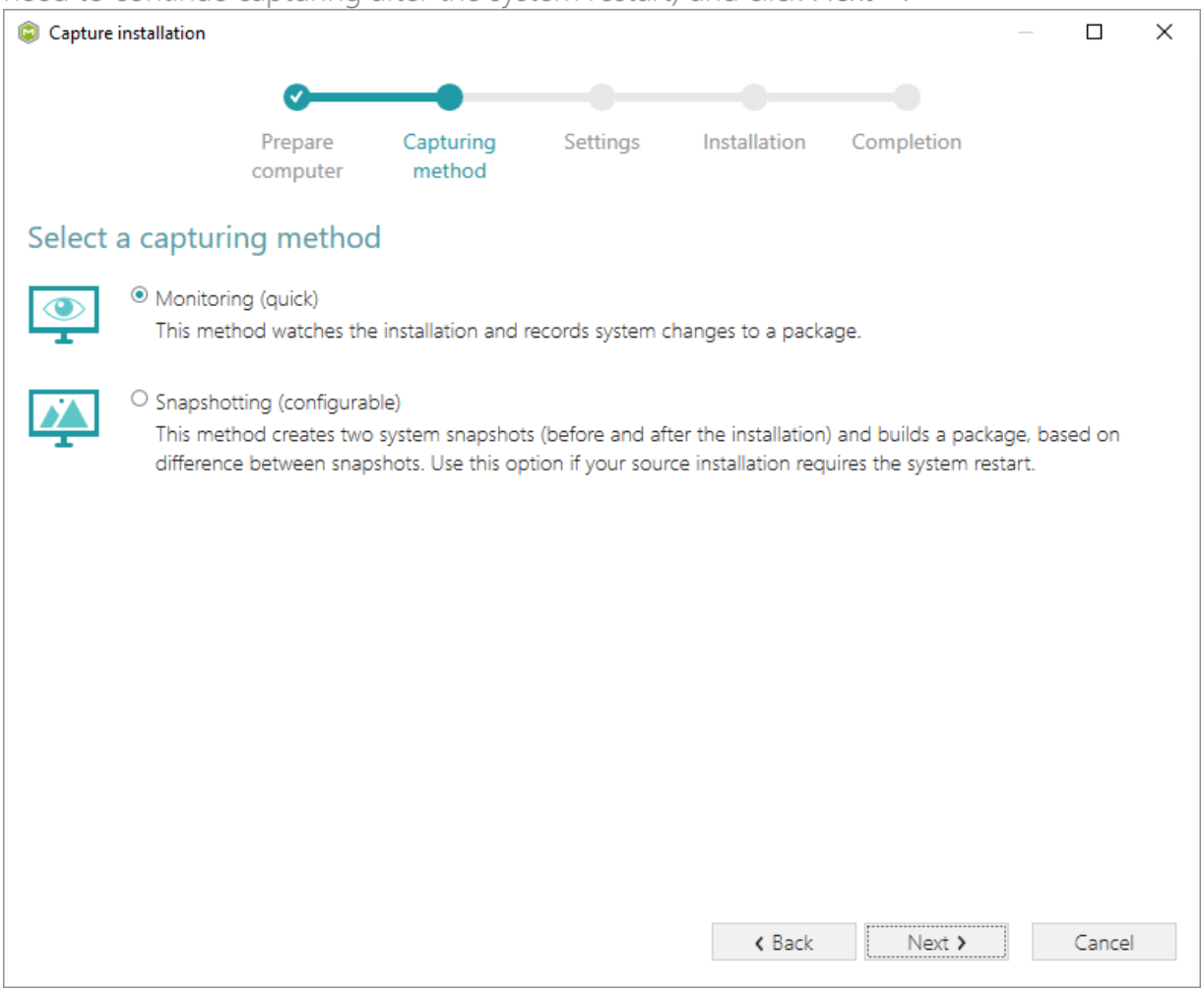
- Click Capture installation.



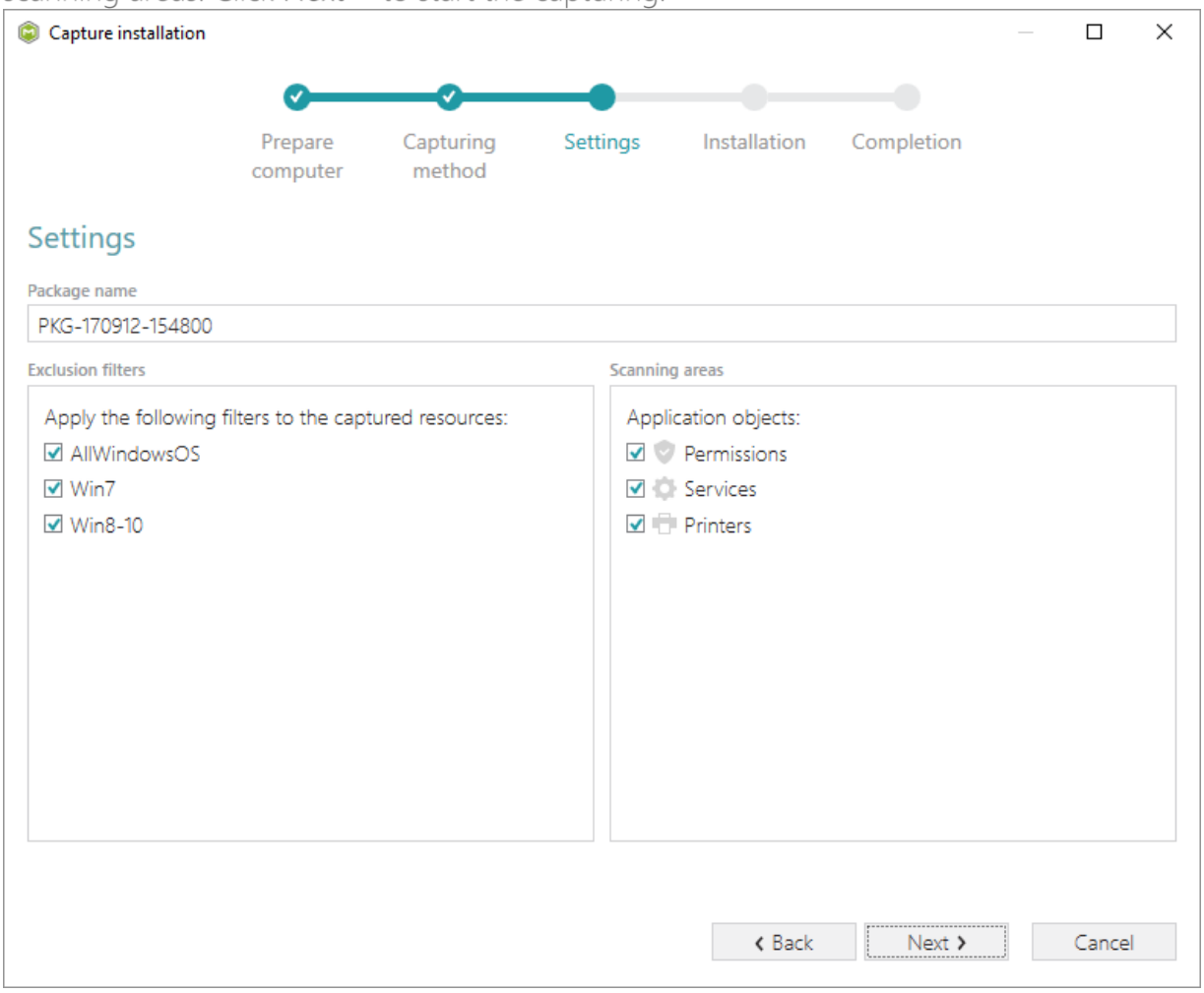
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



- 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 >.

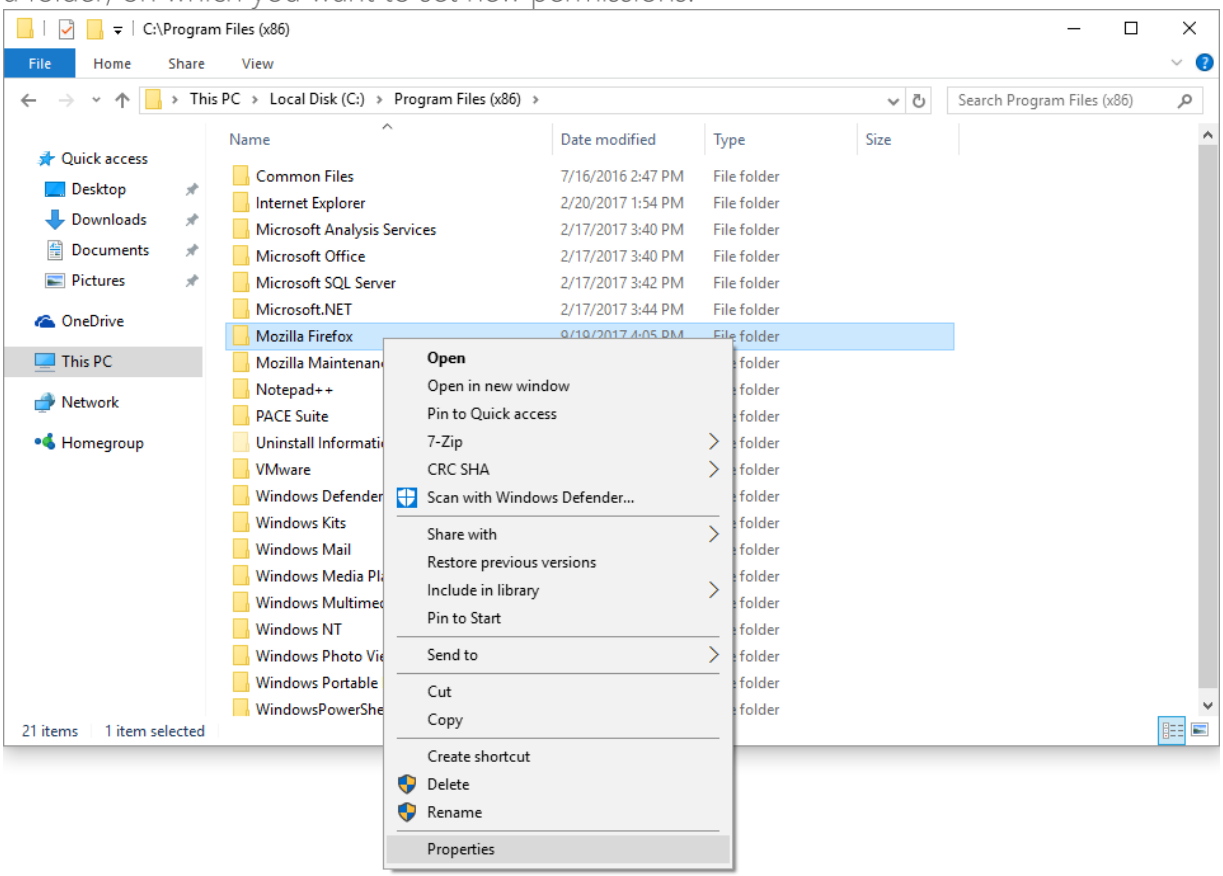


- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.

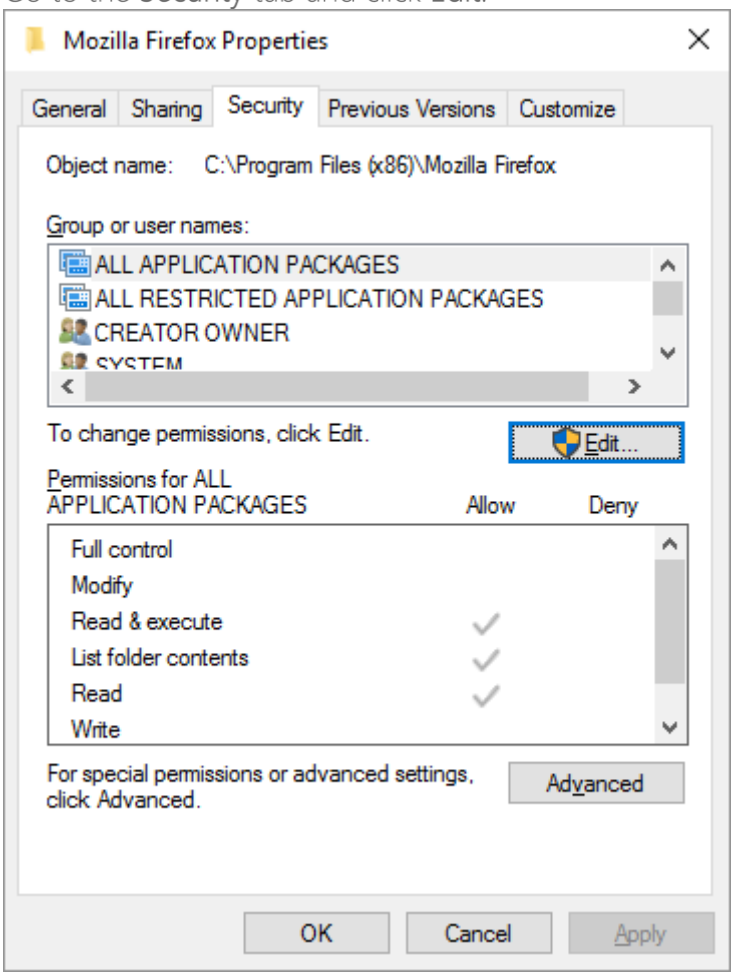



- Now you can make any changes to the file system and registry, which you want to capture and include to the package. For instance, you can create new or copy-paste existing files, import REG file to the system registry, changes permission settings, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.
- If you additionally need to change the default permission settings on a folder or a file and include them to the package, select Properties from the Windows Explorer context menu of

a folder, on which you want to set new permissions.

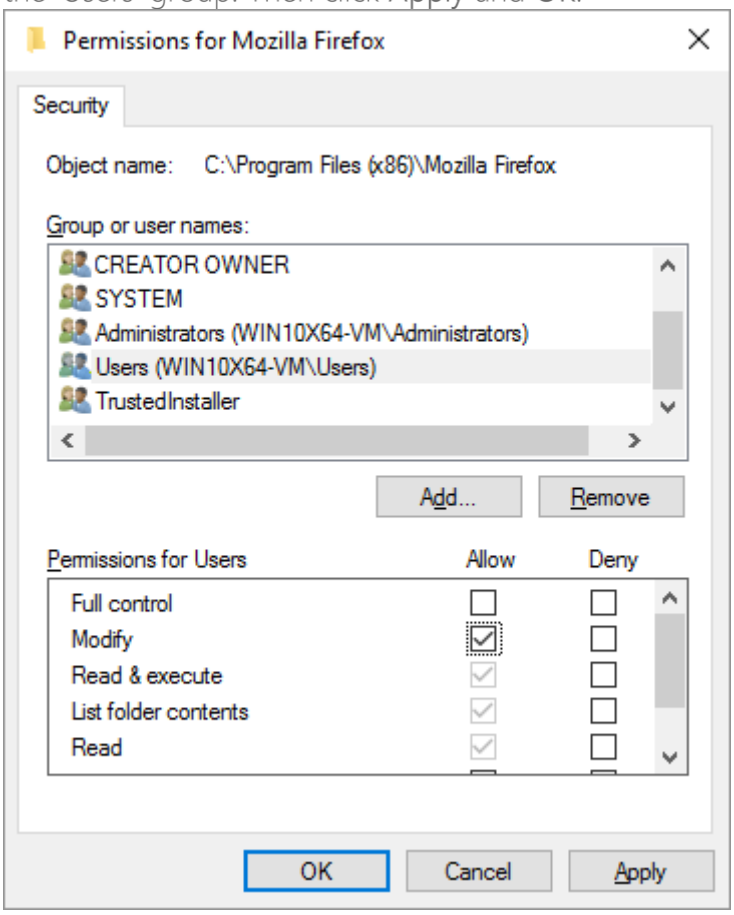


- Go to the Security tab and click Edit.

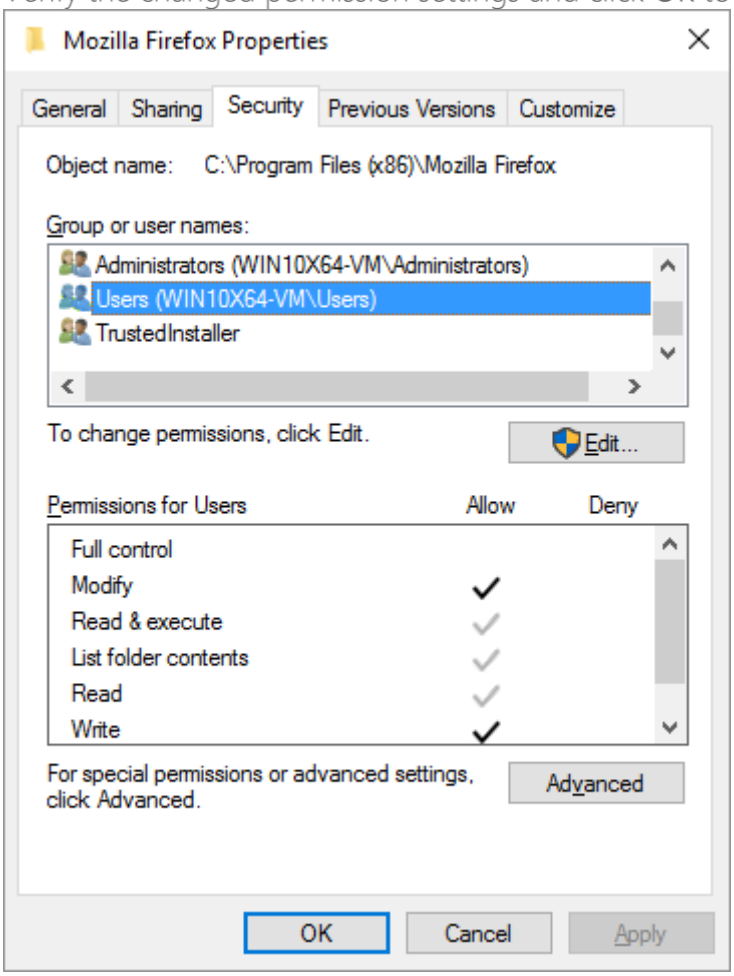



- Select a group or a user, for which you want to change the permissions and then select the necessary access rights. For example, lets set the 'Write' and the 'Modify' access rights for

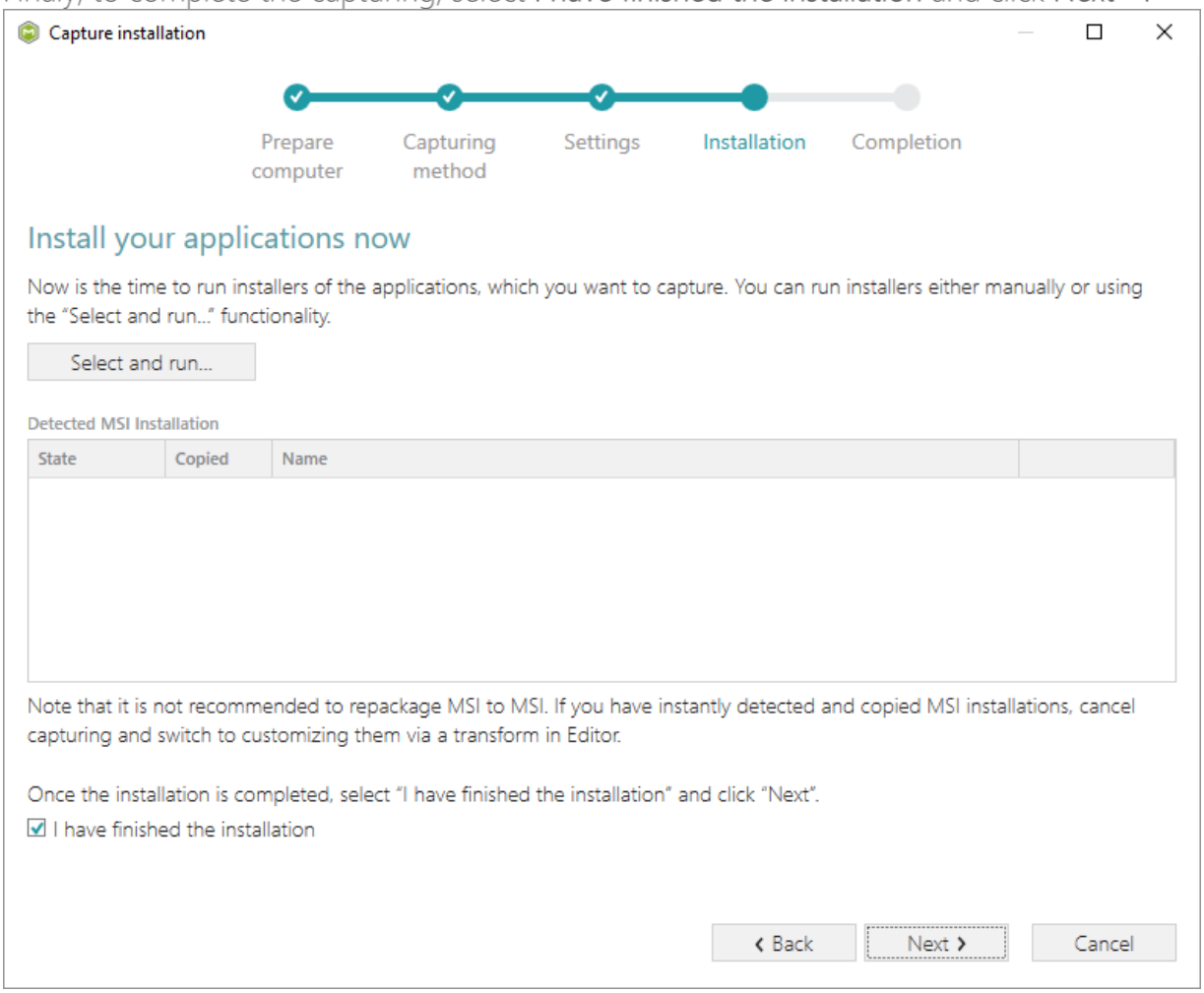
the 'Users' group. Then click Apply and OK.



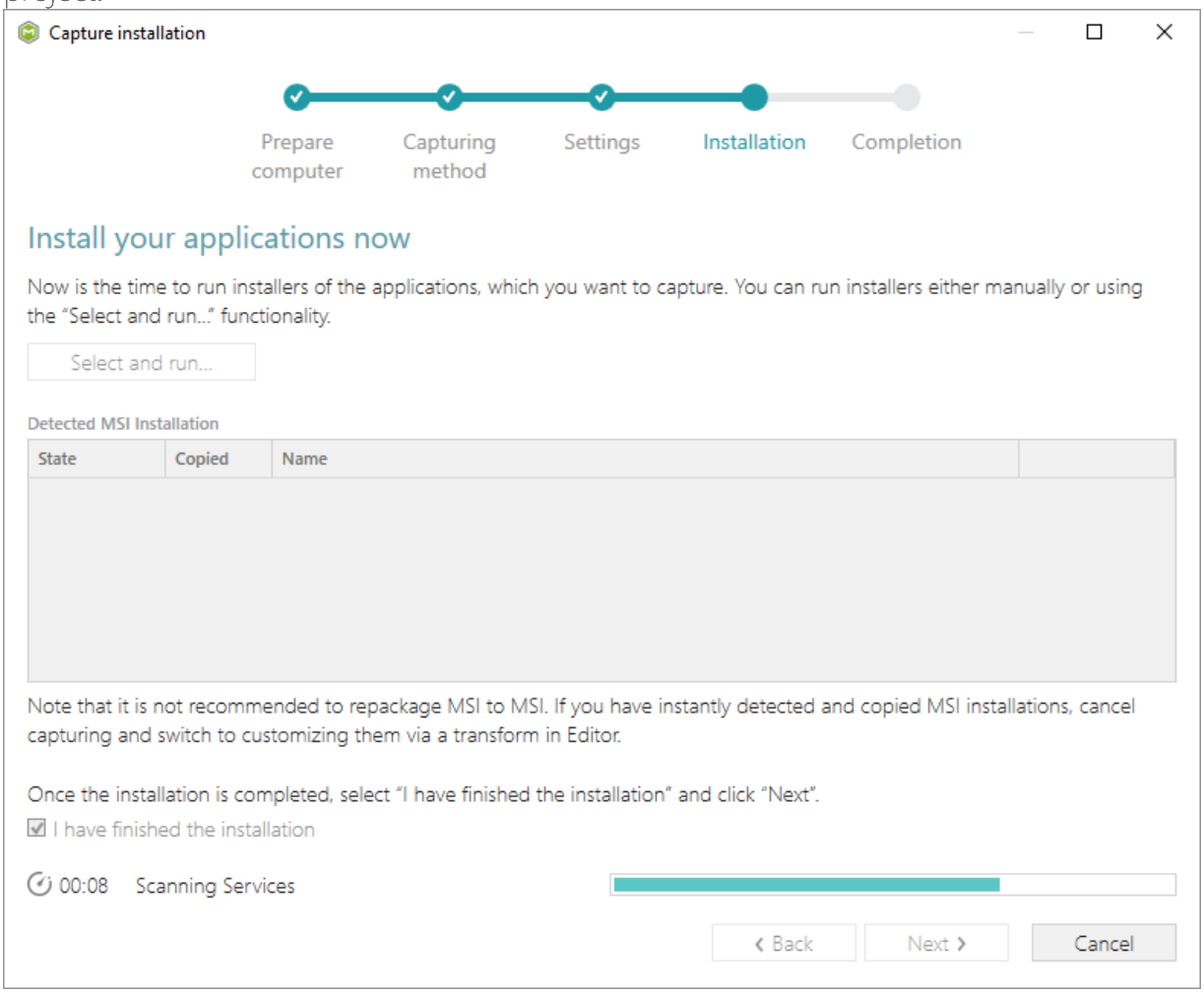
- Verify the changed permission settings and click OK to close the window.



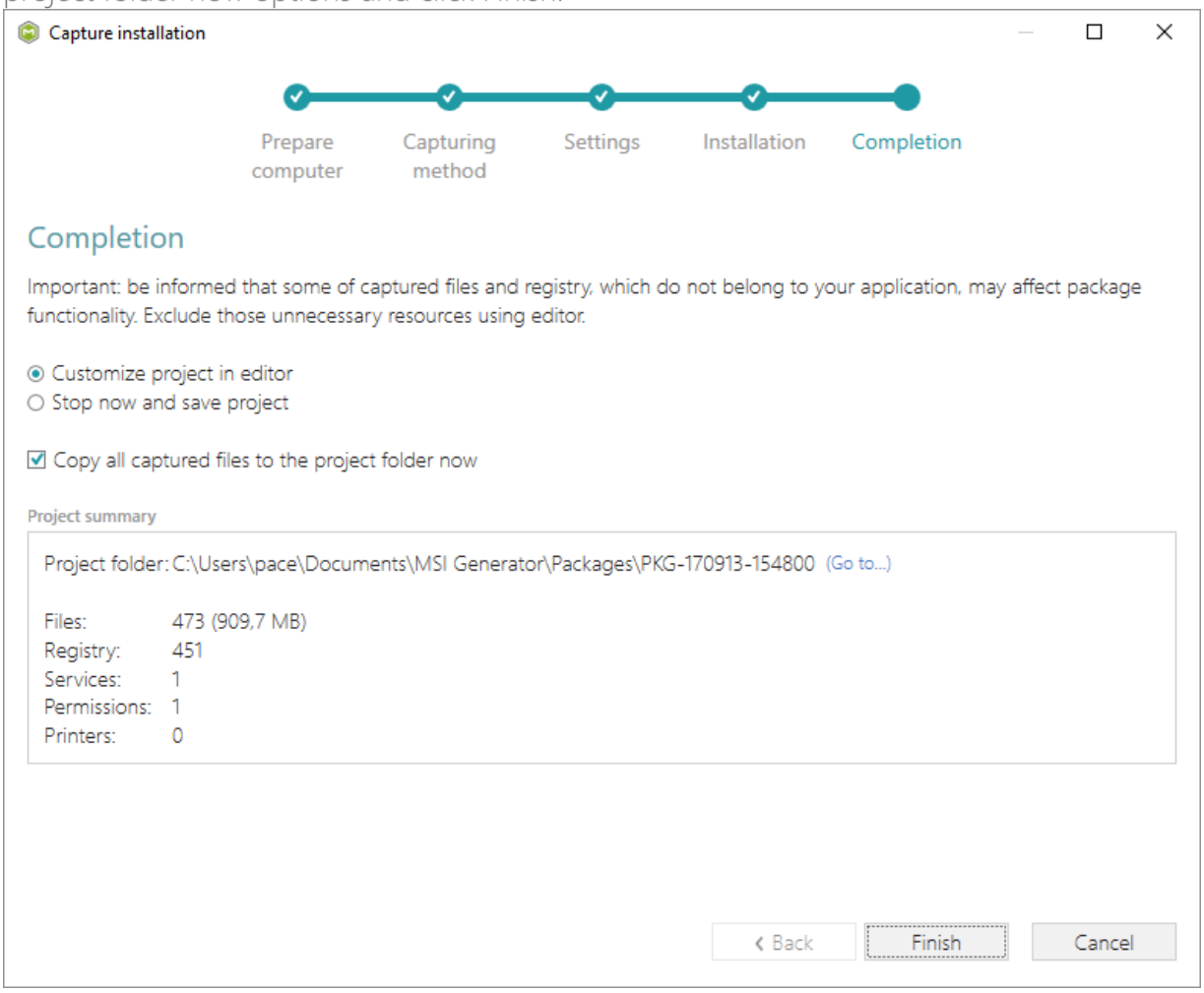
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



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

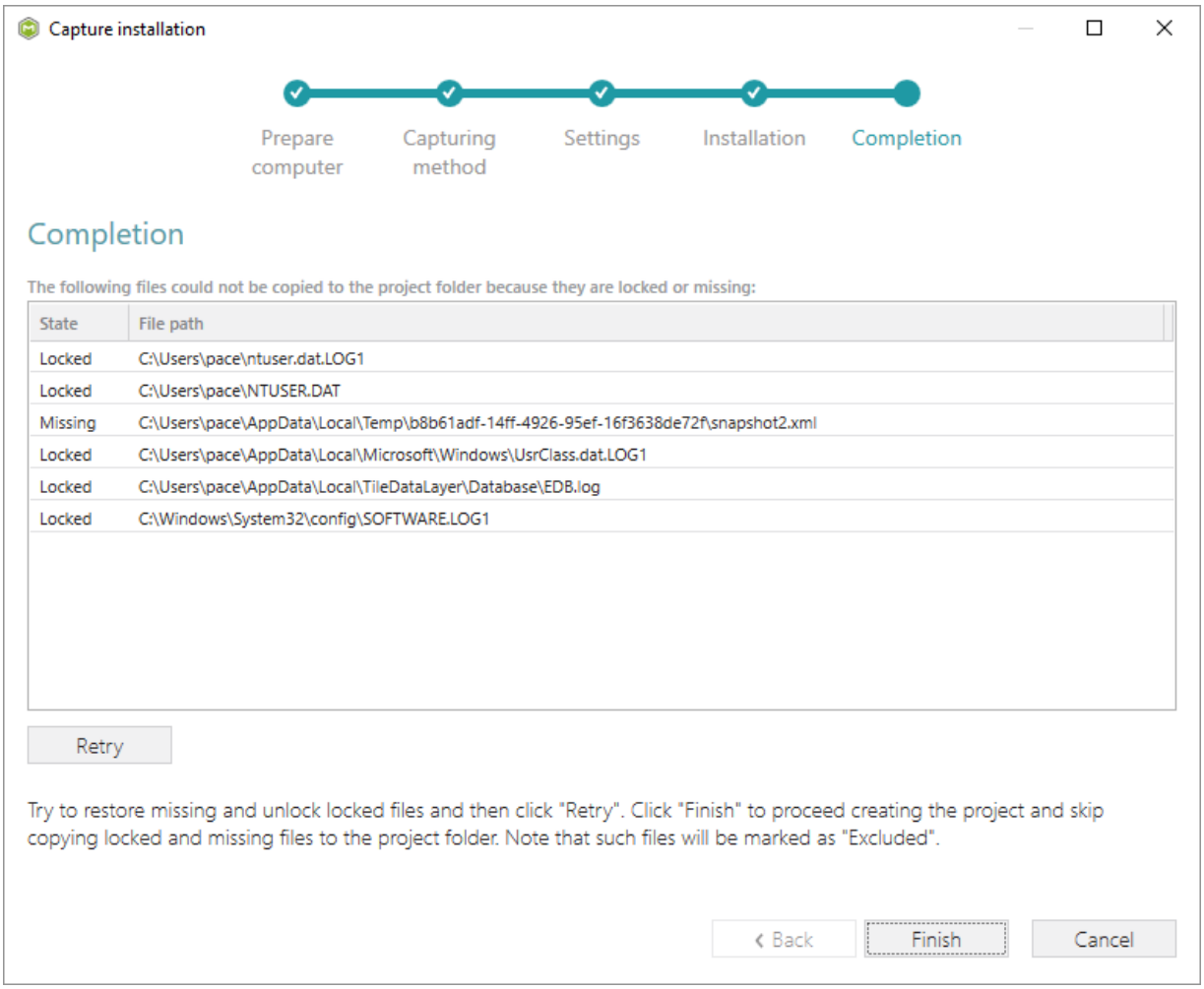


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



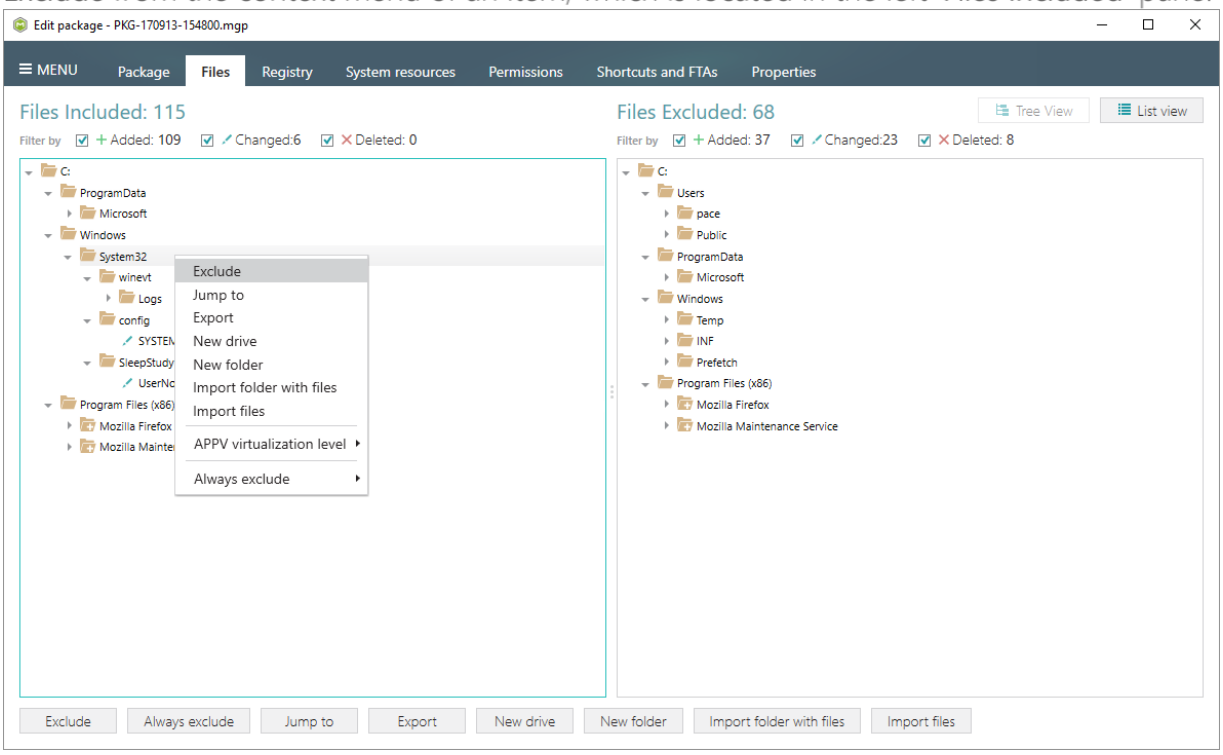

- The following dialog displays captured files, which could not be copied to the project folder because they do not exist anymore or locked by the system or by an application. Try to resolve these issues and then click Retry. Click Finish to skip copying the missing and locked

files and mark them as Excluded resources.

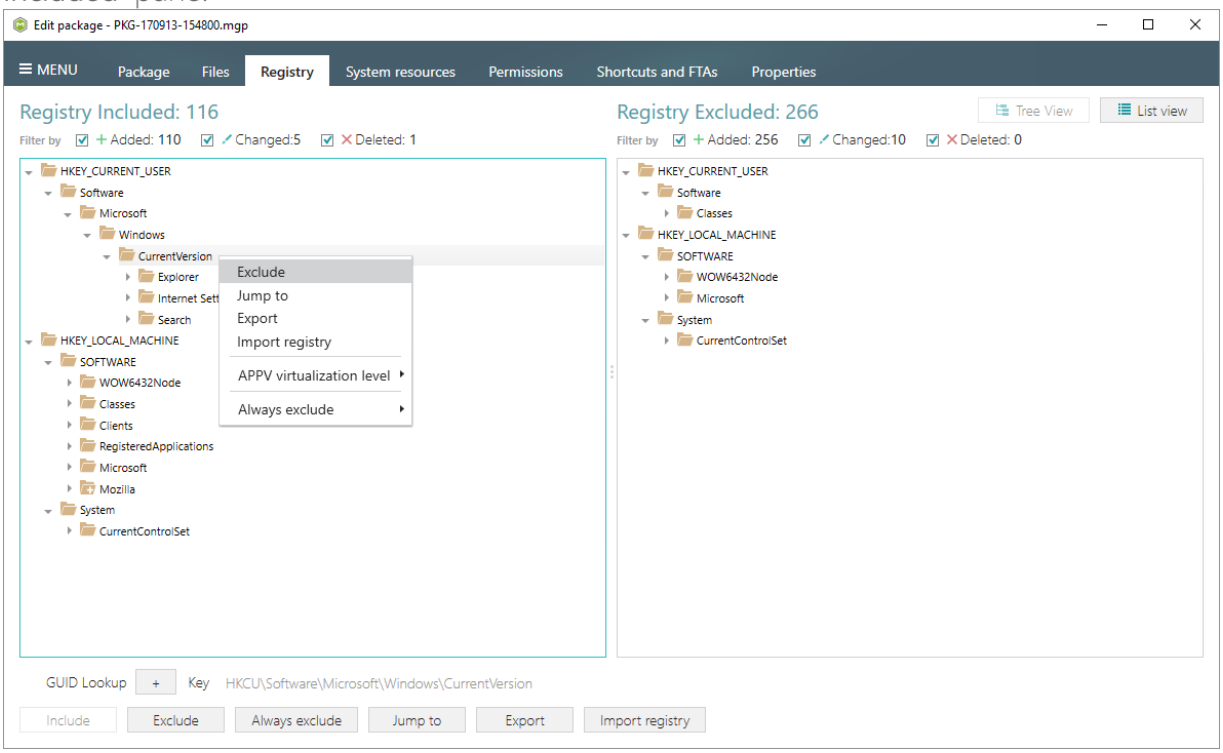



- Once the project is opened in the project editor, review the captured resources at the Files, Registry, System resources, Permissions, and Shortcuts and FTAs tabs and exclude unnecessary ones from the project. Unnecessary resources are files, registry entries, which are usually created or modified in the result of operating system work, and such resources could not be a part of your captured application. Unfortunately, there is no universal rule to discover which of captured files or registry entries should be excluded, so exclude only those ones, which almost 100% do not refer to your captured application (e.g. NOD32 antivirus files couldn't be a part of Firefox application).

- 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.

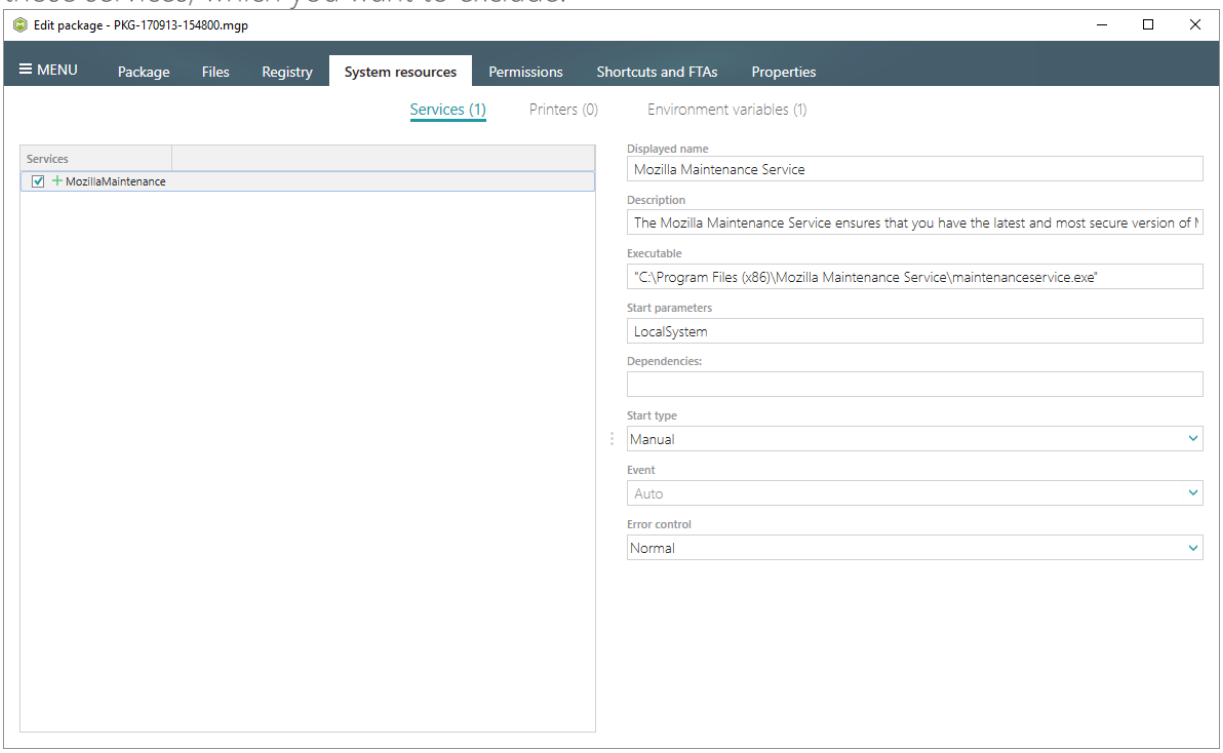



- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.

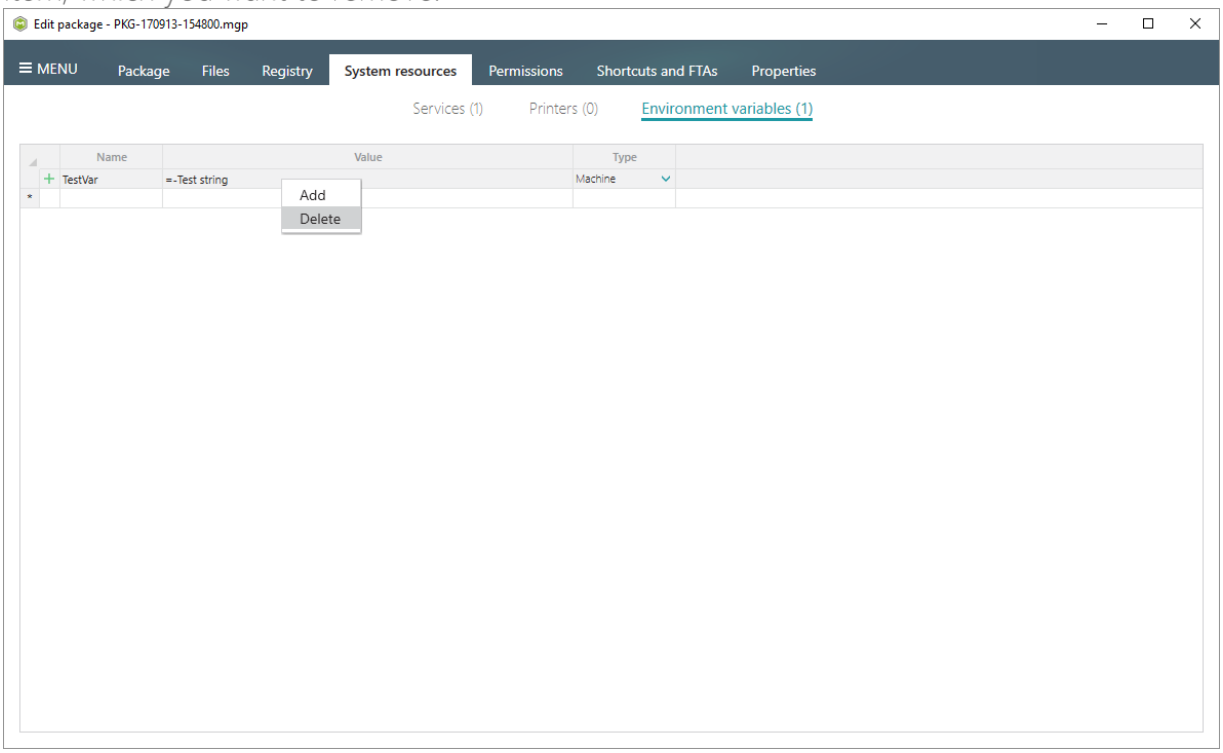



- In order to review and exclude unnecessary services, go to the **System resources** -> **Services** tab, and uncheck the checkbox, located before the service name in the list, for

those services, which you want to exclude.

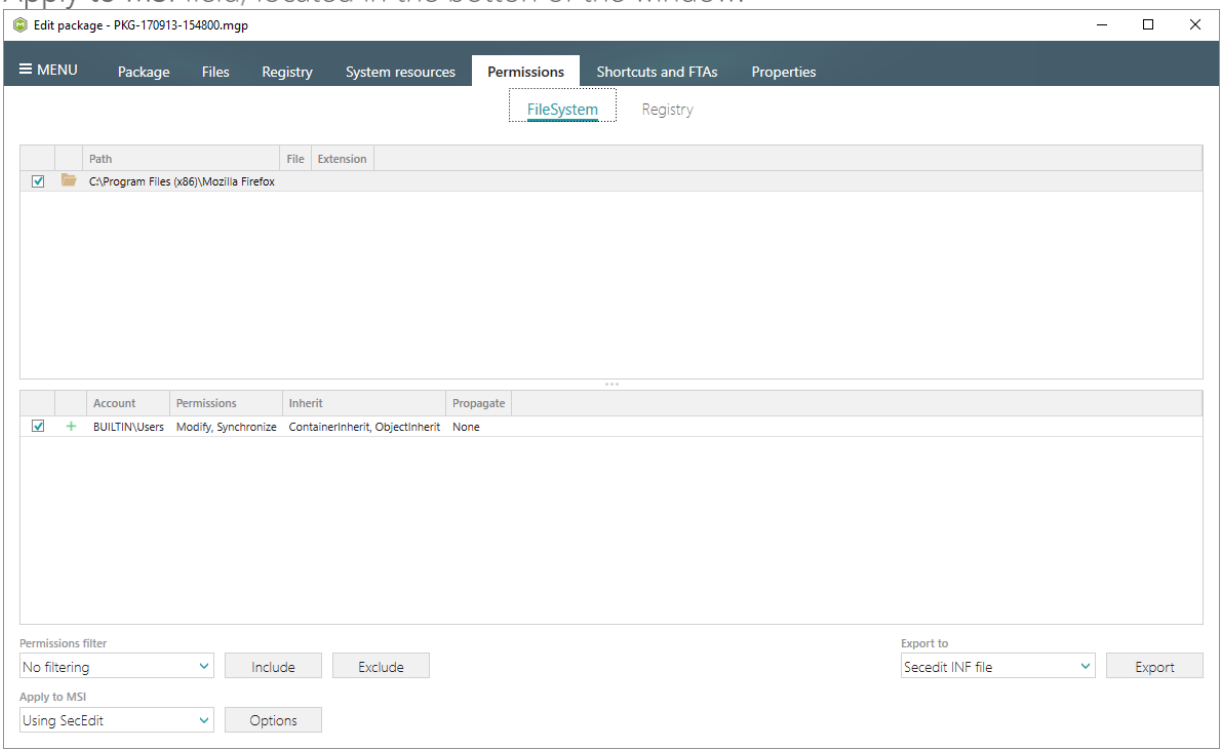



- In order to review and remove unnecessary Environment variables, go to the System resources -> Environment variables tab, and select Delete from the context menu of an item, which you want to remove.

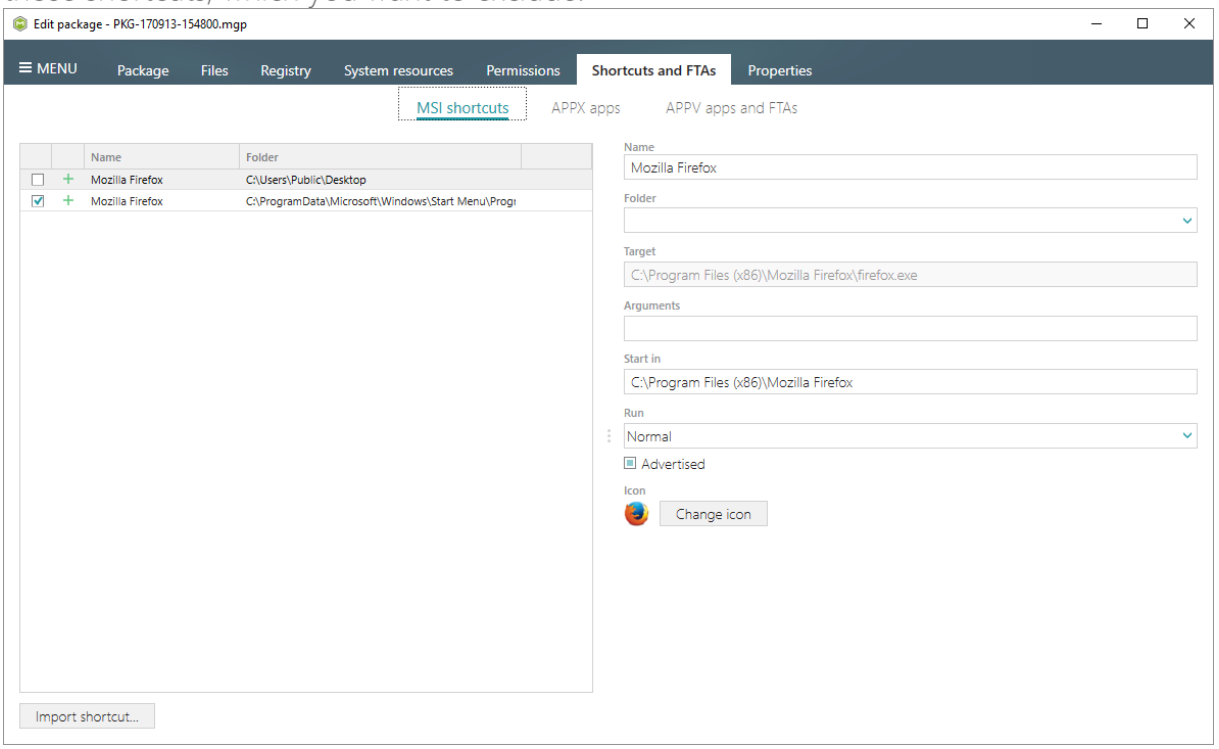



- On the Permissions tab you will find captured permissions for the file system and registry. For excluding unnecessary permission changes, either uncheck the checkbox, located before the path, for which permission changes were detected, or select Do not apply in the

Apply to MSI field, located in the botton of the window.

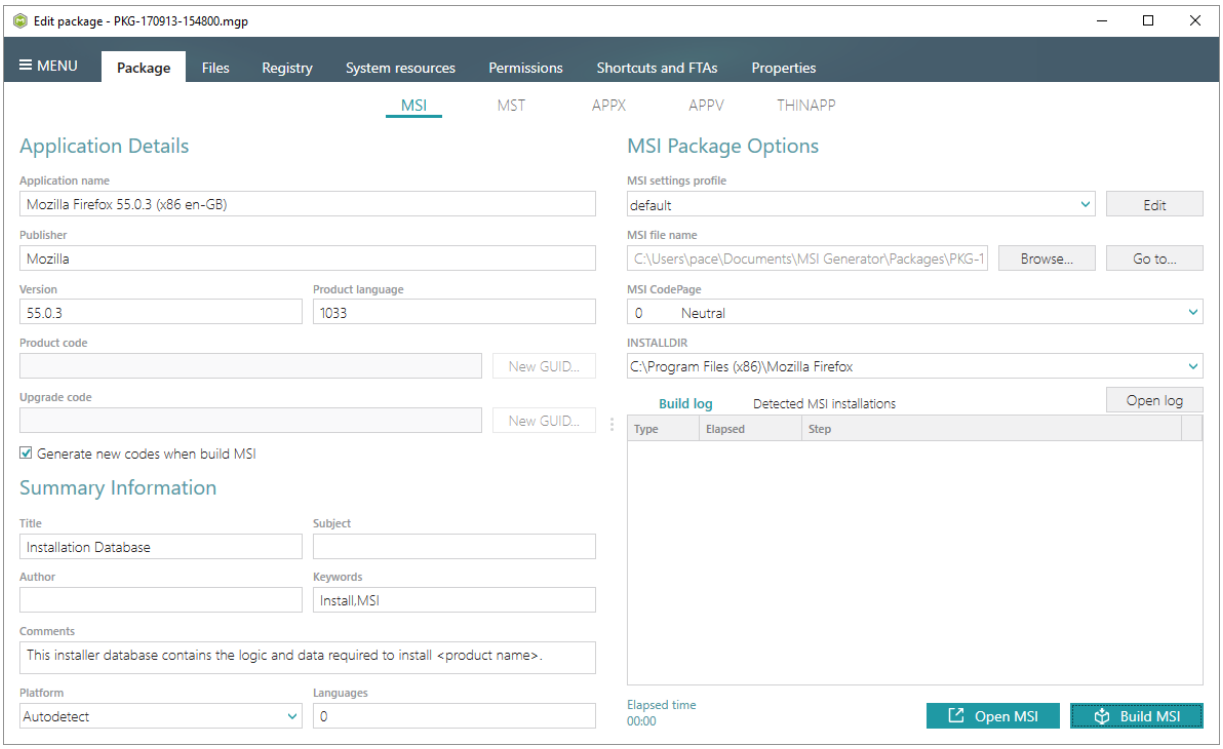



- In order to review and exclude unnecessary shortcuts, go to the Shortcuts and FTAs -> MSI shortcuts tab, and uncheck the checkbox, located before the shortcut name in the list, for those shortcuts, which you want to exclude.

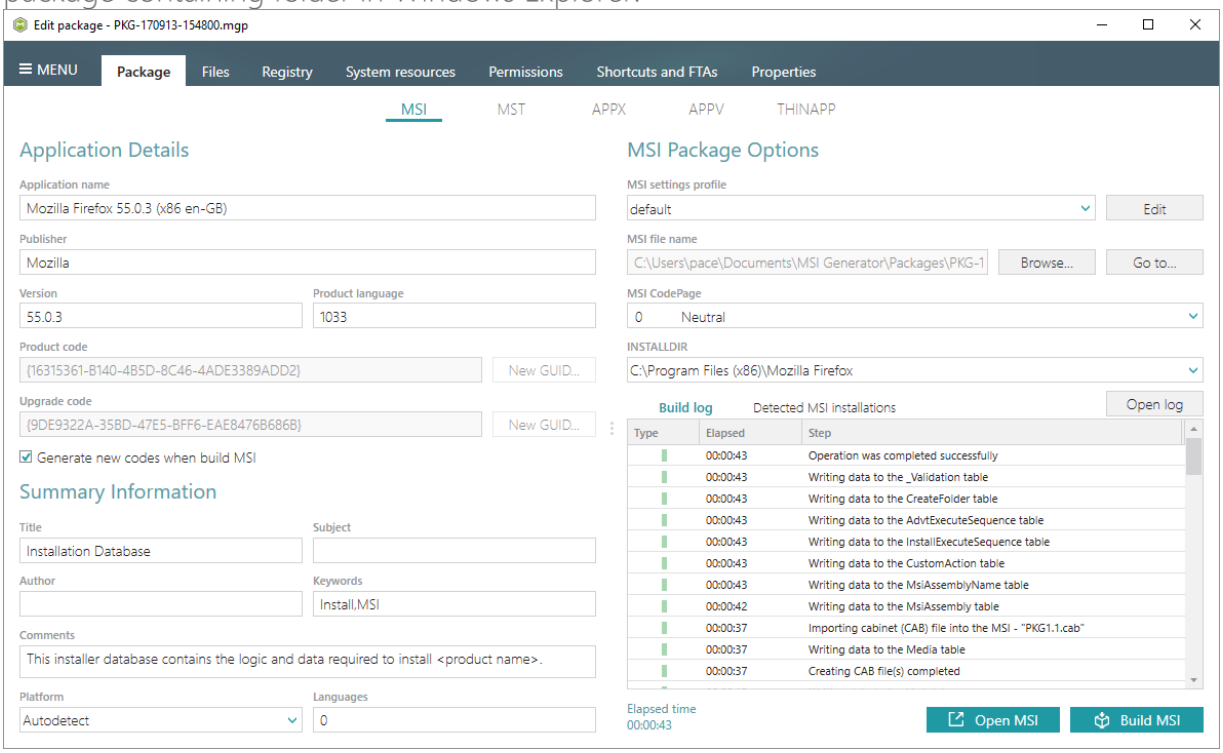



- Finaly, to build an MSI package from the project, navigate to the Package -> MSI tab, update Application details such as name, publisher, version, language, select INSTALLDIR

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:

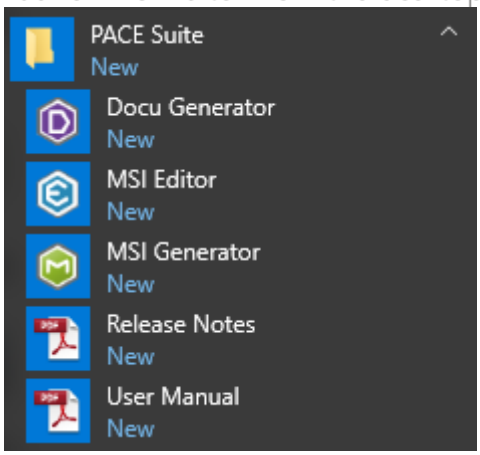
- Create Blank MST, described in section 3.2.1
  Create a new empty transform file (MST) for your MSI package in MSI Editor. All changes, made to the opened MSI database, will be saved to the MST.
- Create Response MST, described in section 3.2.2
  Capture all your inputs and changes, made to the Windows 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 file system and registry permission changes.

### 3.2.1 Create Blank MST

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

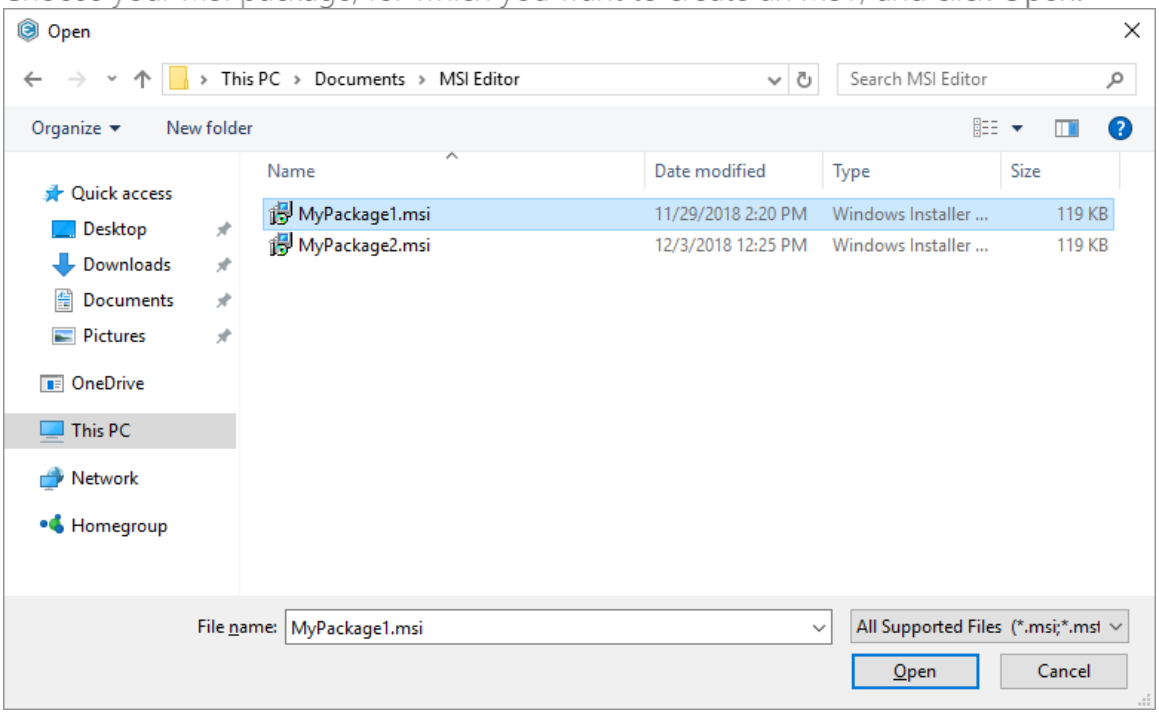
NOTE PACE Suite supports two scenarious of saving changes to MST:
1. Open MSI -> create MST -> change MSI resources -> save (described below)
2. Open MSI -> change MSI resources -> save changes as MST

[1]. Launch MSI Editor from the desktop or the start menu shortcut.

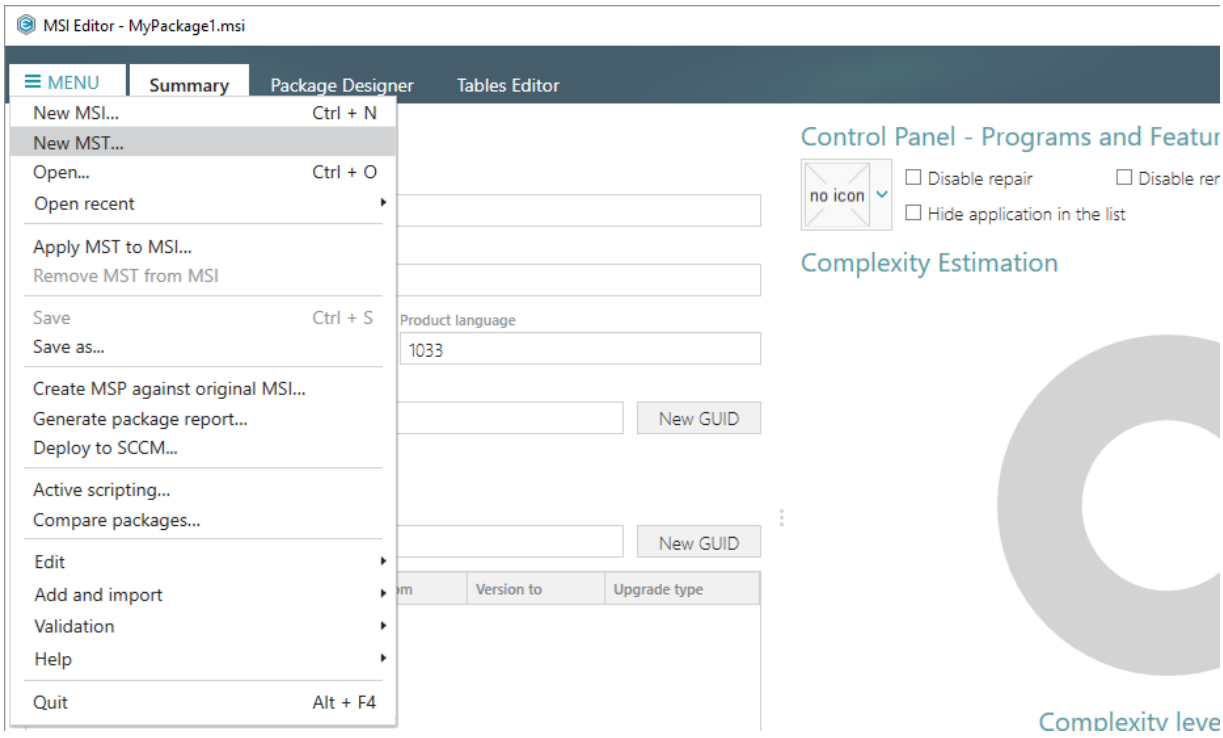


[2]. Select Open... from MENU.

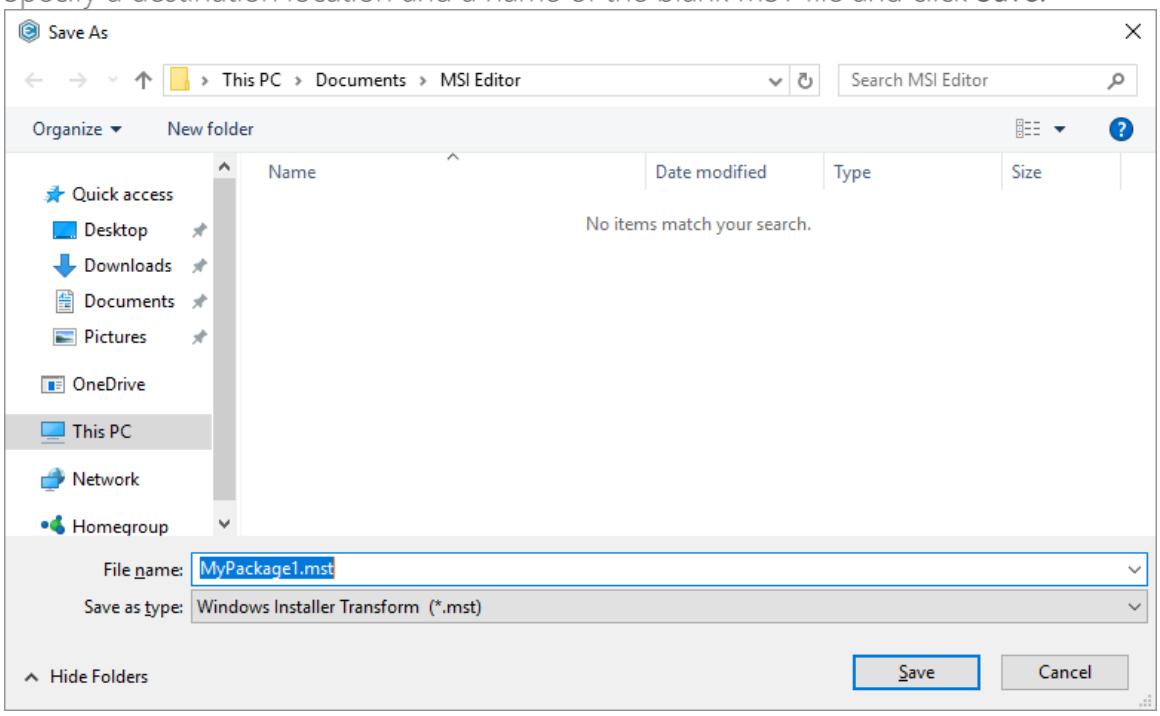
[3]. Choose your MSI package, for which you want to create an MST, and click Open.



[4]. Select New MST... from MENU.

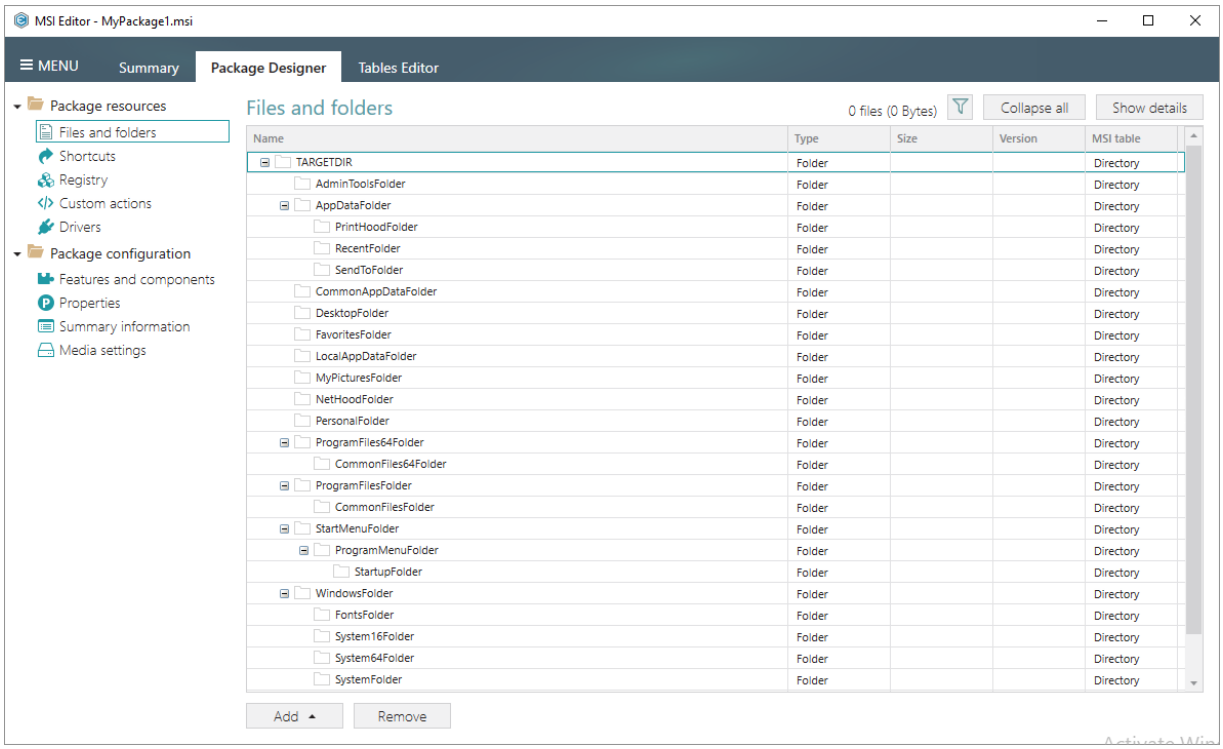



[5]. Specify a destination location and a name of the blank MST file and click Save.

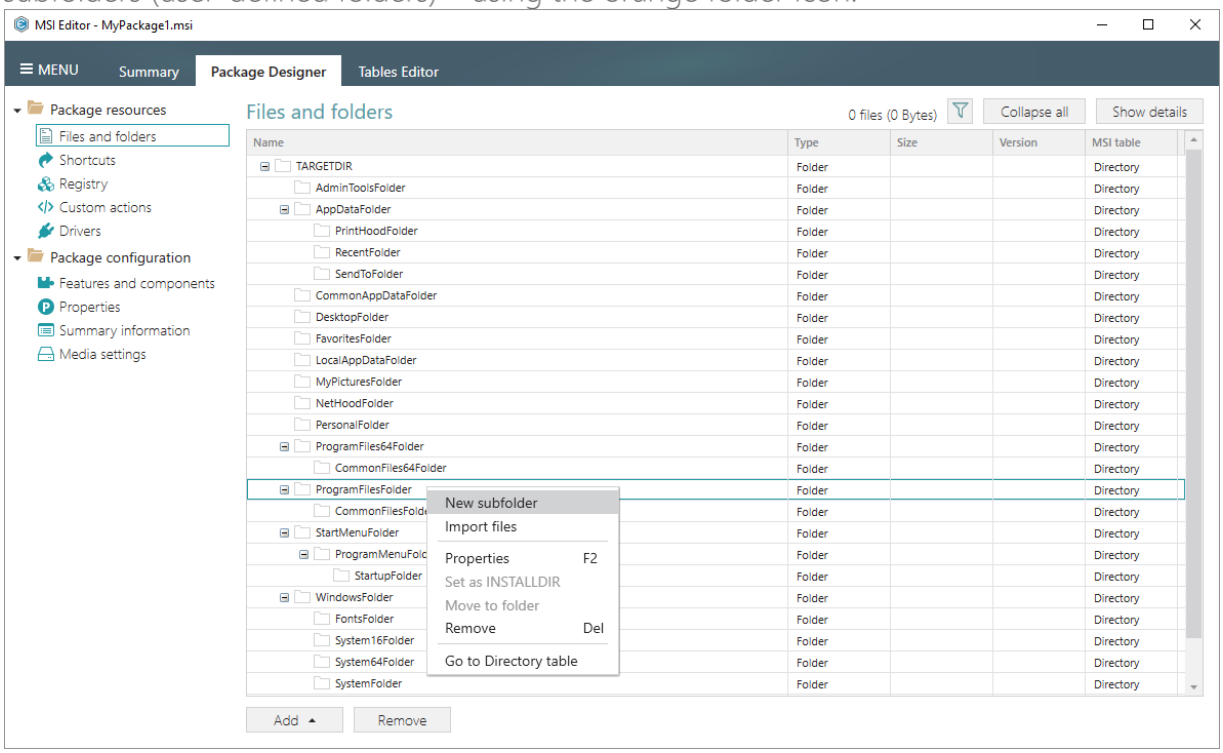



[6]. Now you can add and edit resources of the opened MSI package via the MST. The steps from [7] to [16] describe how to create a new folder and import files to this folder; the steps from [17] to [25] describe how to create a new registry key, value, and import registry entries from the REG files; the steps from [26] to [34] describe how to create a new shortcut and import existing shortcuts from the file system to the package; and the last step [35] describes how to save all these changes to MST.

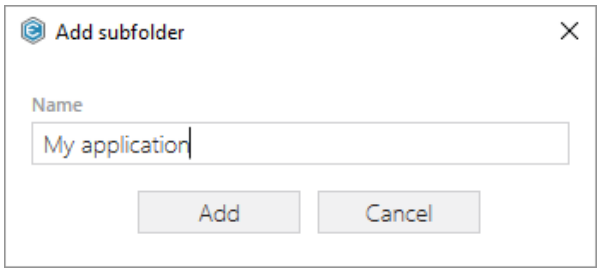
[7]. In order to create a new folder and import files to this folder, go to the **Package Designer –> Files and folders** tab.



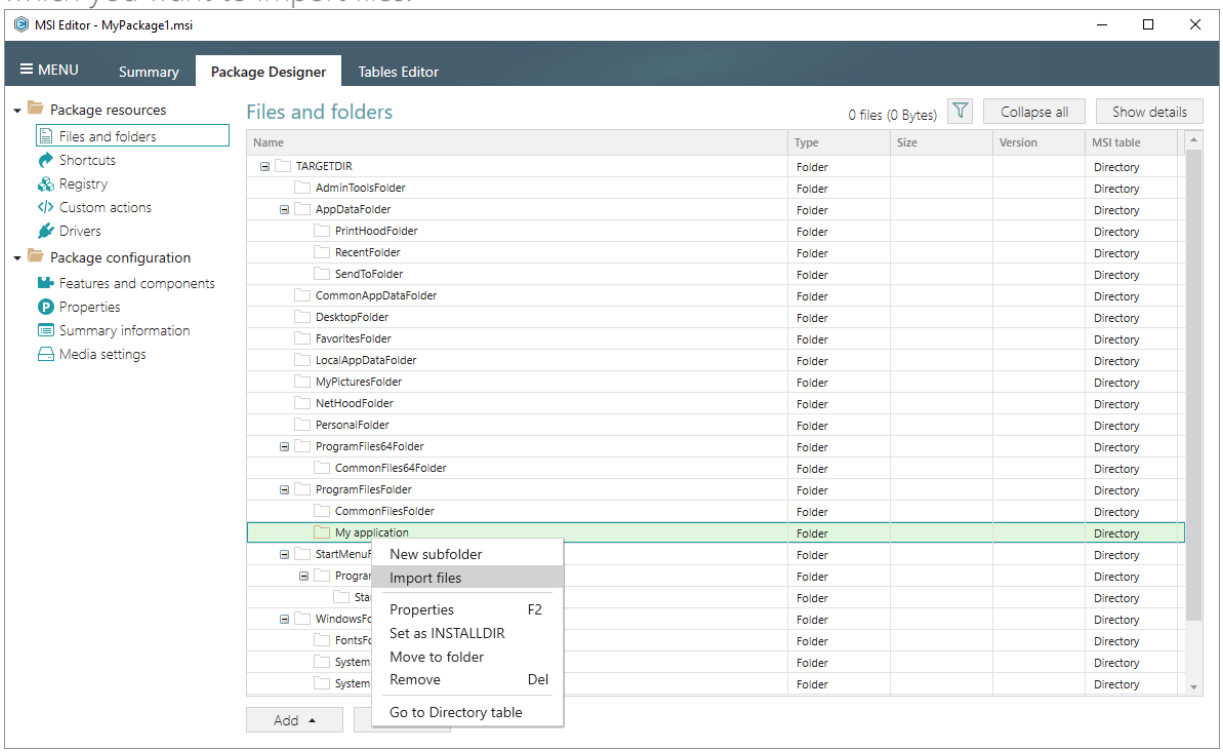

[8]. Select **New subfolder** from the context menu of a folder, where you want to create a new subfolder. Note that system folders are displayed using the grey folder icon and the subfolders (user-defined folders) – using the orange folder icon.



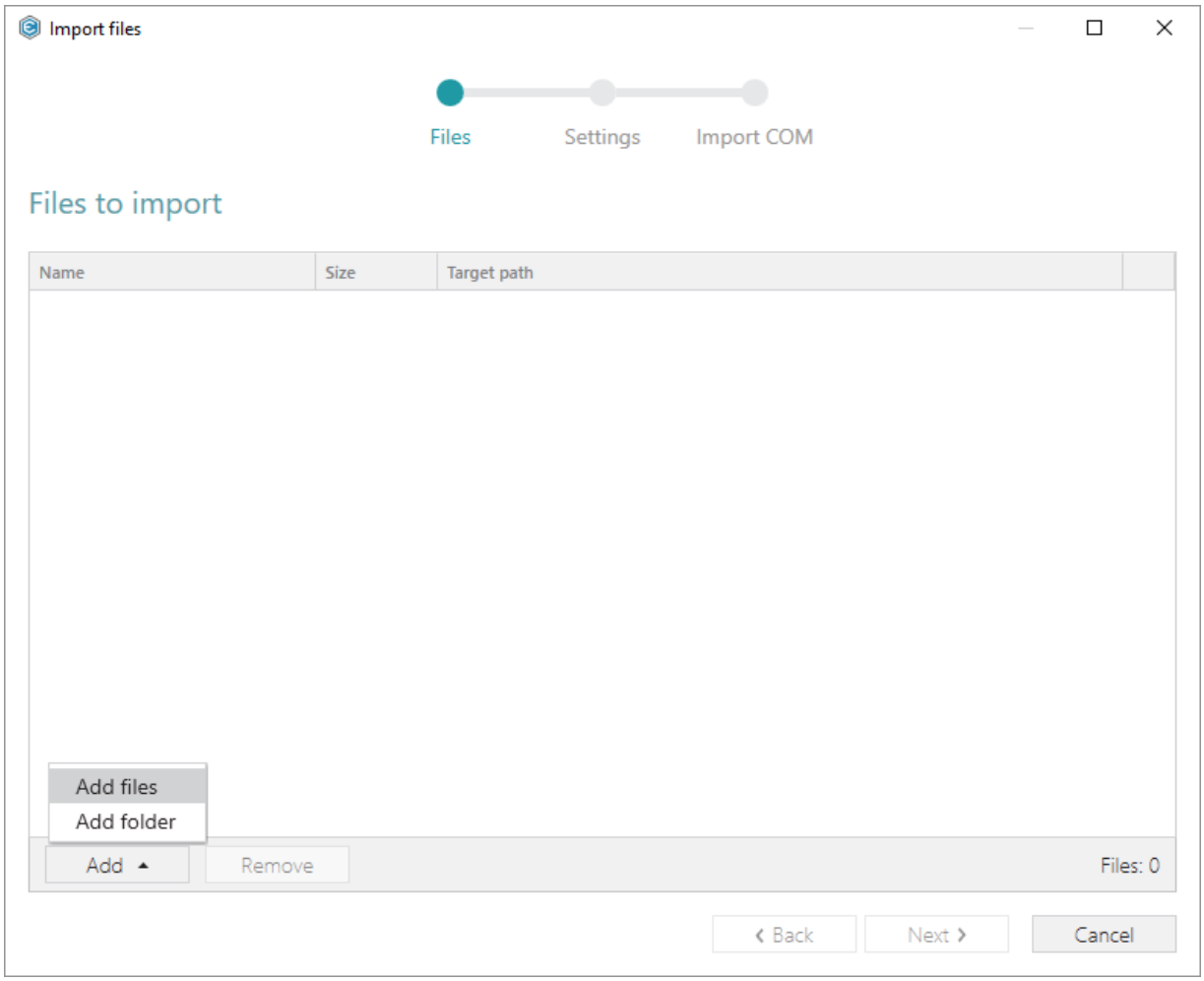
[9]. Enter a folder name and click **Add**.



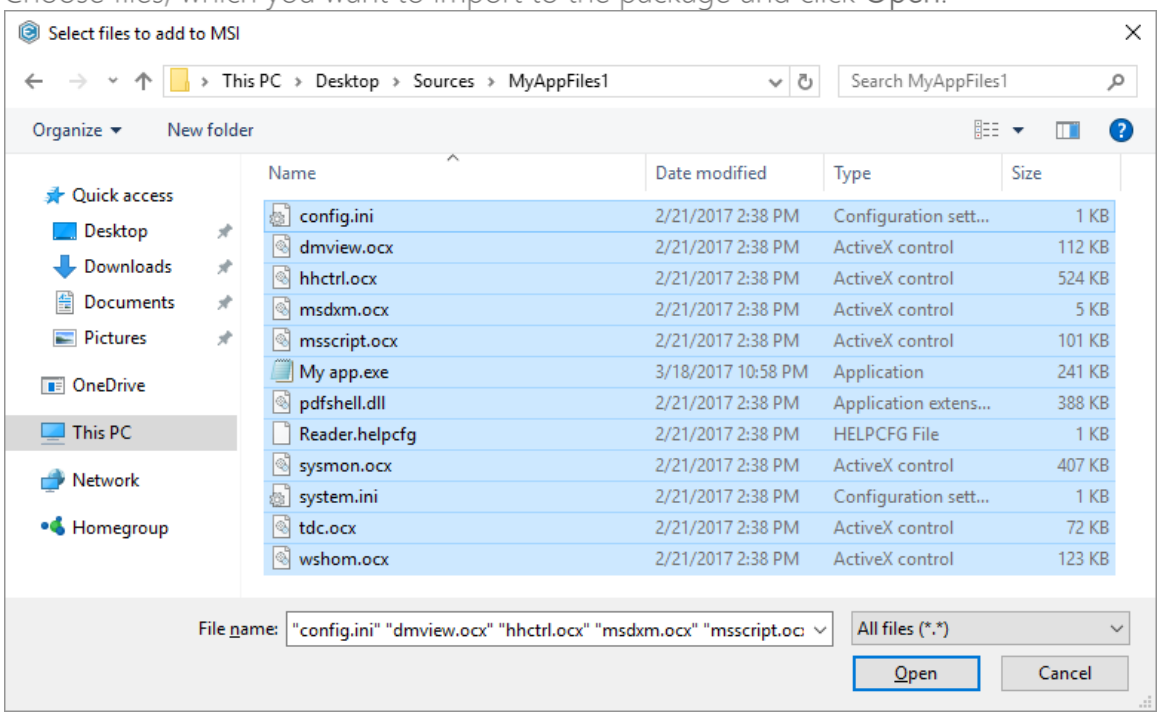

[10]. For adding files to your package, select **Import files** from the context menu of a folder, to which you want to import files.



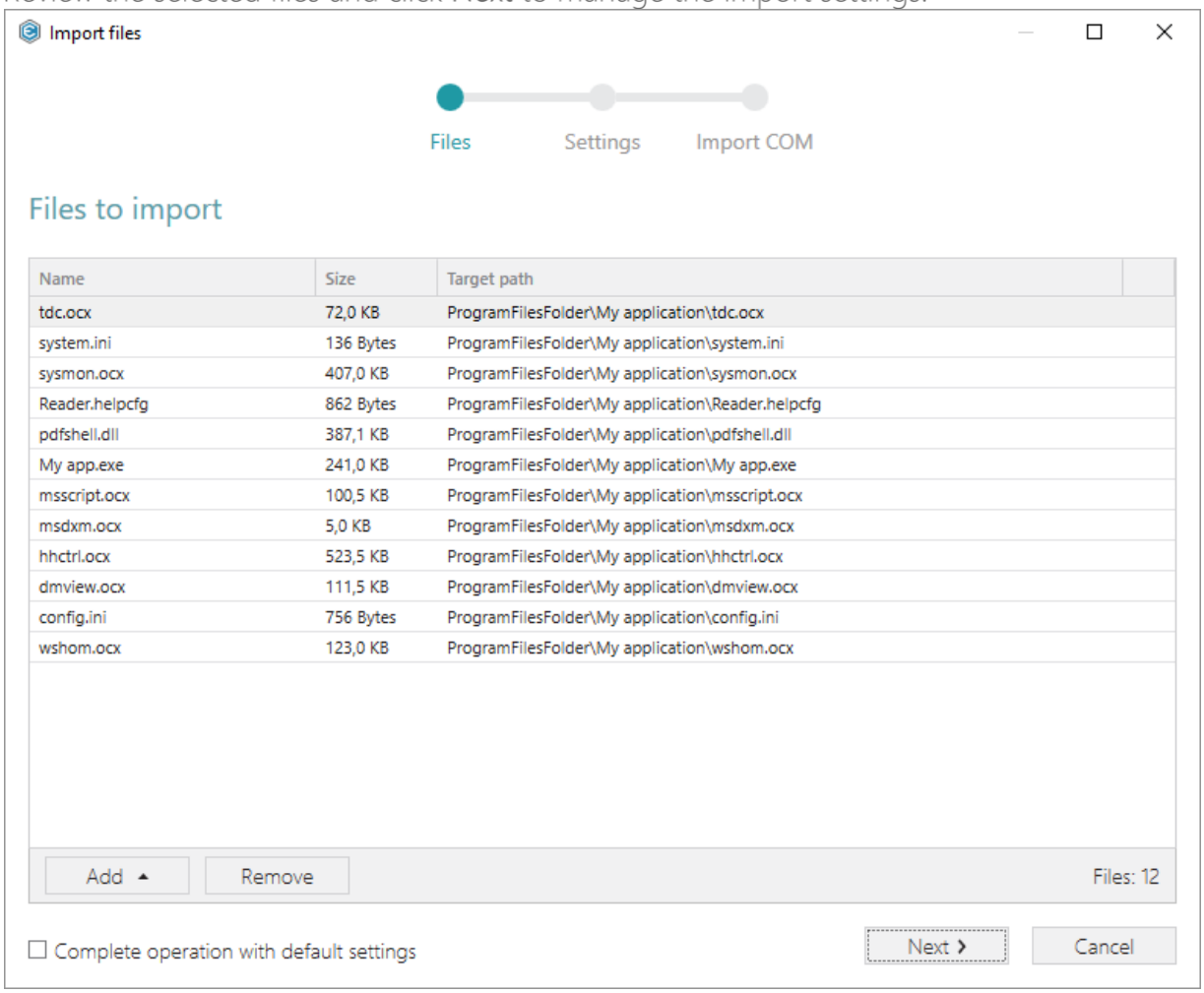
[11]. Click **Add** -> **Add files** to choose files in a folder or **Add** -> **Add folder** to choose a folder with subfolders and files.



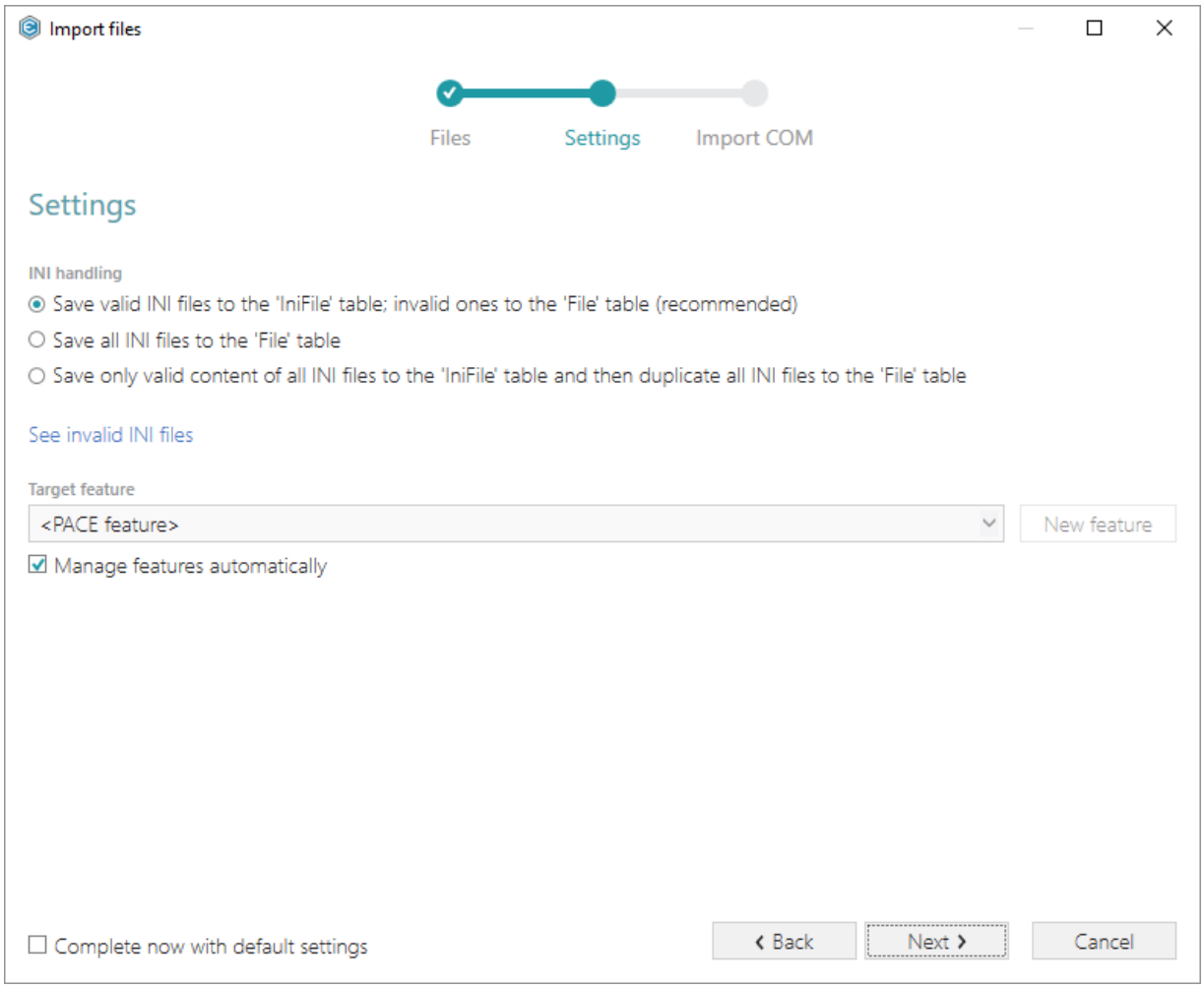

[12]. Choose files, which you want to import to the package and click **Open**.



[13]. Review the selected files and click **Next** to manage the import settings.




[14]. At this step, you can manage how to save INI files, select an MSI Feature, to which components with files will be assigned. Below, in the tables, you will find the detailed description of these settings. Note that files compression types and other Media settings are represented on the separate **Media settings** tab of the **Package Designer**. Choose the desired settings and click **Next** to continue.




## INI handling

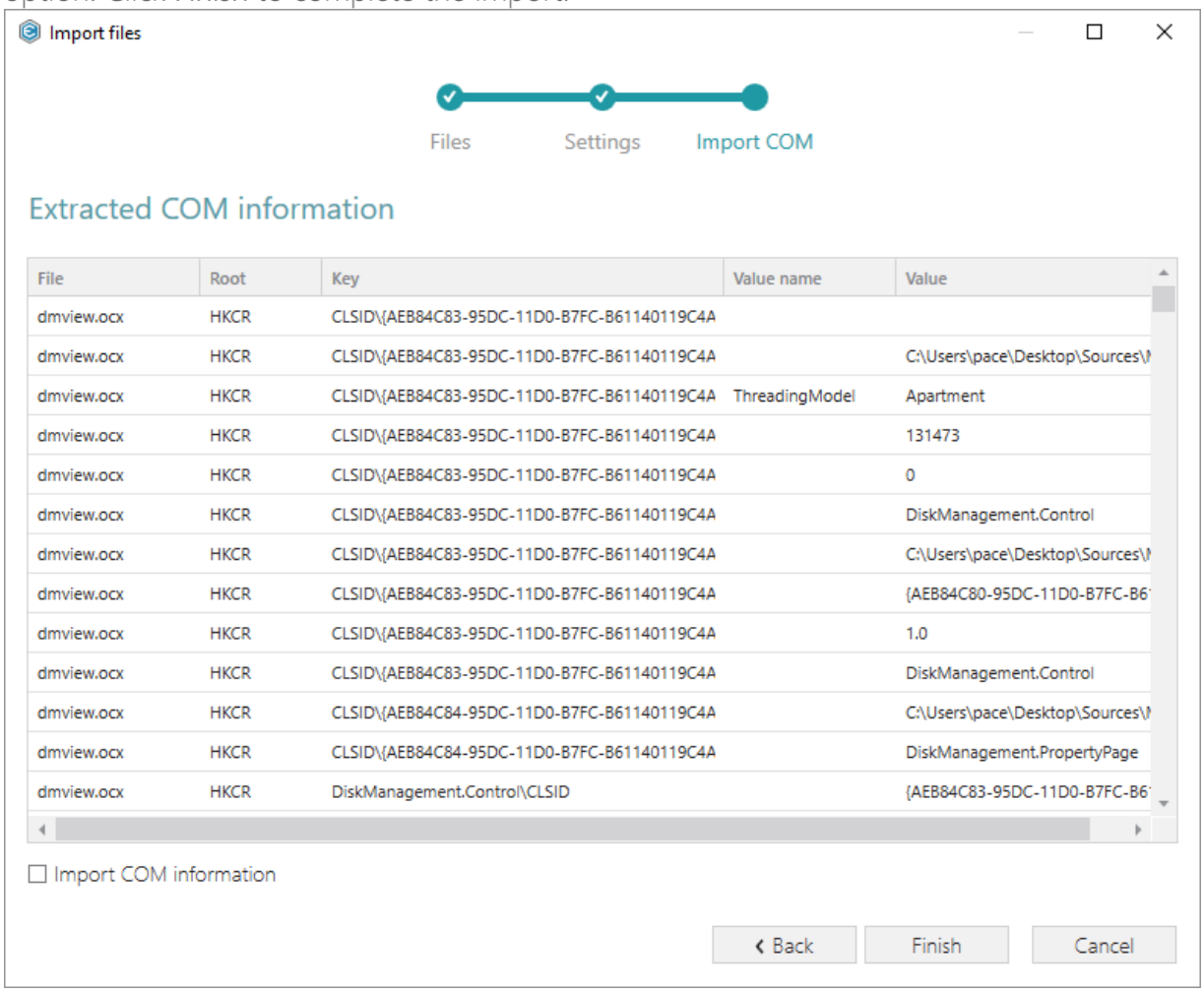
| | |
|---|---|
| Save valid INI files to the 'IniFile' table; invalid ones to the 'File' table (recommended) | INI files, which comply with the INI file format, will be saved to the 'IniFile' table.<br>All other INI files, which contain unsupported data, will be saved to the 'File' table as binary files. |
| 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.<br>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.<br>Feature structure is based on the packaging best practices and depends on the destination location of files. |

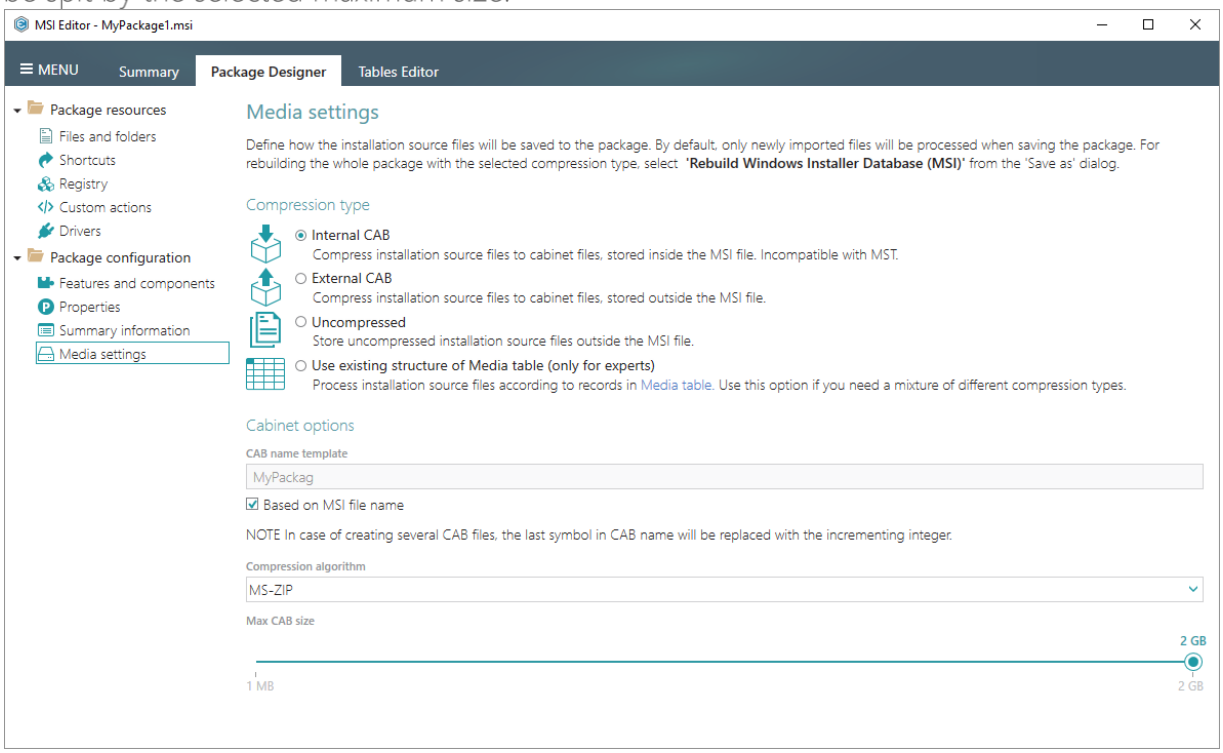
| | |
|---|---|
| <a feature, selected from the list> | Components with files will be assigned to a feature, selected from the list. |

[15]. On the last step you can review COM information, extracted from the selected files. For importing this COM information along with files, select the Import COM information option. Click Finish to complete the import.

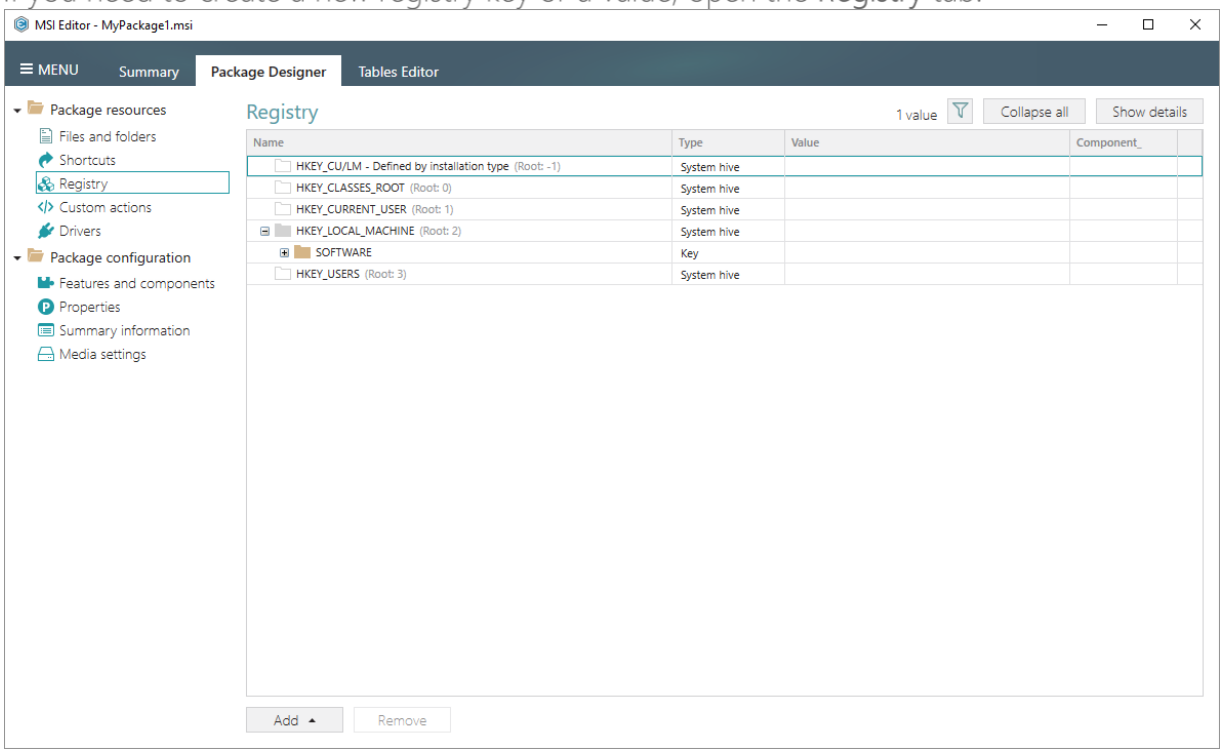



[16]. Go to the Media settings tab of the Package Designer and choose the necessary compression type for the imported files. For the compressed files you can manage cabinet options such as CAB name, compression algorithm and maximum CAB size. Cabinets will

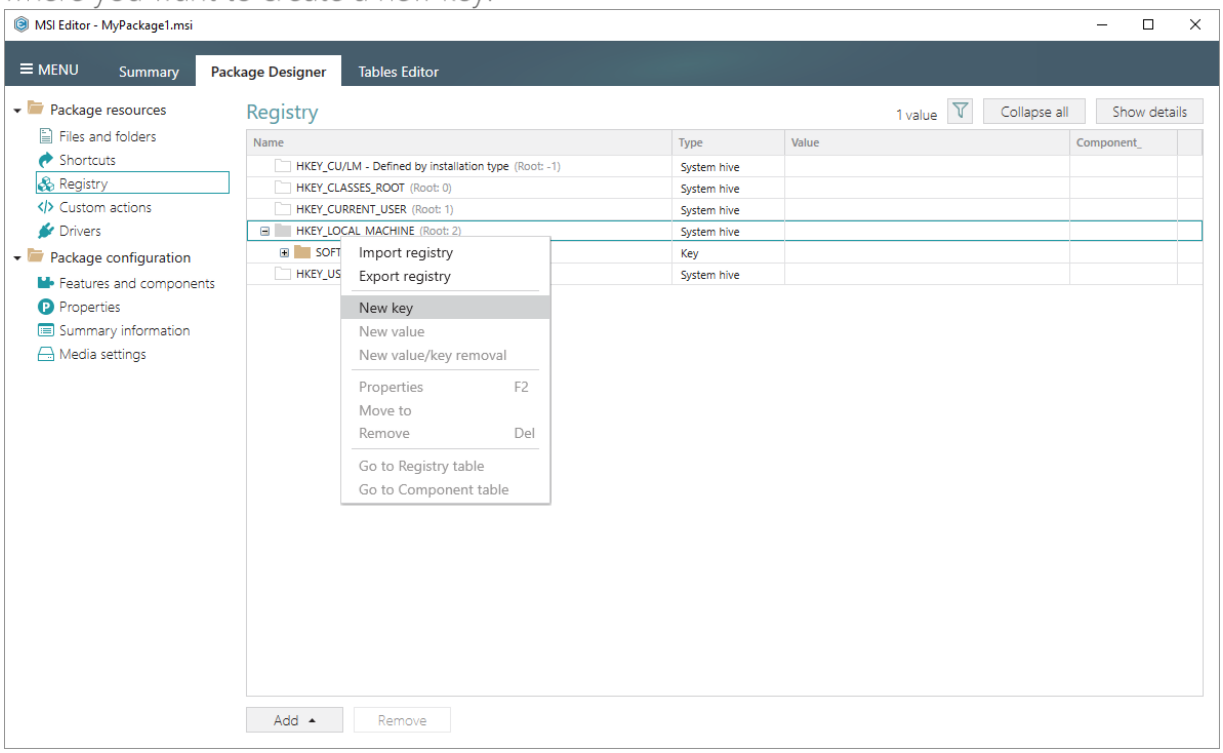
be spit by the selected maximum size.



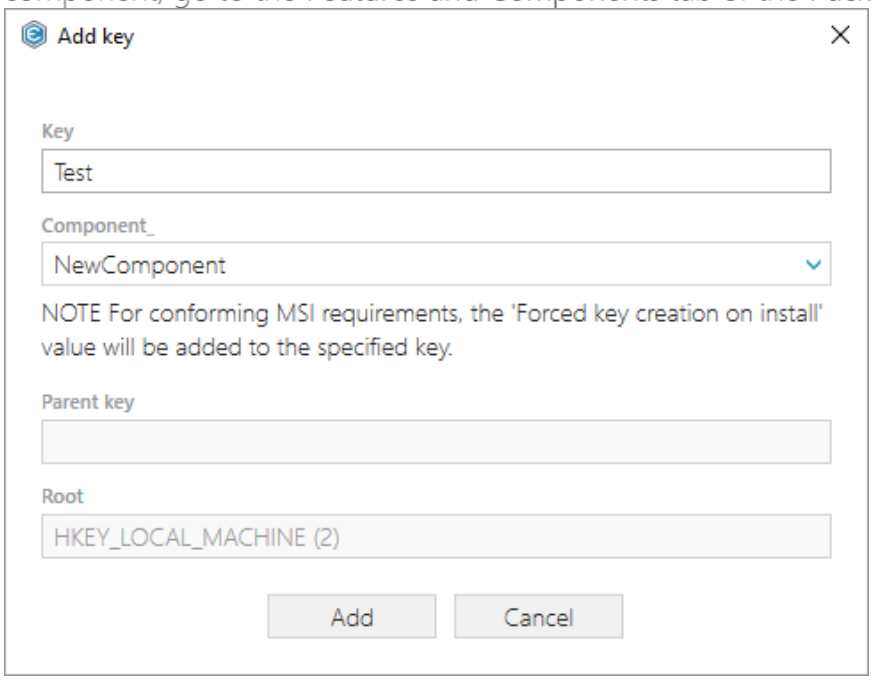

[17]. If you need to create a new registry key or a value, open the **Registry** tab.



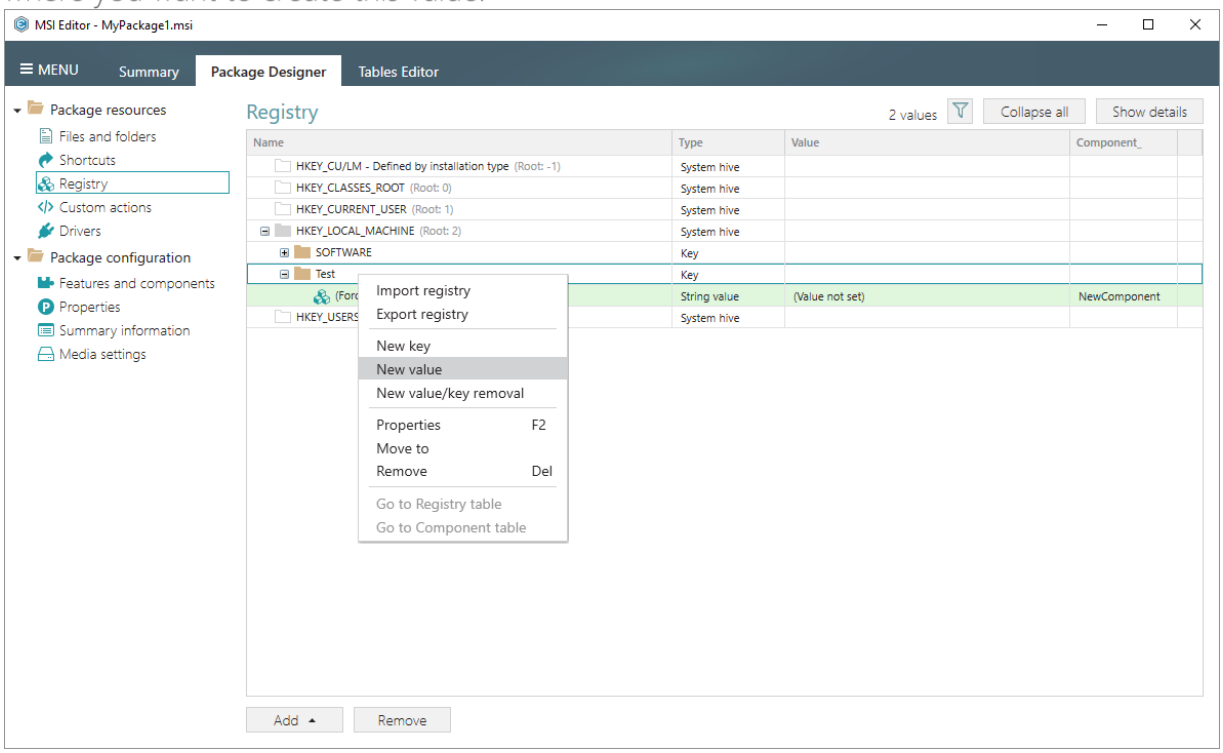
[18]. Select **New key** from the context menu of a root hive (e.g. HKEY_LOCAL_MACHINE), where you want to create a new key.




[19]. 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**.



[20]. In order to create a new registry value, select **New value** from the context menu of a key, where you want to create this value.

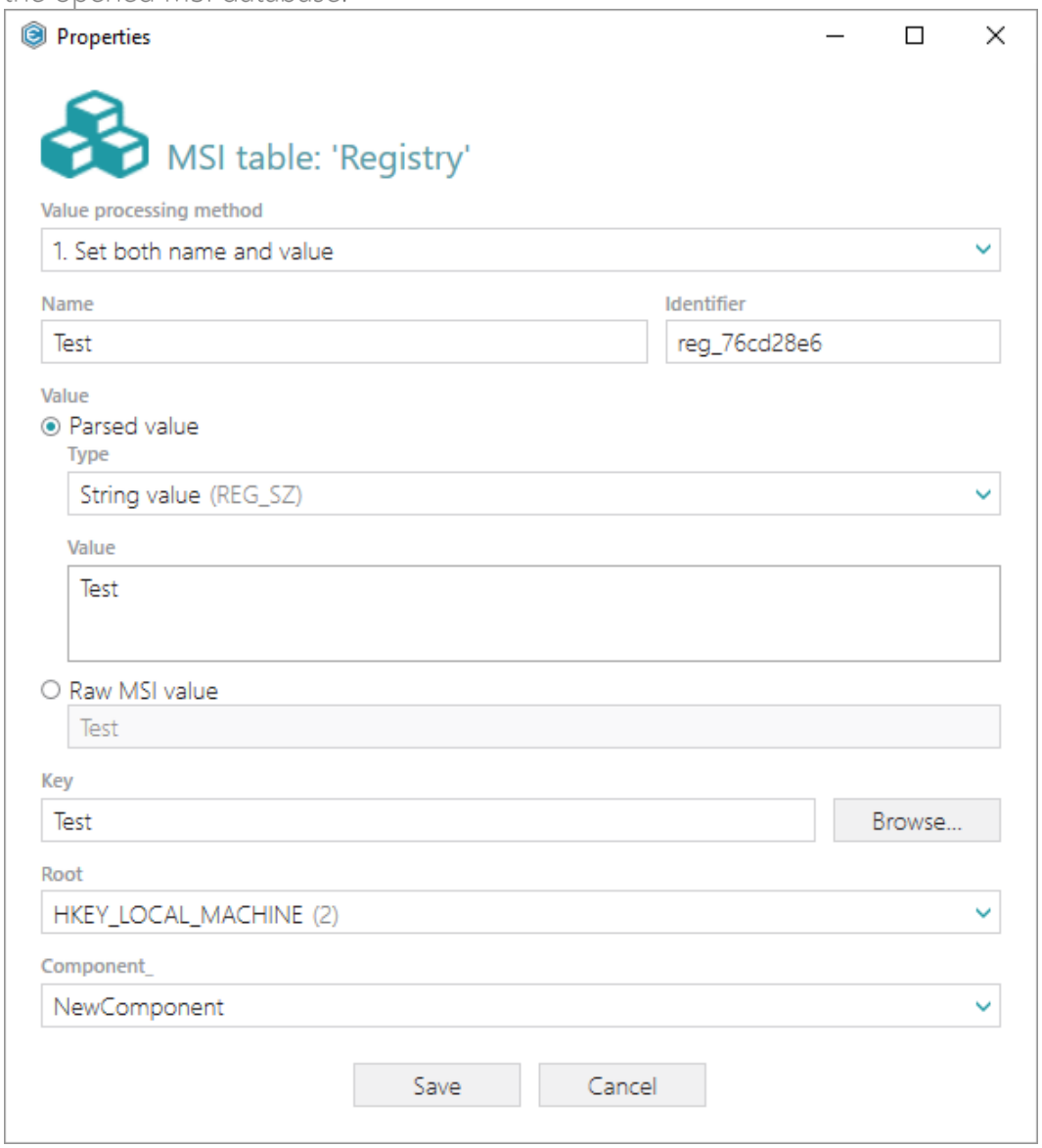



[21]. For creating an ordinary registry value, specify a name and its value, choose a component from the list, to which this value will be assigned, and click **Save**.

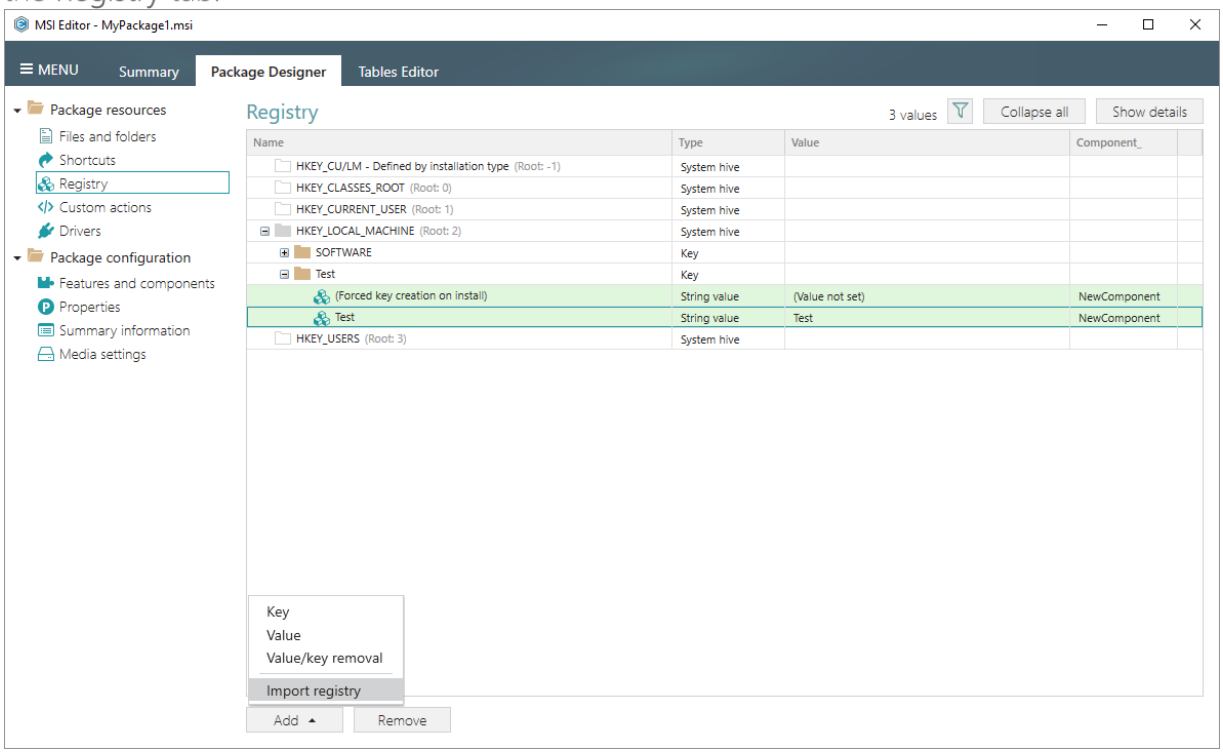
For creating a Default key value, or securing a key creation and/or removal, select a respective option from the **Value processing method** field.

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

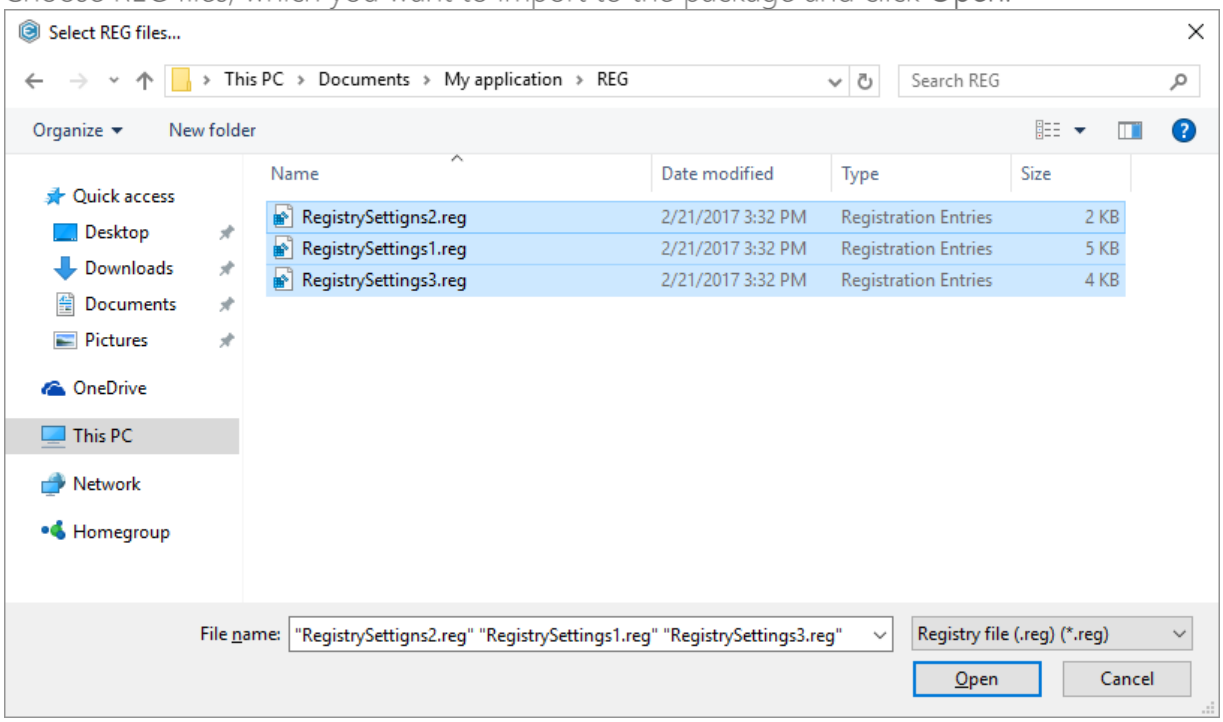
the opened MSI database.



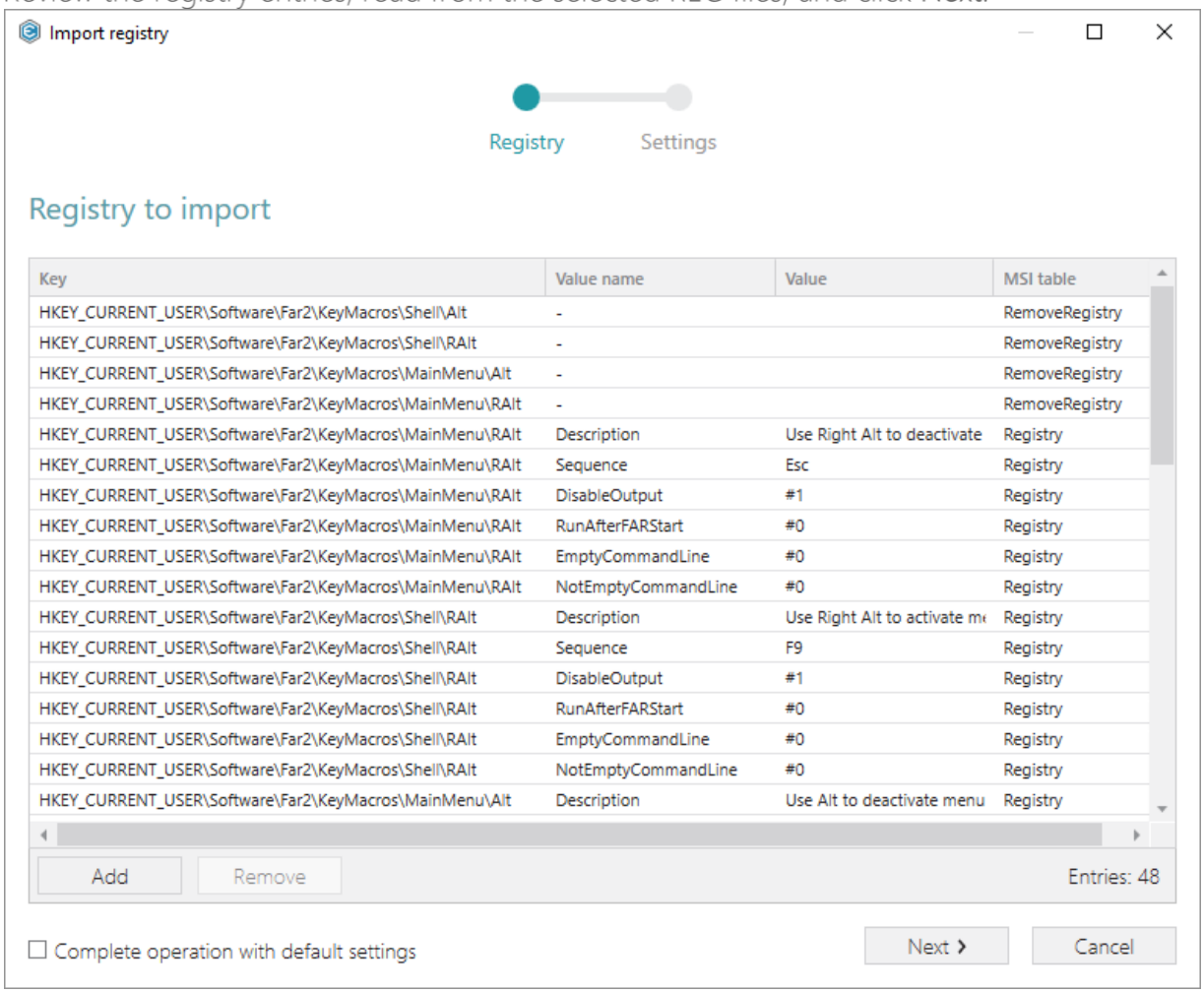
[22]. For importing registry from the REG file, select **Add** -> **Import registry** from the toolbox of the **Registry** tab.




[23]. Choose REG files, which you want to import to the package and click **Open**.

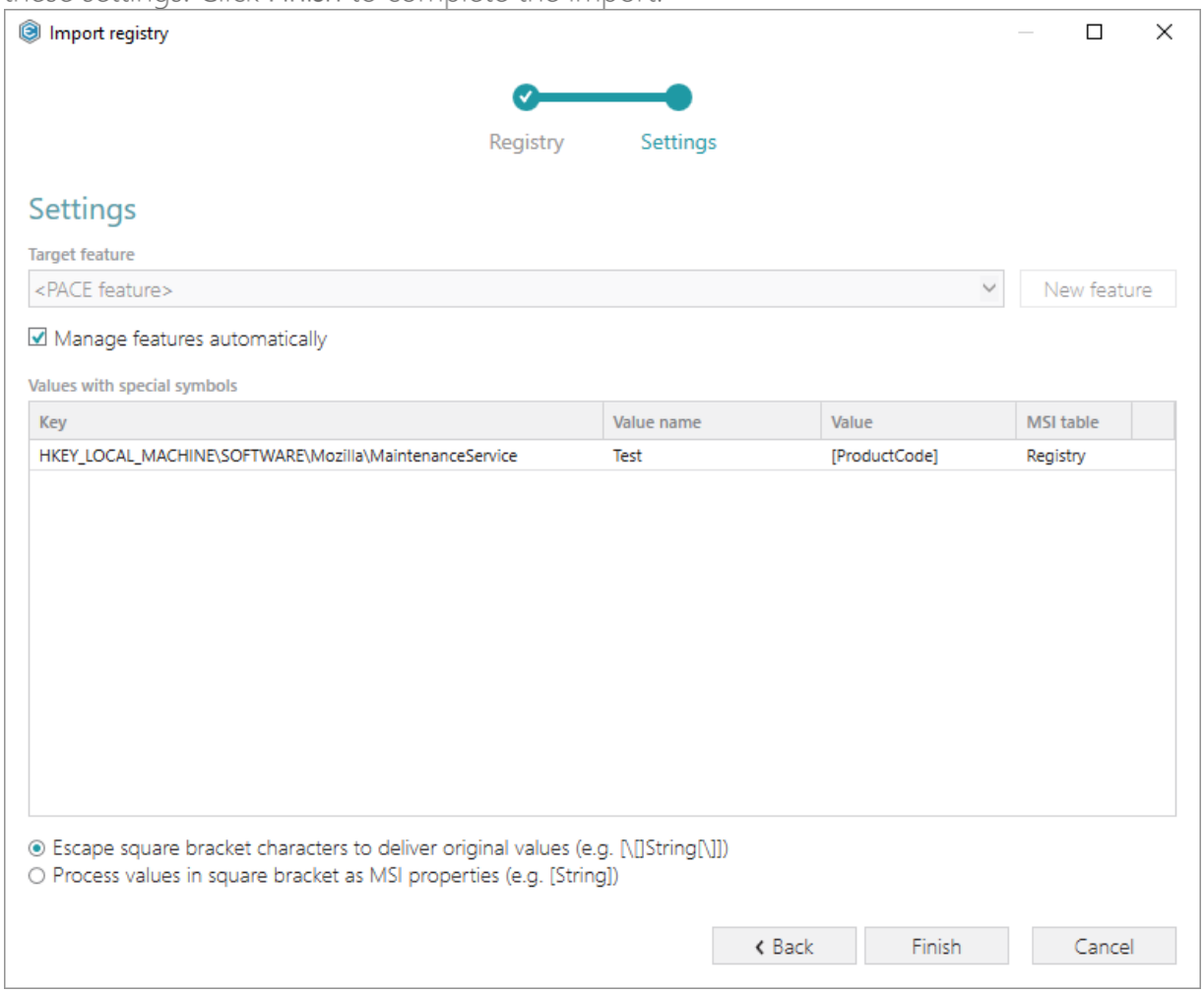


[24]. Review the registry entries, read from the selected REG files, and click Next.




[25]. 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

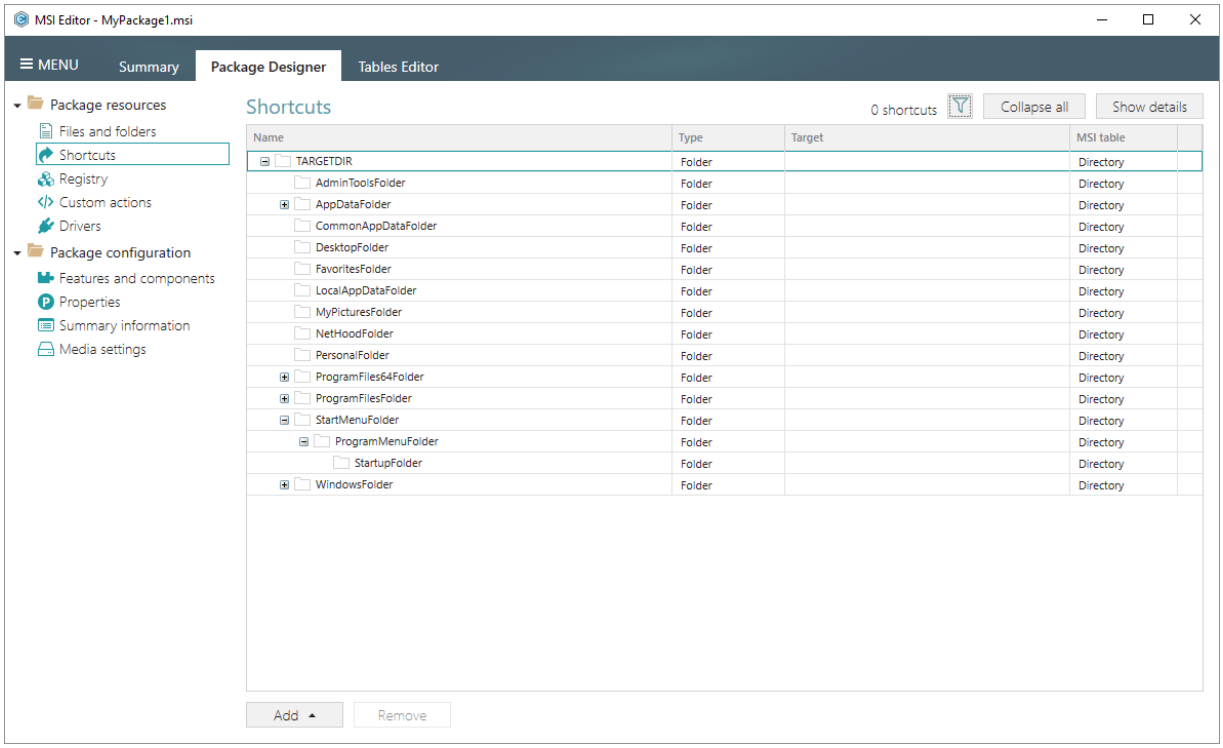
| | |
|---|---|
| Manage features automatically | The specific PACE features will be created (or used if they exist) for the components, containing the imported registry. Feature structure is based on the packaging best practices and depends on the registry key location. |
| <a feature, selected from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

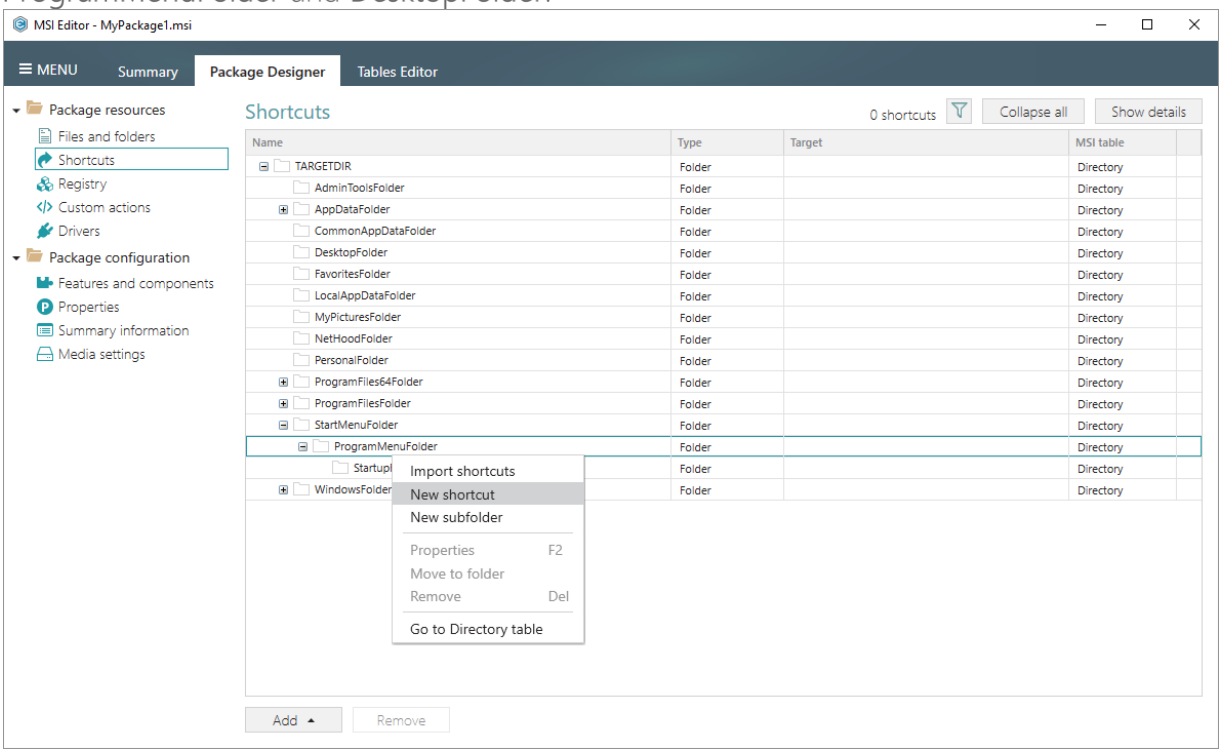
| | |
|---|---|
| Process values in square brackets as MSI properties | Values in square brackets, displayed in the list, will be saved to MSI as MSI properties. Thus, each MSI property will be replaced with its value during the MSI installation. If a property does not exist in MSI during the installation,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

[26]. If you need to create a new or import the existing shortcuts to the package, go to the Shortcuts tab.



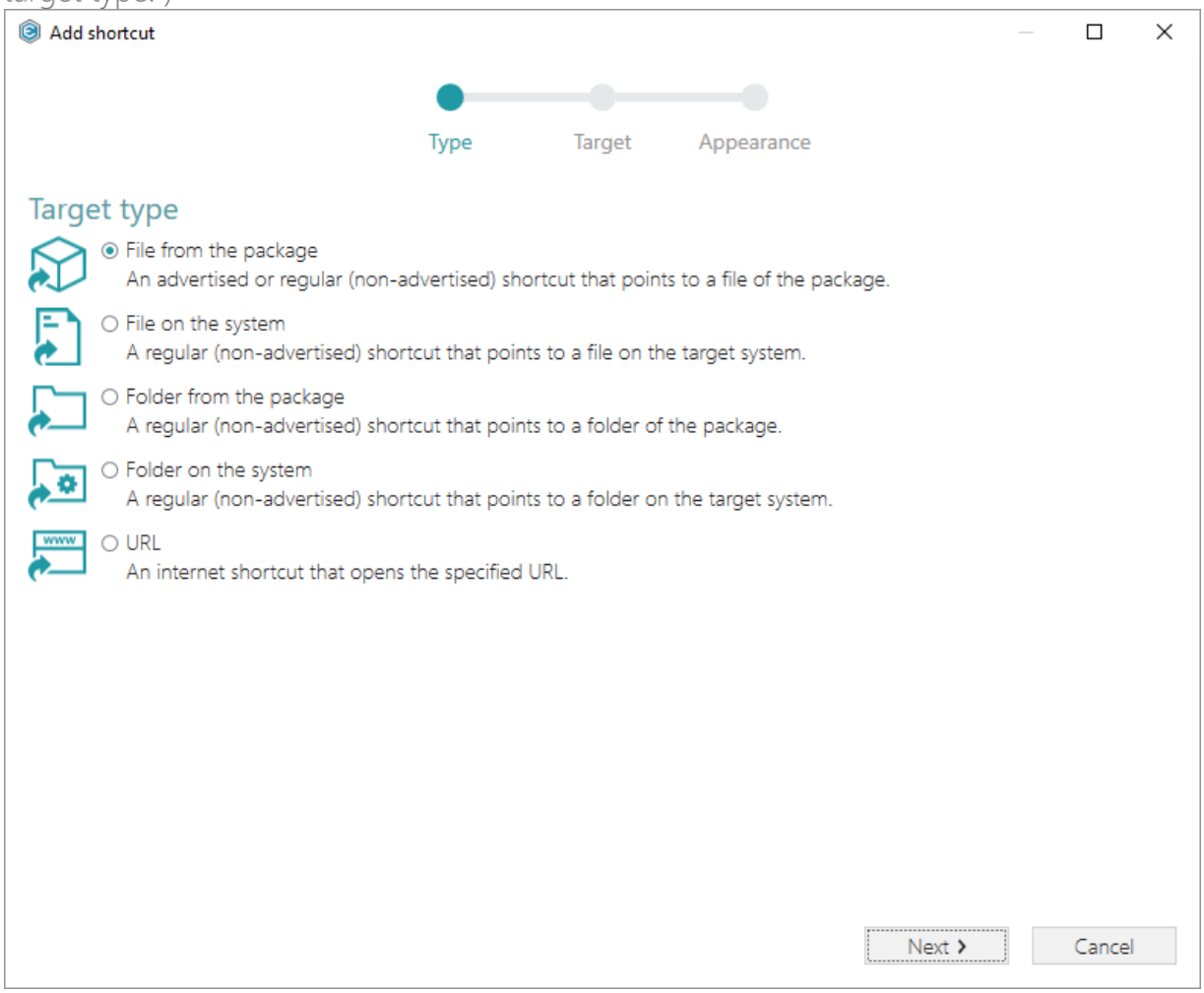

[27]. In order to add a new shortcut, select New shortcut from the context menu of a folder, where you want to create this shortcut. Usually, application shortcuts are installed to the

ProgramMenuFolder and DesktopFolder.

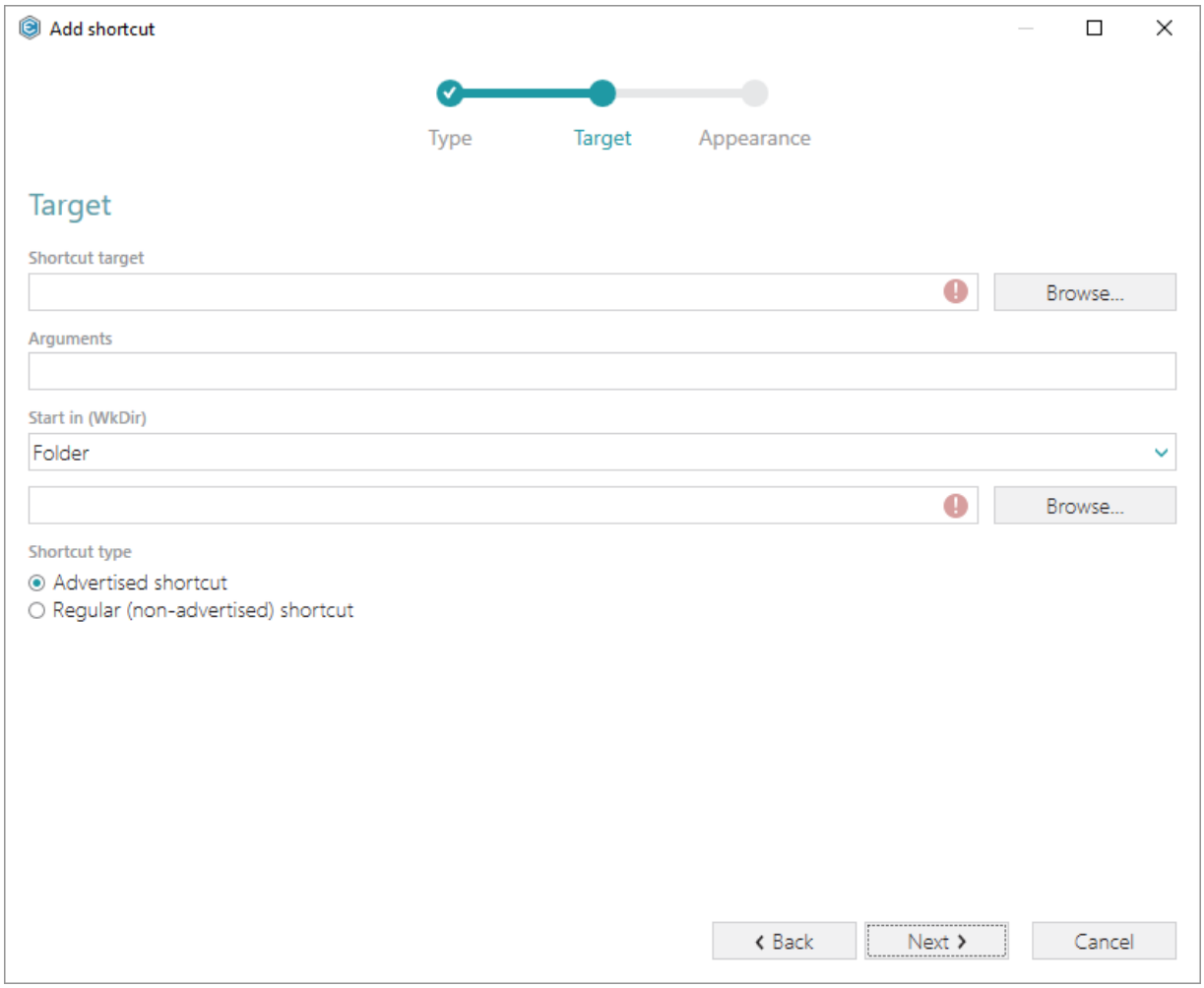



[28]. Select a type of the shortcut target and click Next.
(The following instructions describe creation of the shortcut of the 'File from the package'

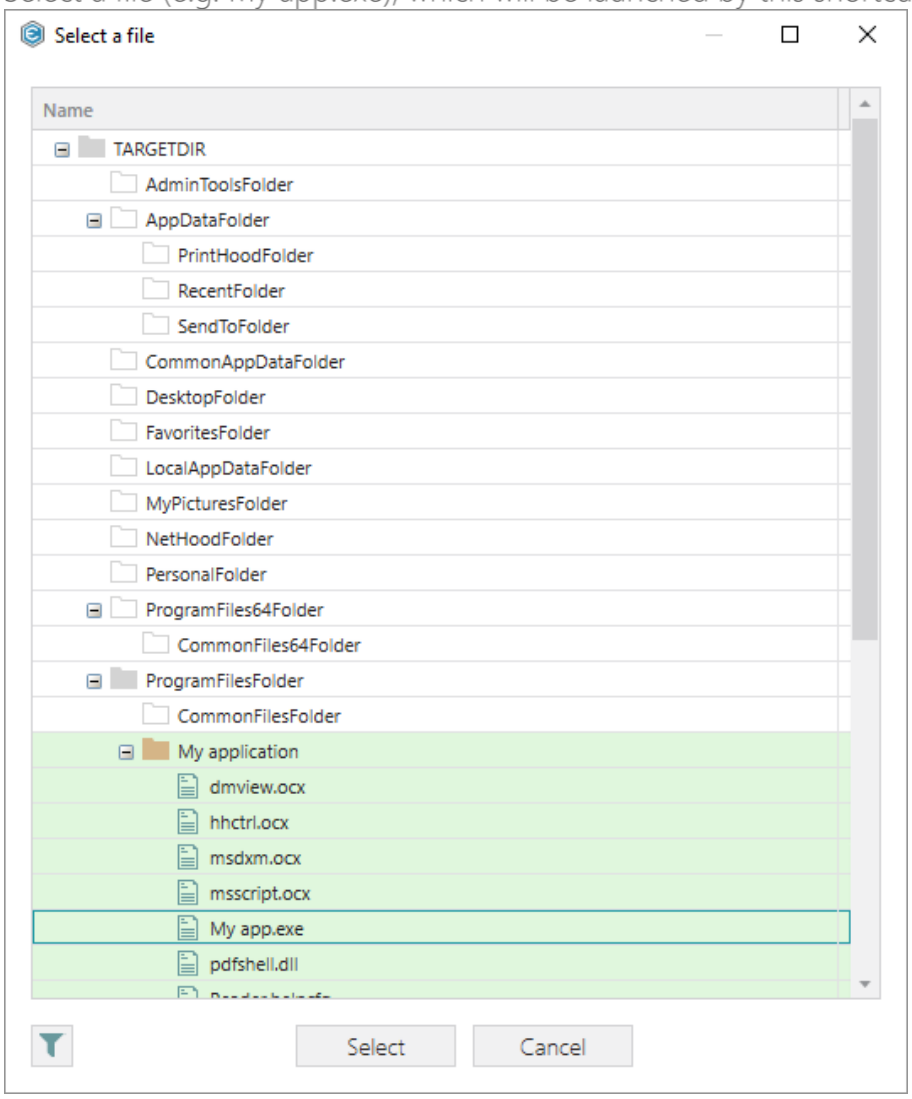
target type. )


Add shortcut
Type
Target
Appearance
Target type
File from the package
An advertised or regular (non-advertised) shortcut that points to a file of the package.
File on the system
A regular (non-advertised) shortcut that points to a file on the target system.
Folder from the package
A regular (non-advertised) shortcut that points to a folder of the package.
Folder on the system
A regular (non-advertised) shortcut that points to a folder on the target system.
URL
An internet shortcut that opens the specified URL.
Next ›
Cancel

[29]. On the second step, click **Browse**, located next to the **Shorcut target** field, to set a target of the shortcut.

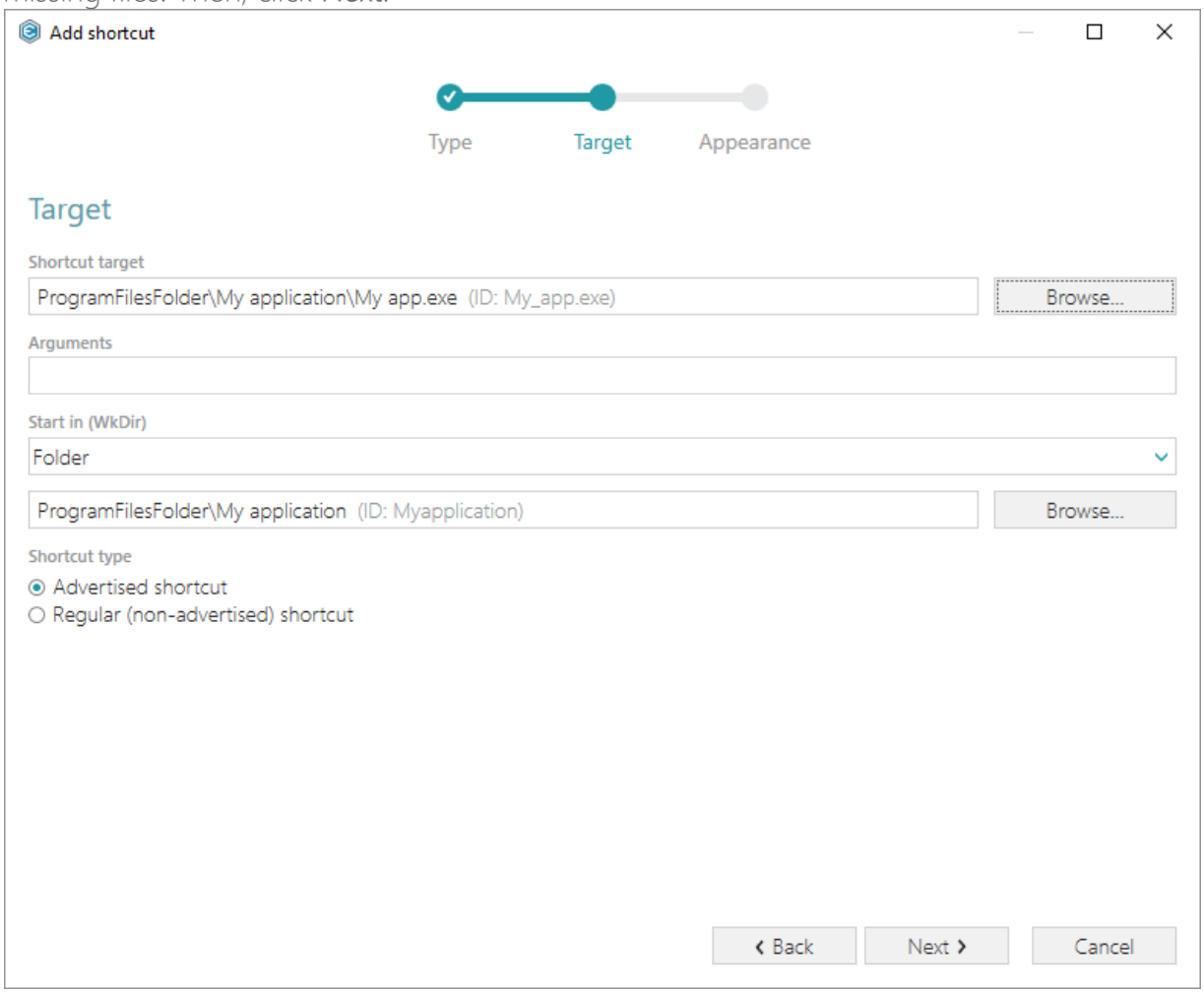


[30]. Select a file (e.g. My app.exe), which will be launched by this shortcut, and click **Select**.

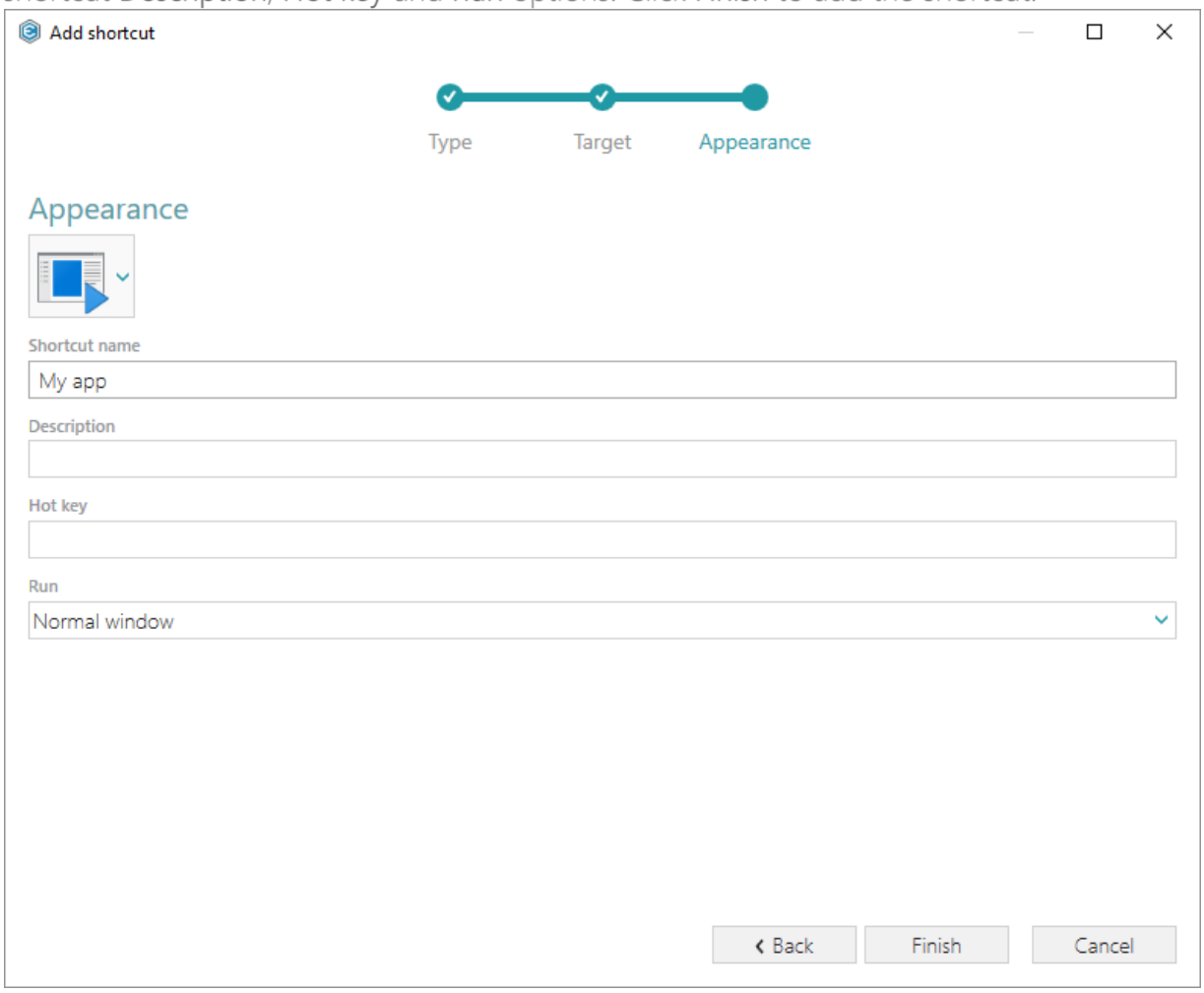



[31]. If needed, enter a shortcut **Arguments**, change the **Start in** path (select a folder or a property), and select a **Shortcut type**. Note that only the Advertised shortcut supports advertisement and repair. This means that if the executable to which the shortcut is pointing is not there then windows installer will repair the application and replace the

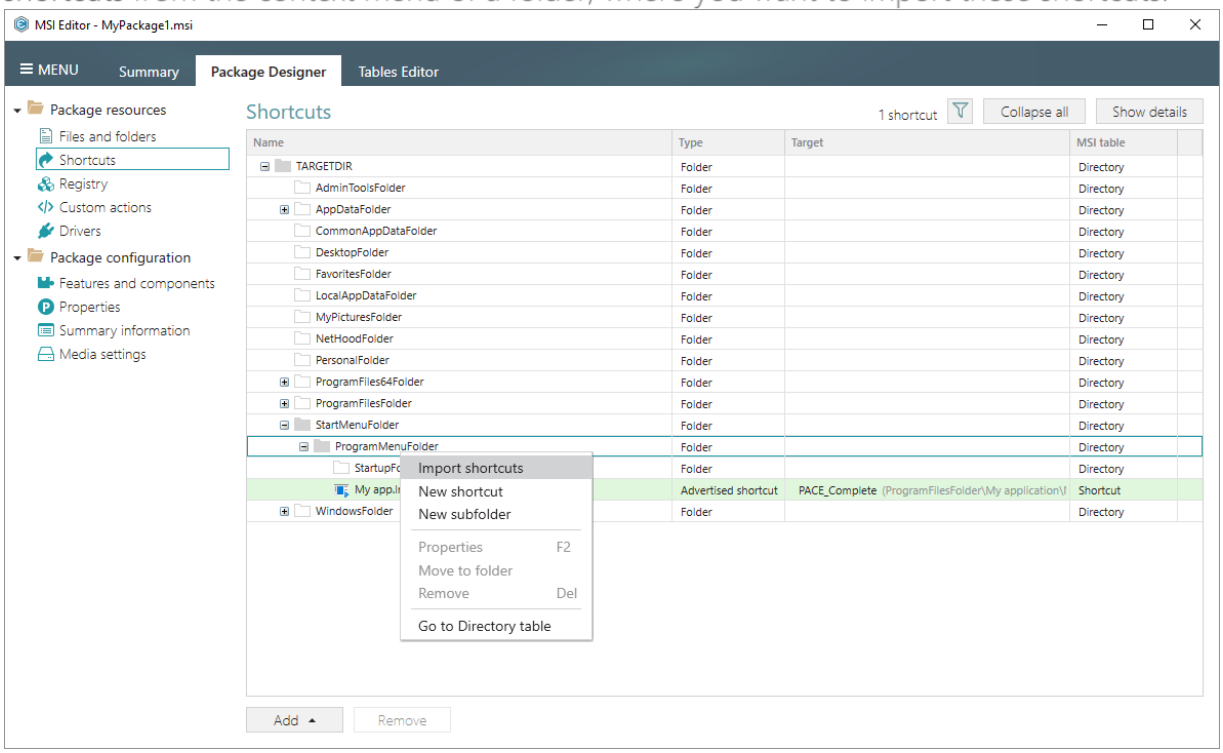
missing files. Then, click **Next**.



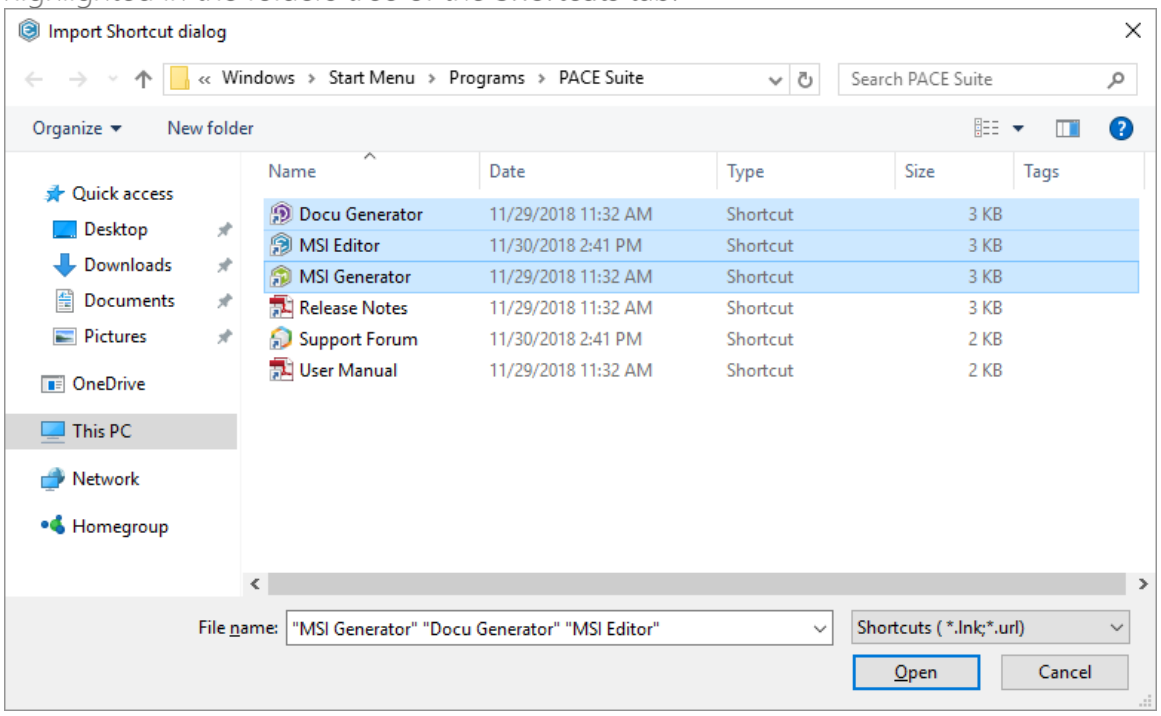
[32]. On the last step, enter a **Shortcut name**, choose a shortcut icon and, if needed, specify the shortcut **Description**, **Hot key** and **Run** options. Click **Finish** to add the shortcut.



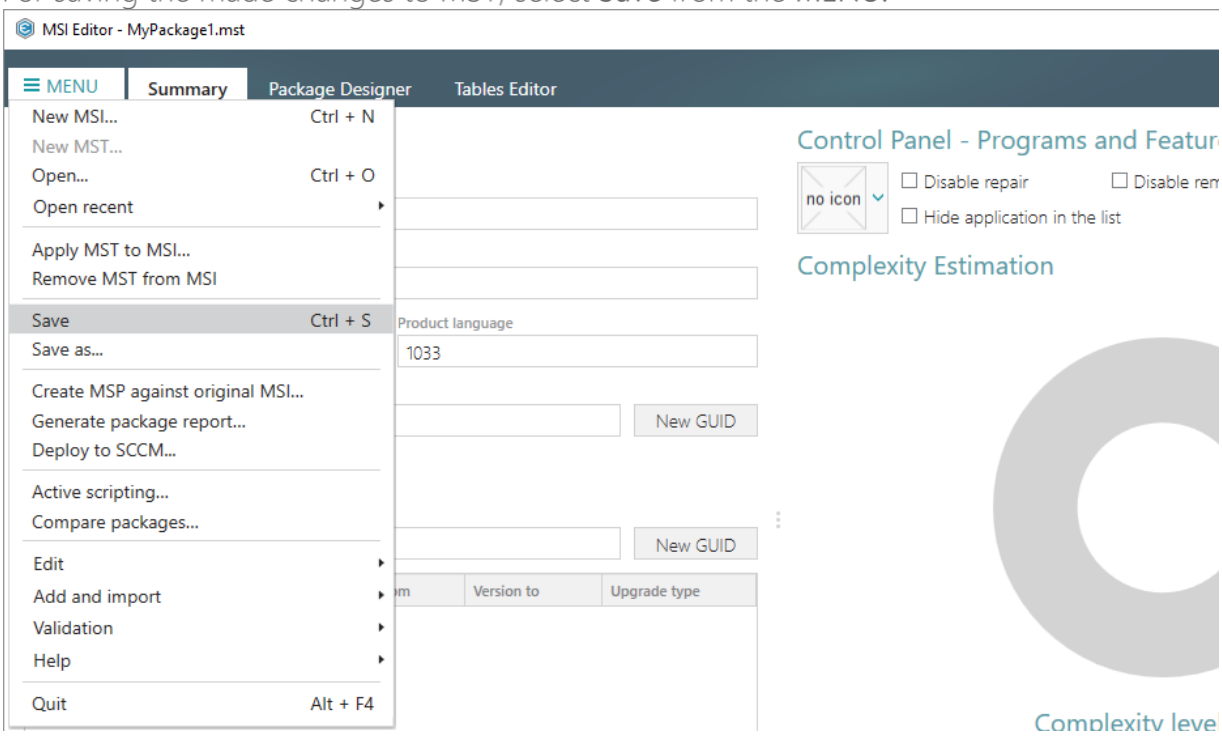
[33]. In order to import existing shortcuts from the file system to the package, select **Import shortcuts** from the context menu of a folder, where you want to import these shortcuts.




[34]. Choose shortcuts you would like to import and click **Open**. The imported shortcuts will be highlighted in the folders tree of the **Shortcuts** tab.



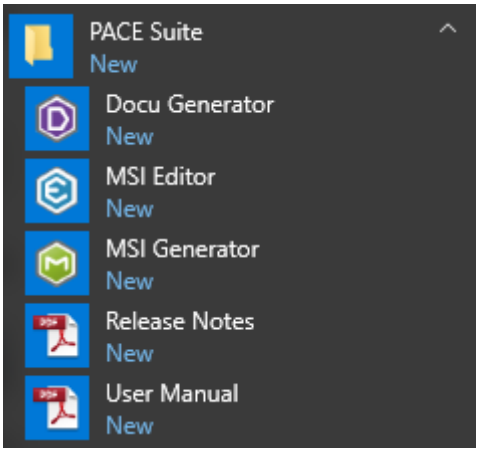
[35]. For saving the made changes to MST, select Save from the MENU.



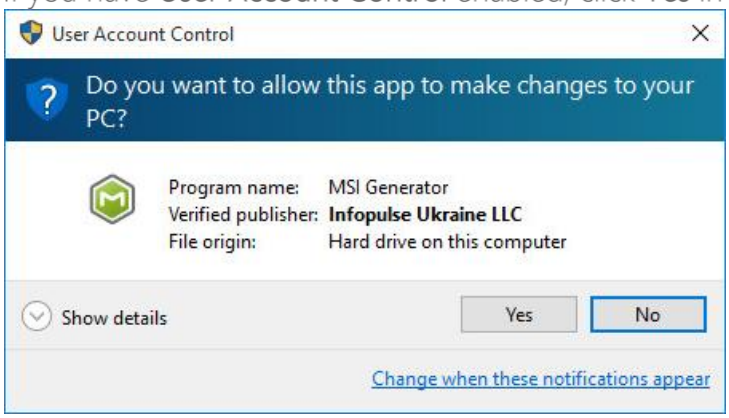

## 3.2.2 Create Response MST

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.

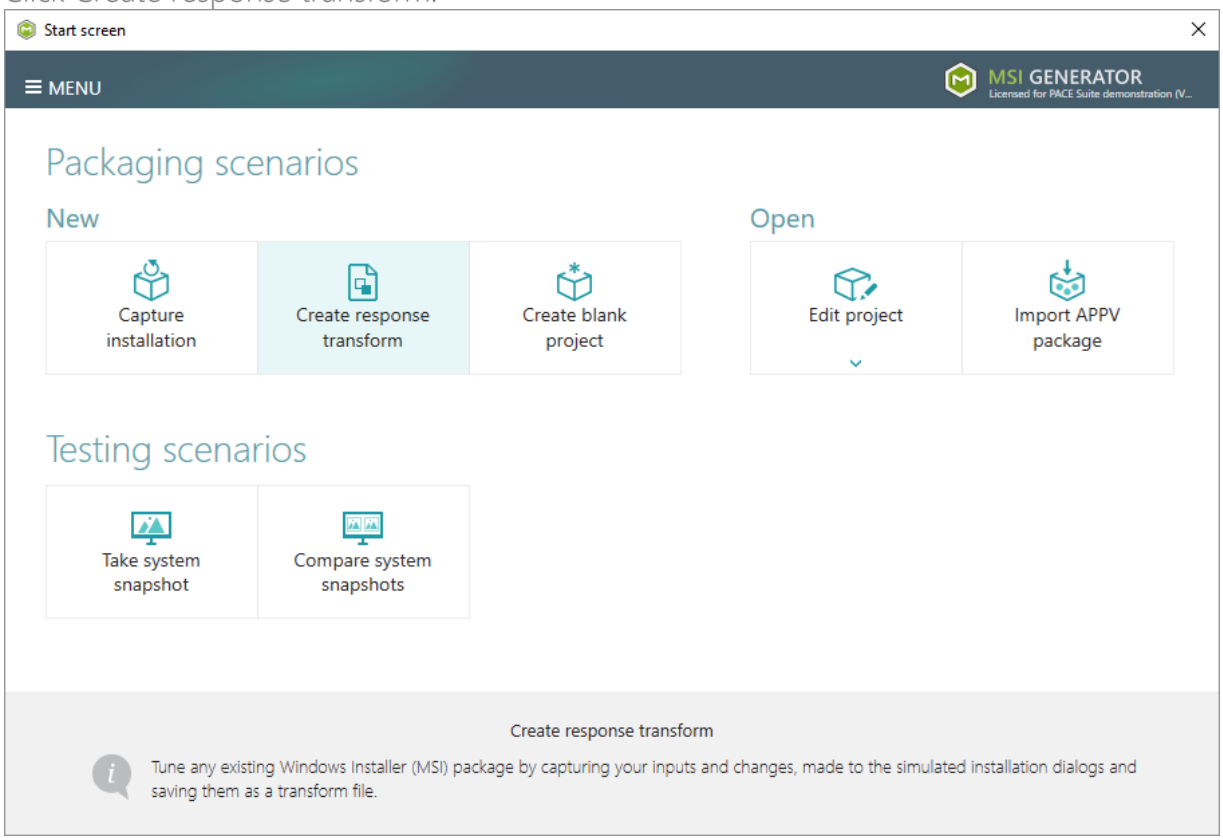
[1]. Launch MSI Generator from the desktop or the start menu shortcut.



[2]. If you have User Account Control enabled, click Yes in the opened window.

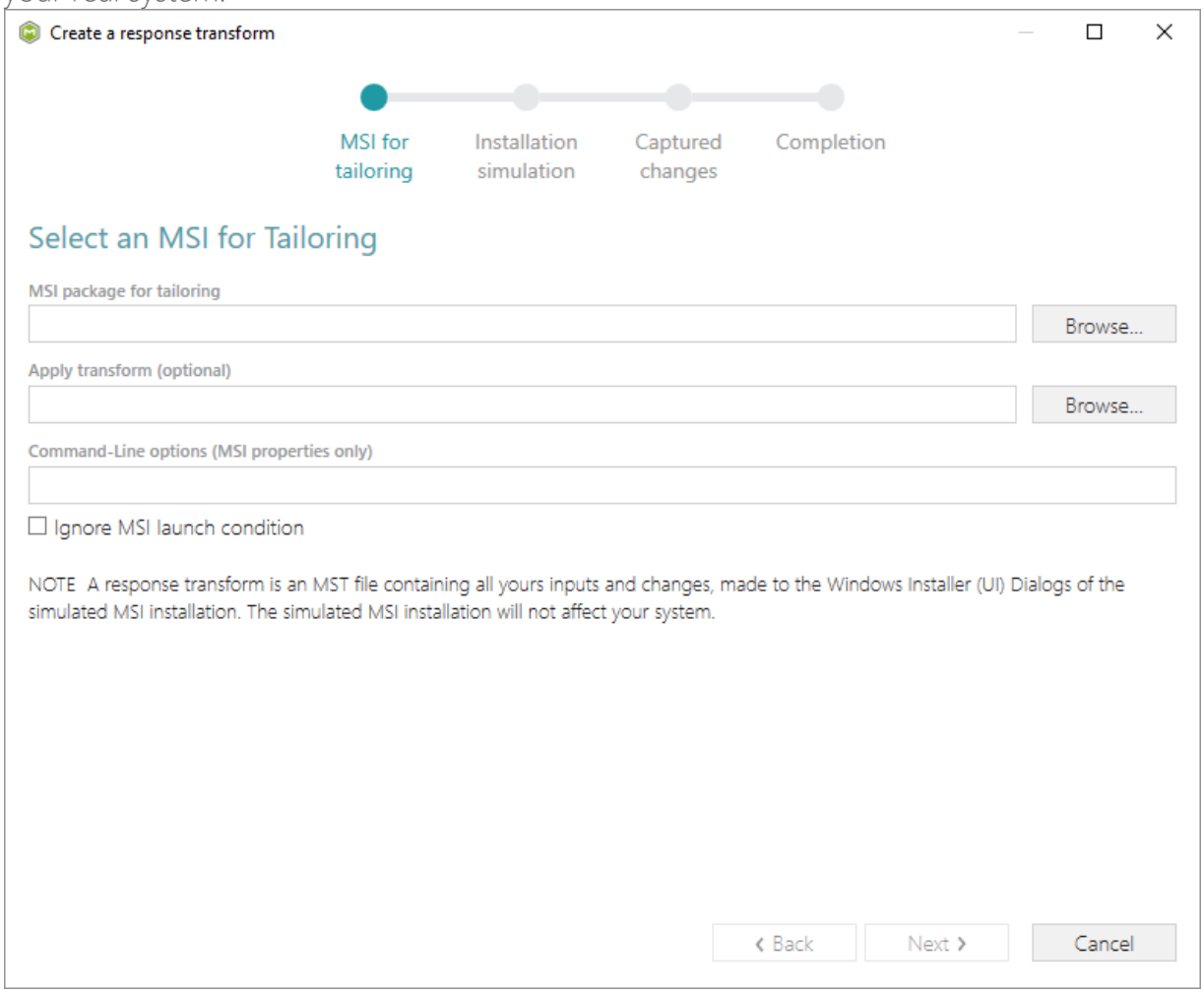



[3]. Click Create response transform.

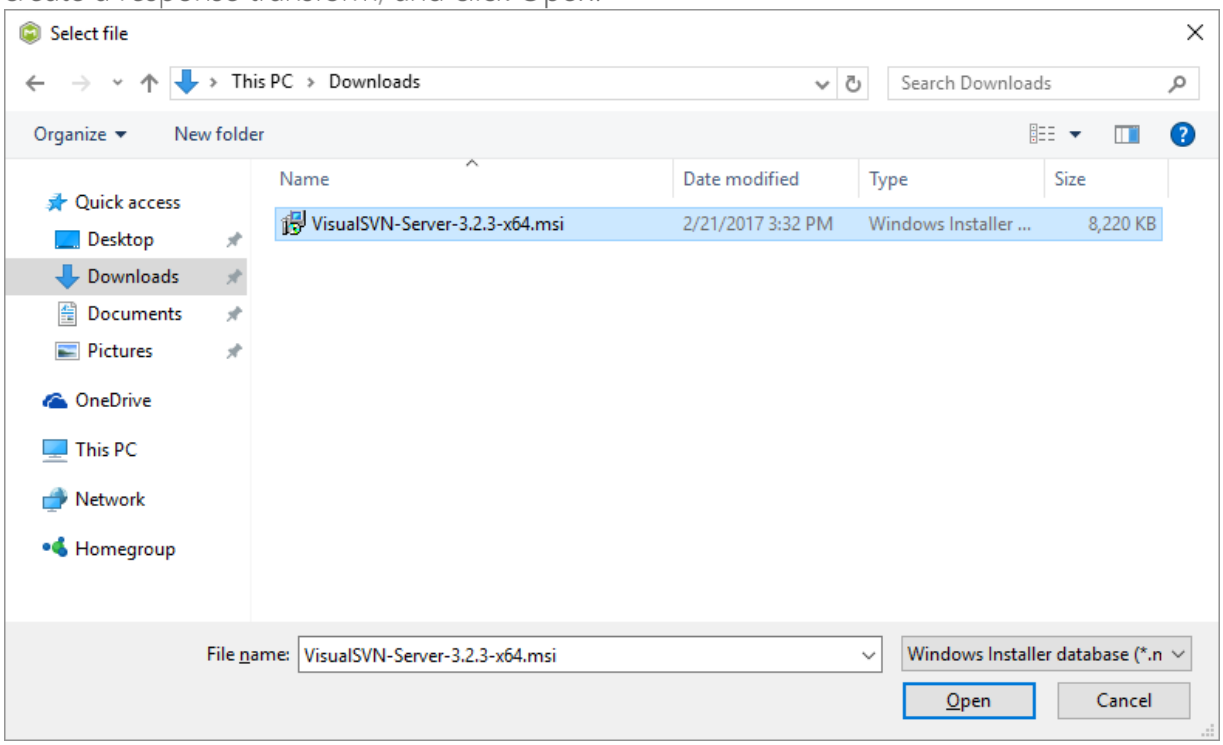



[4]. Click Browse..., located next to the MSI package for tailoring field, to select an MSI package for tailoring. Note that the MSI installation will be simulated and will not affect

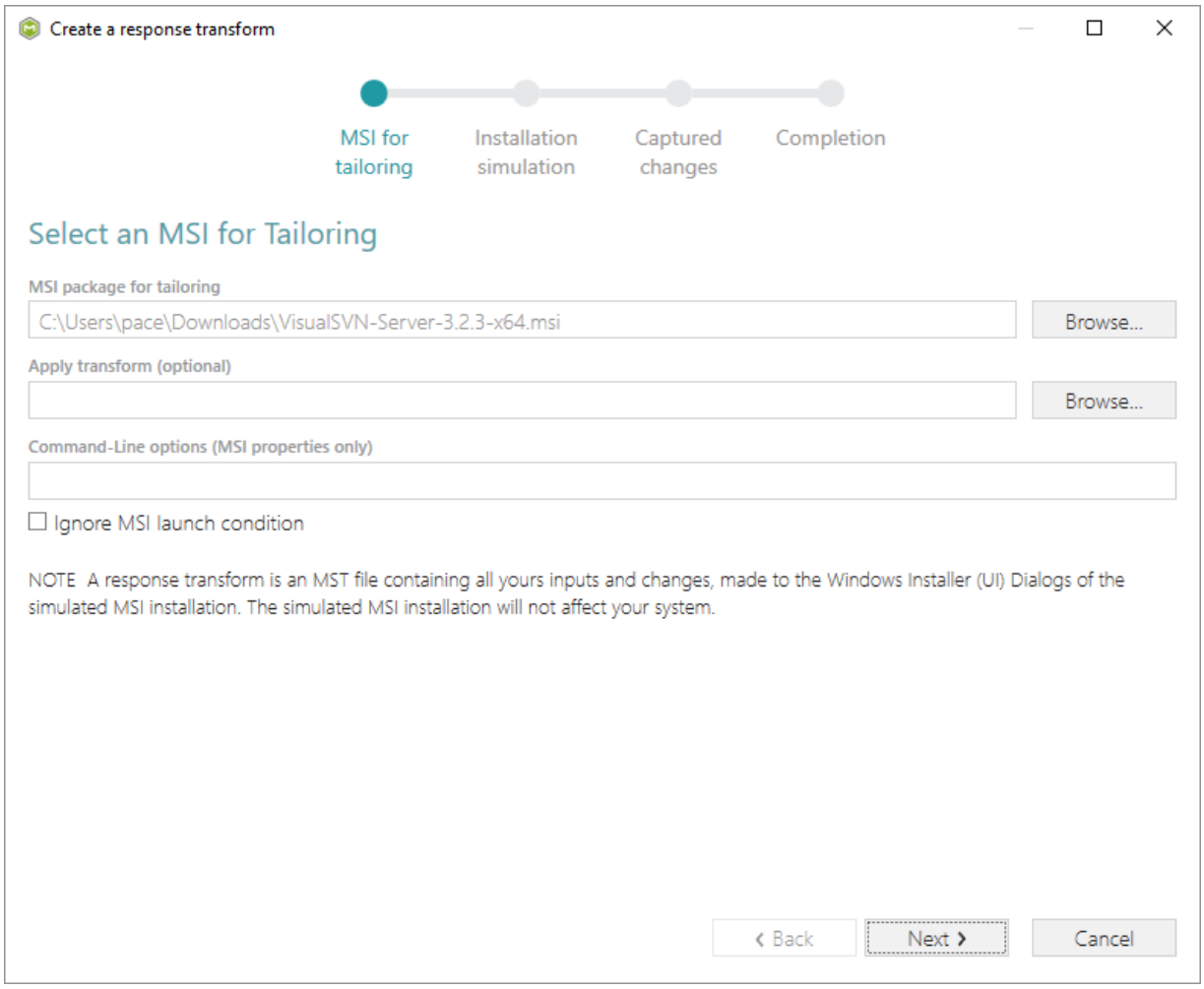
your real system.



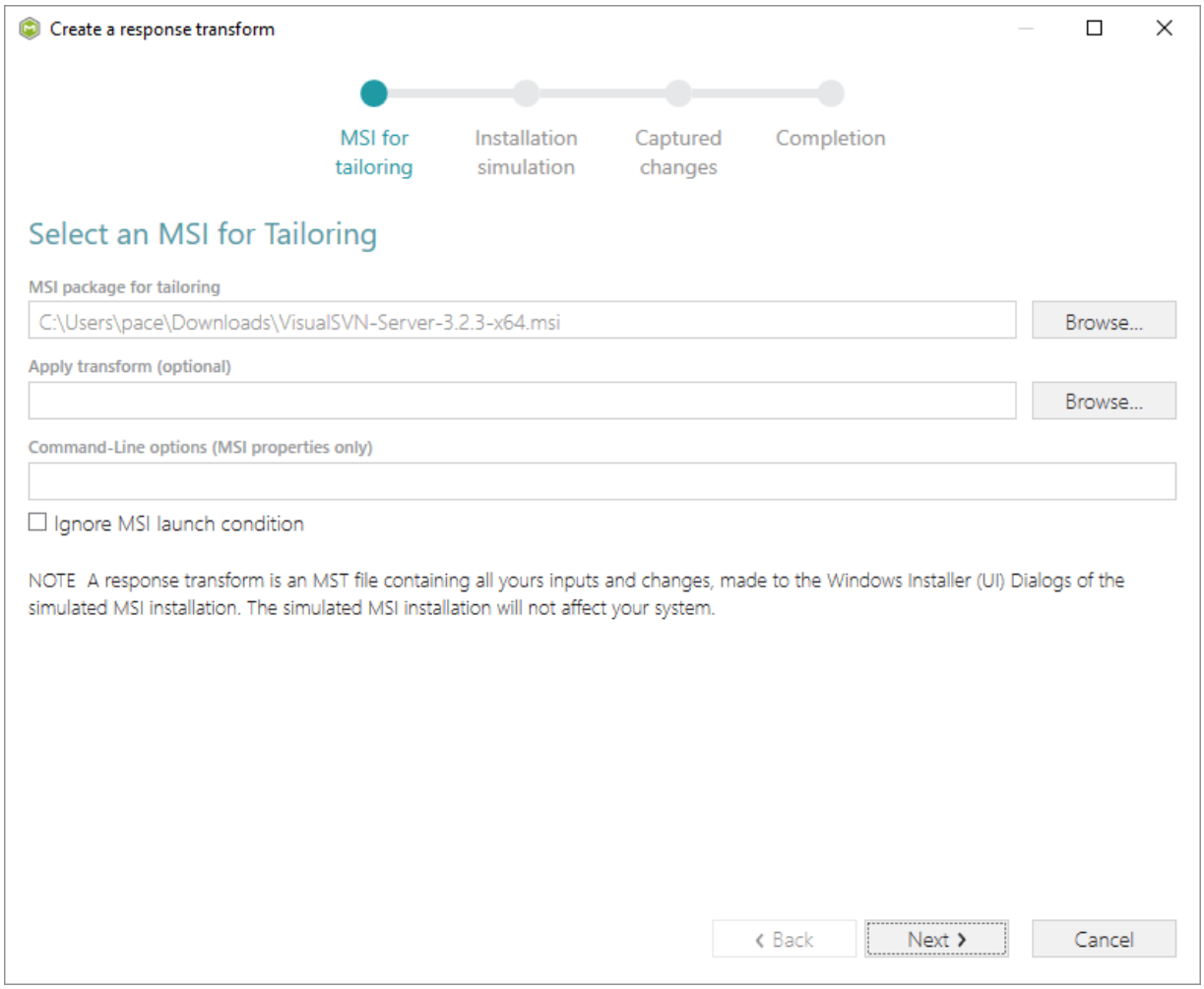

[5]. Choose an MSI package (e.g. VisualSVN-Server-3.2.3-x64.msi), for which you want to create a response transform, and click Open.



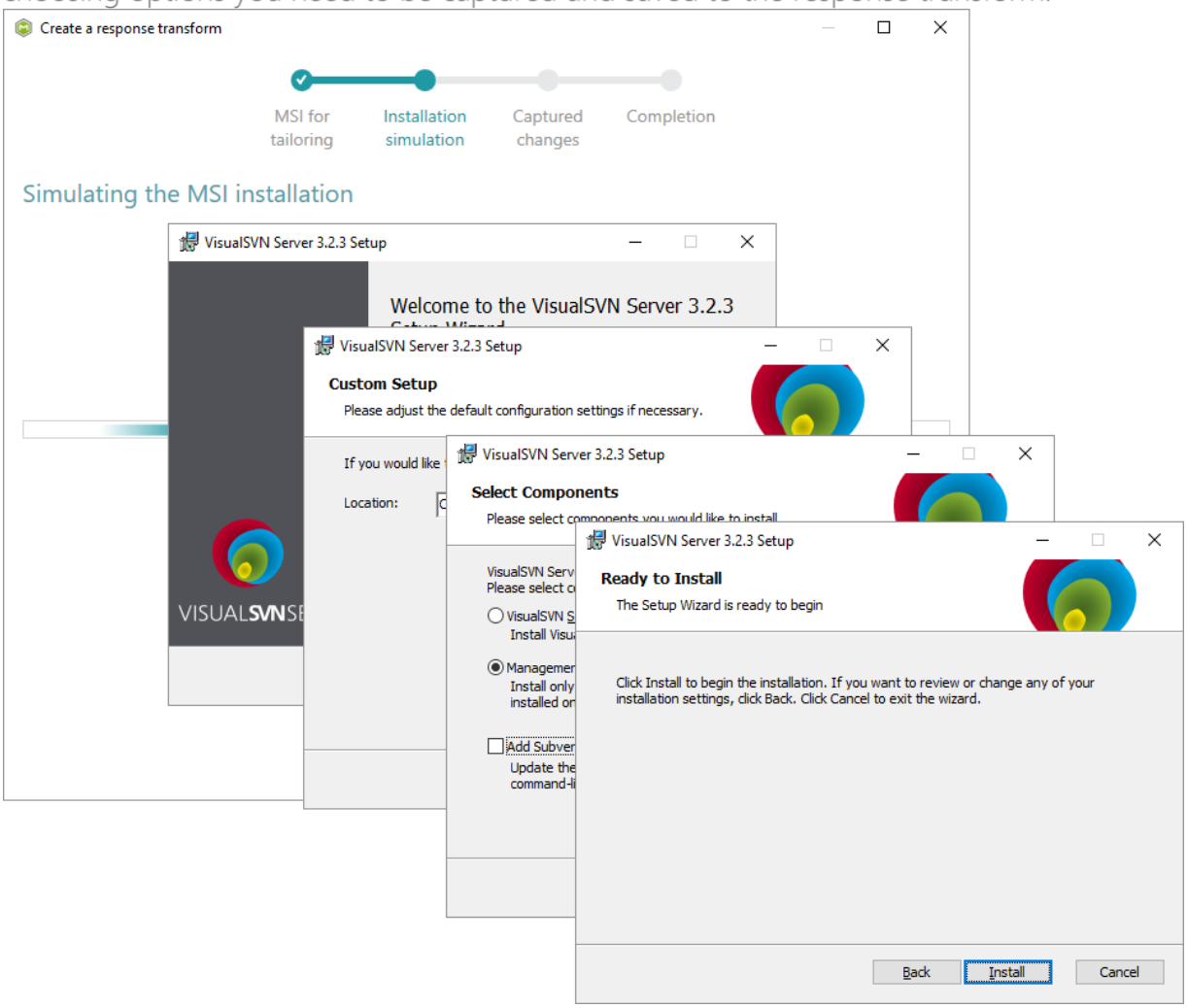
[6]. The **Apply transform (optional)** and **Command-Line options (MSI properties only)** fields are optional and they allow applying additional MST and setting MSI properties before the MSI tailoring. Select the **Ignore MSI launch condition** option to skip all condition from the LaunchCondition MSI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



[7]. Click **Next** to start the MSI installation simulation.

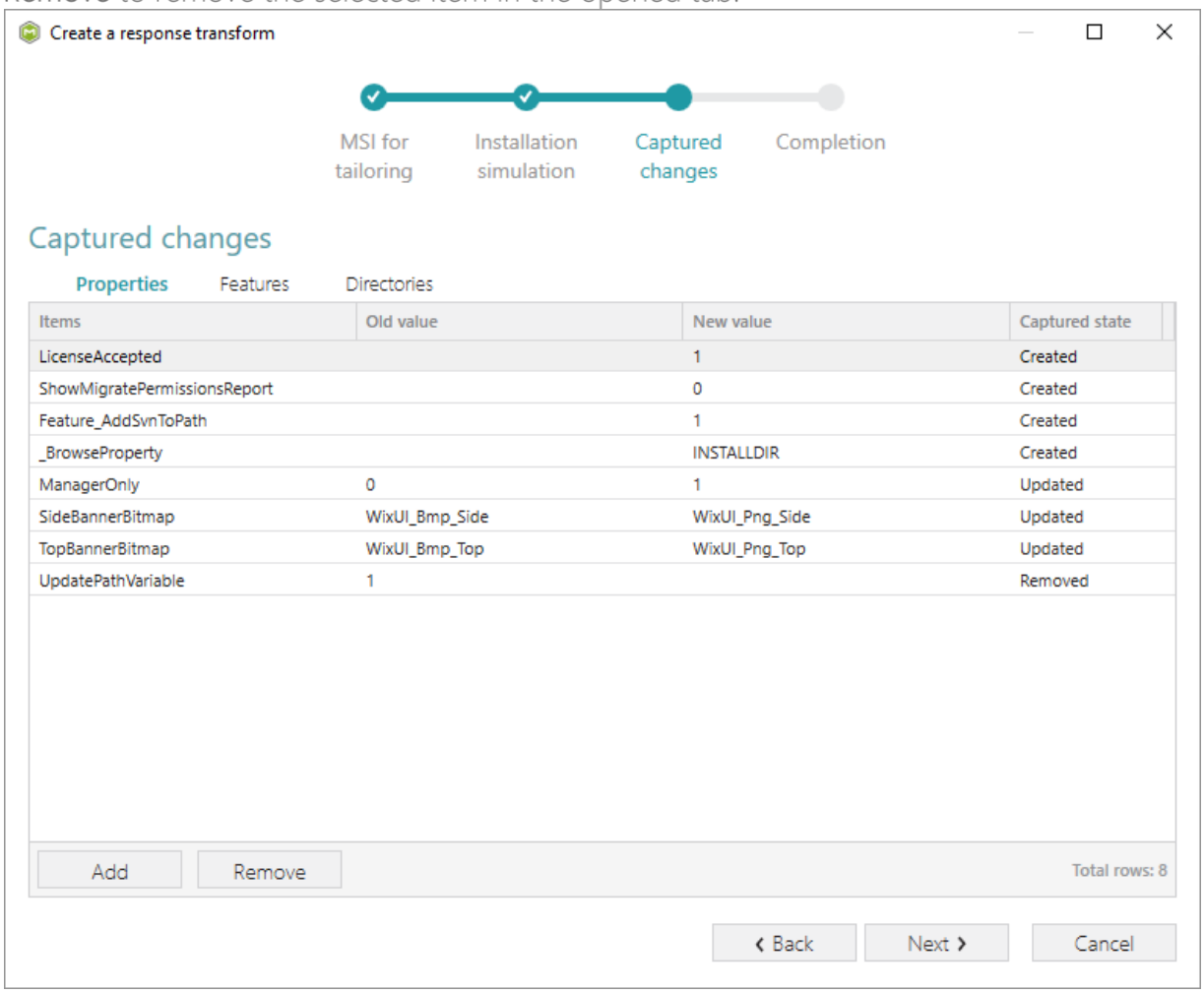


[8]. Complete the simulated MSI installation following the Windows Installer dialogs and choosing options you need to be captured and saved to the response transform.

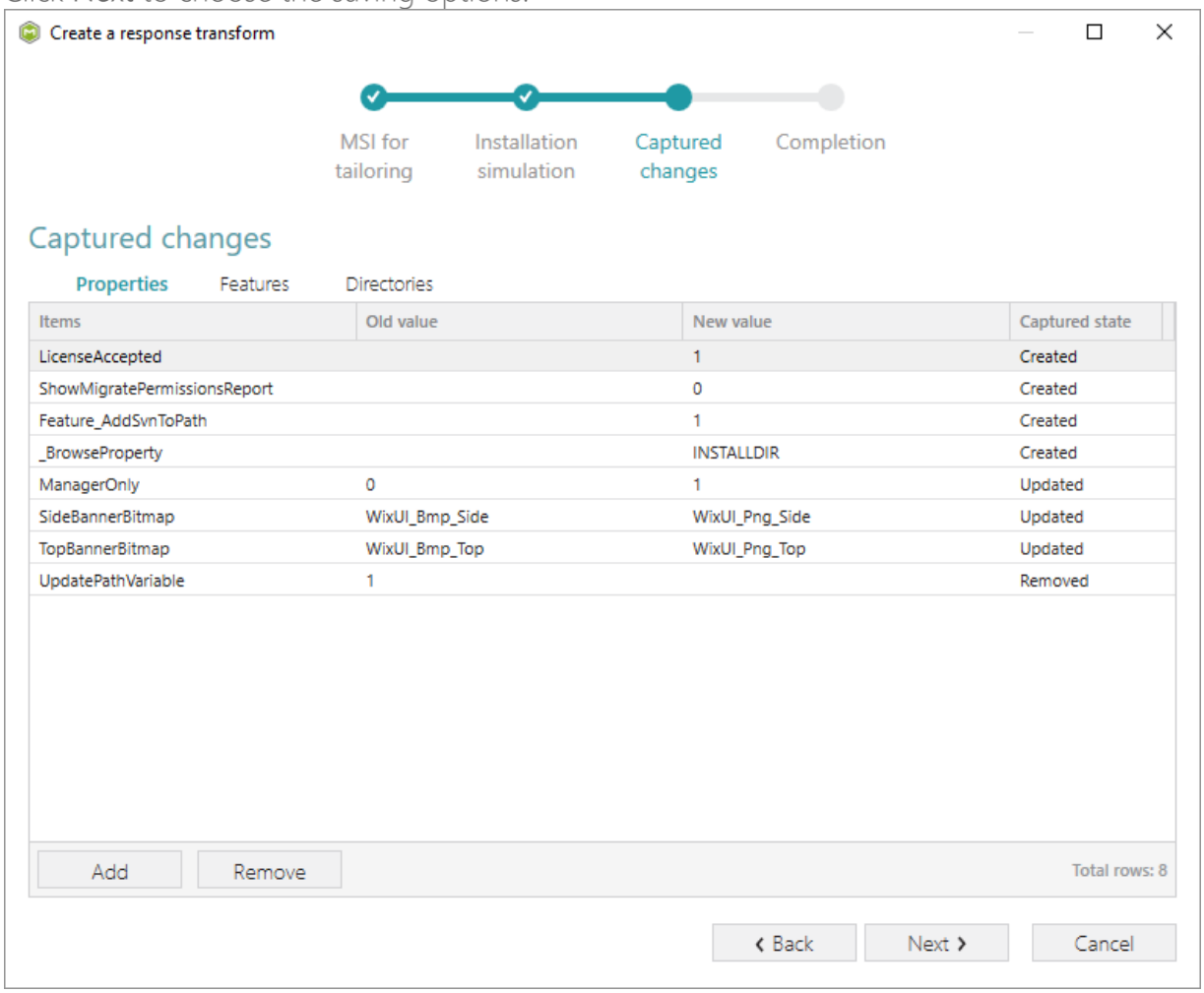



[9]. Review captured information, which will be saved to the response transform (MST) file, by opening the **Properties**, **Features** and **Dicrectories** tabs. Click **Add** to create new or click

**Remove** to remove the selected item in the opened tab.

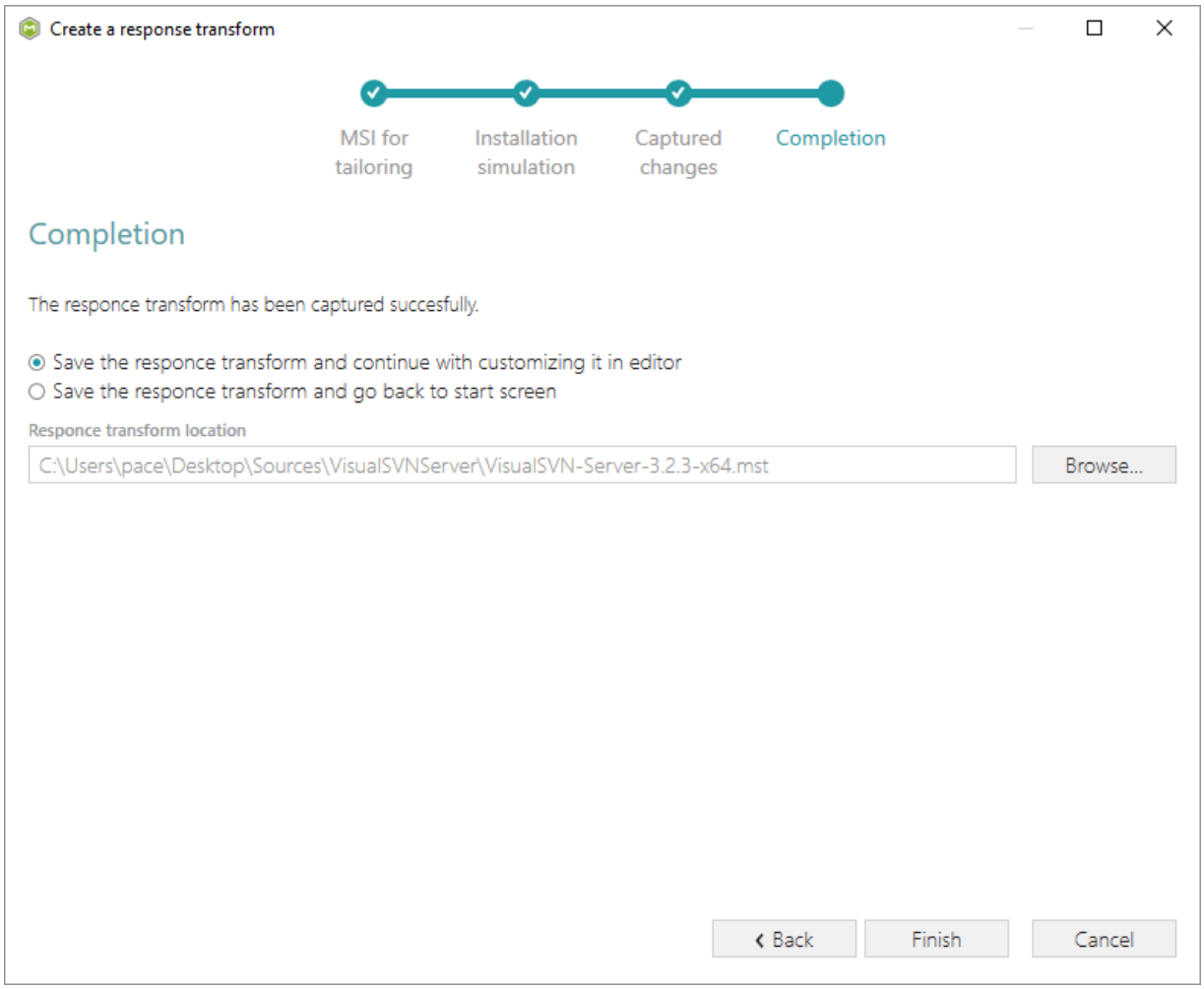


[10]. Click **Next** to choose the saving options.




[11]. 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

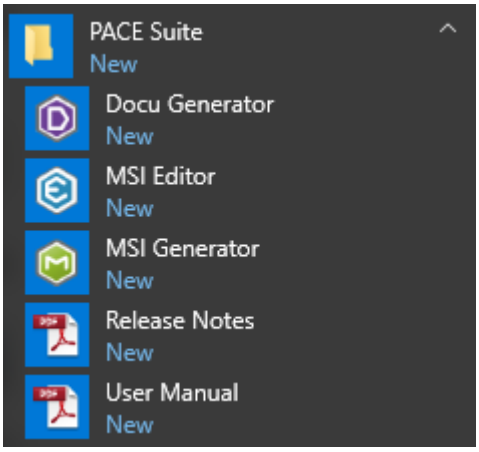
transform.



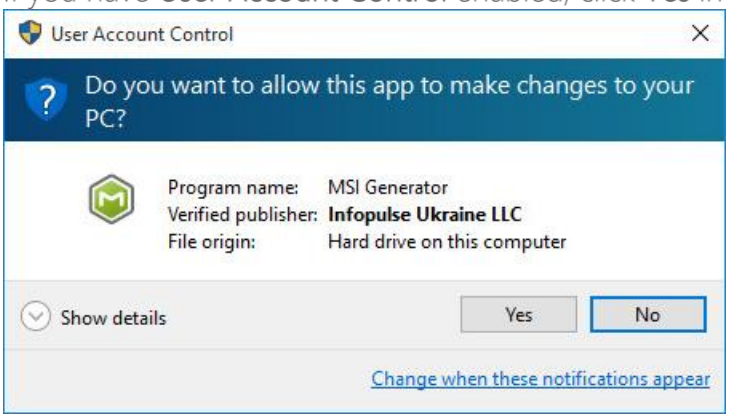

## 3.2.3 Capture System Changes to MST

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 file system and registry permission changes.

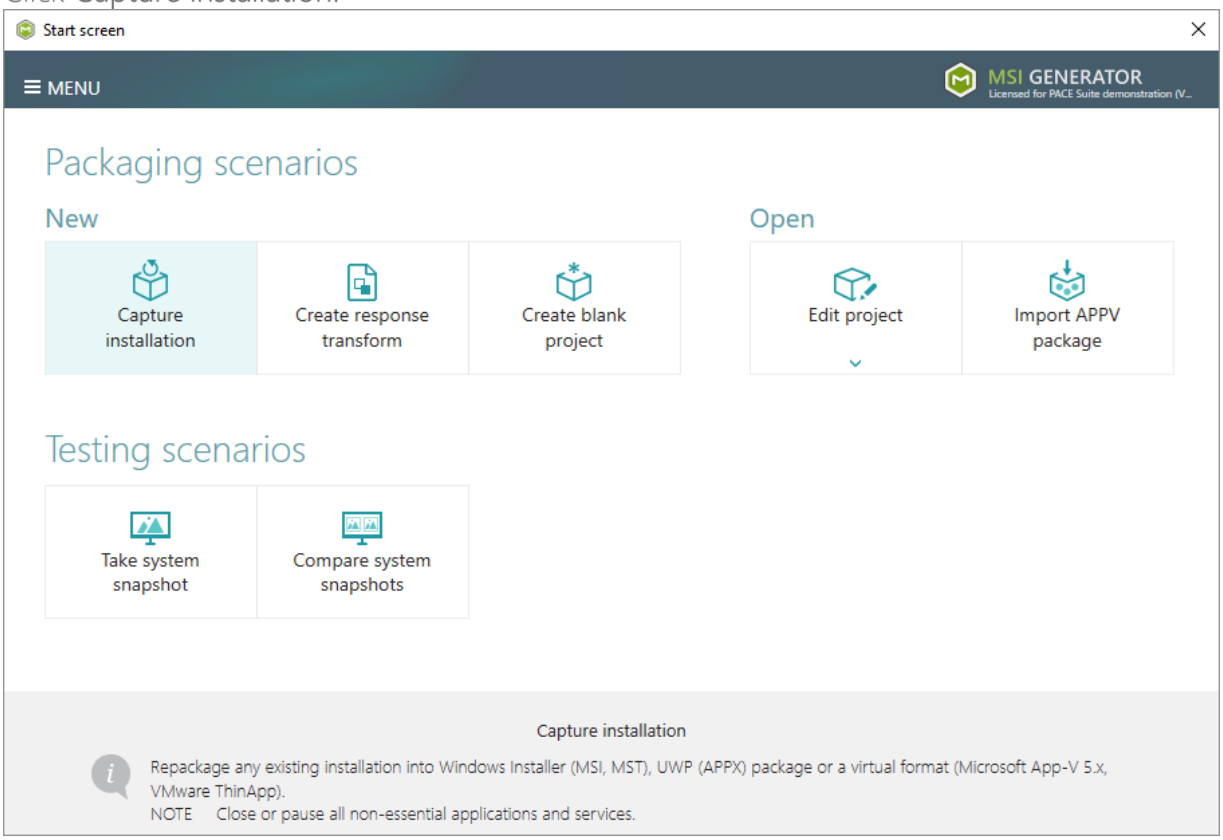
- Launch MSI Generator from the desktop or the start menu shortcut.



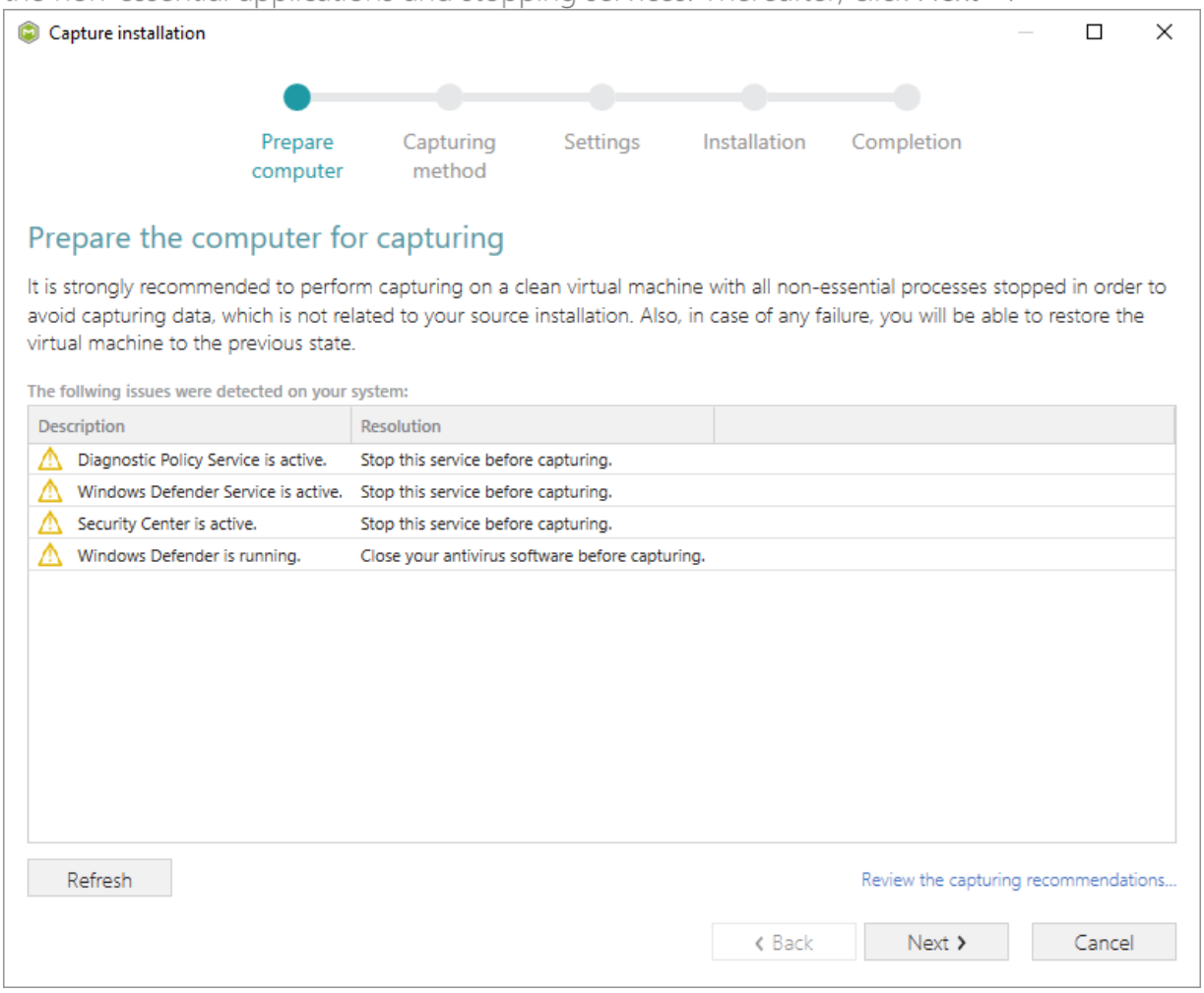
- If you have User Account Control enabled, click Yes in the opened window.



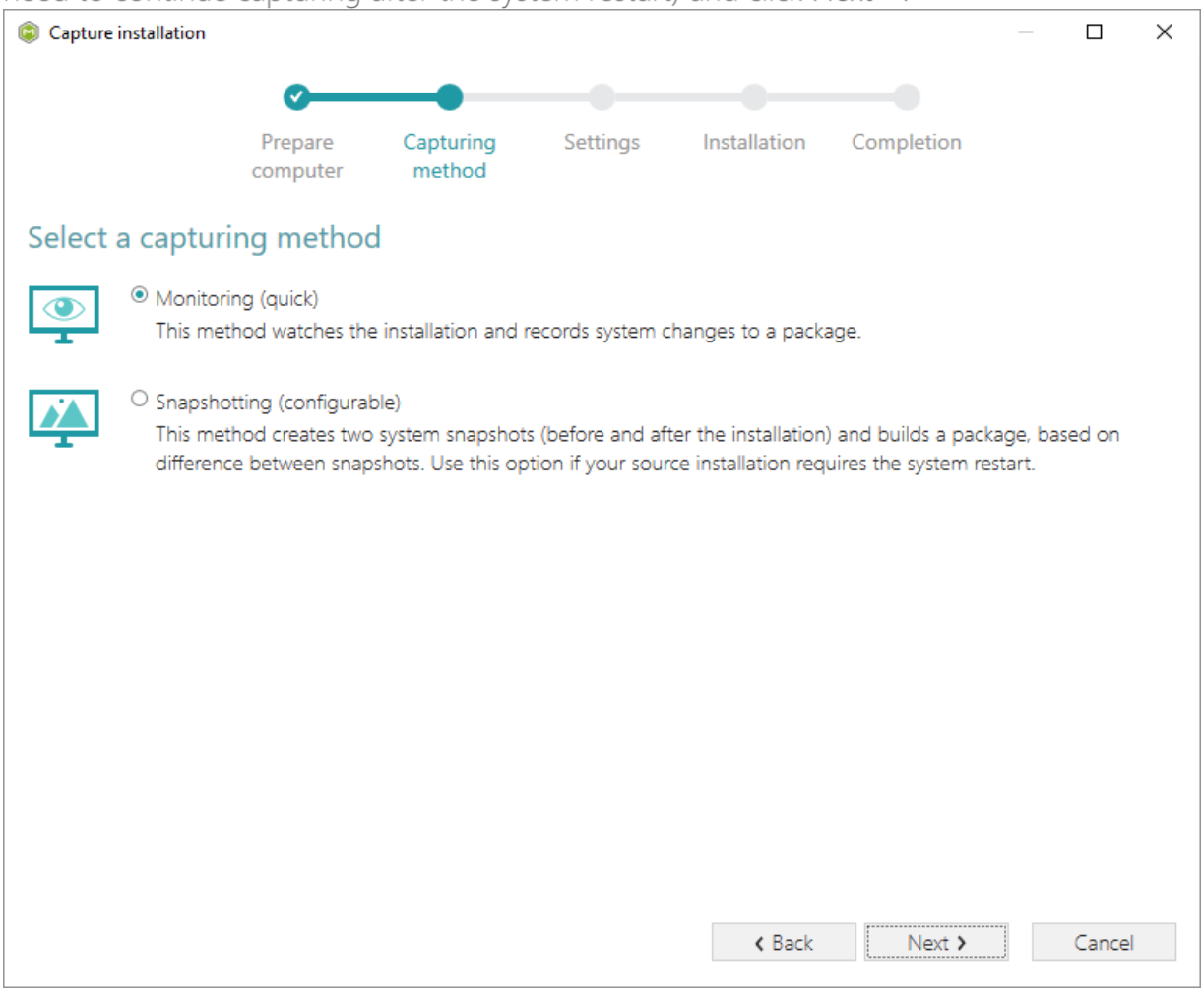

- Click Capture installation.



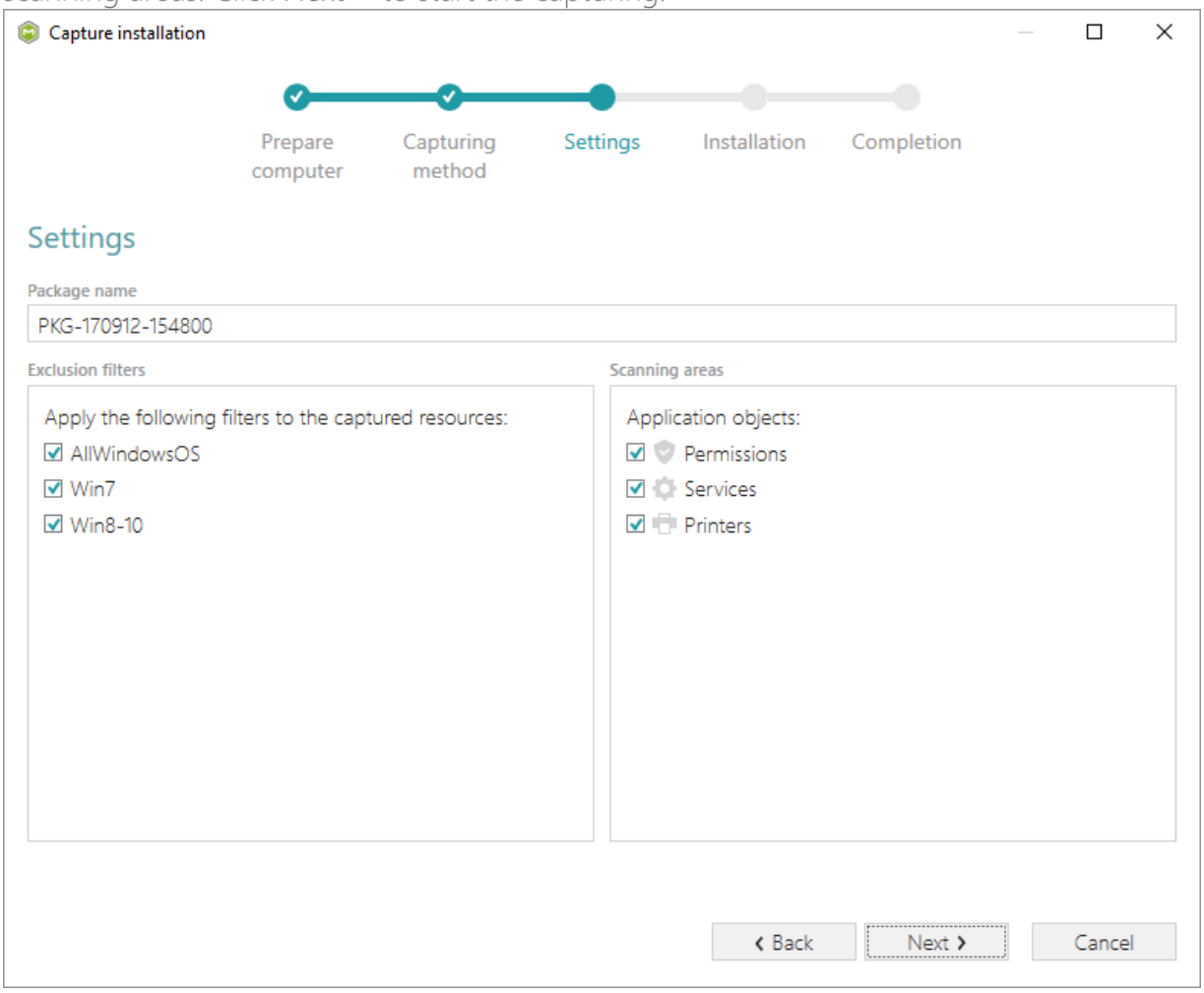
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



- 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 >**.

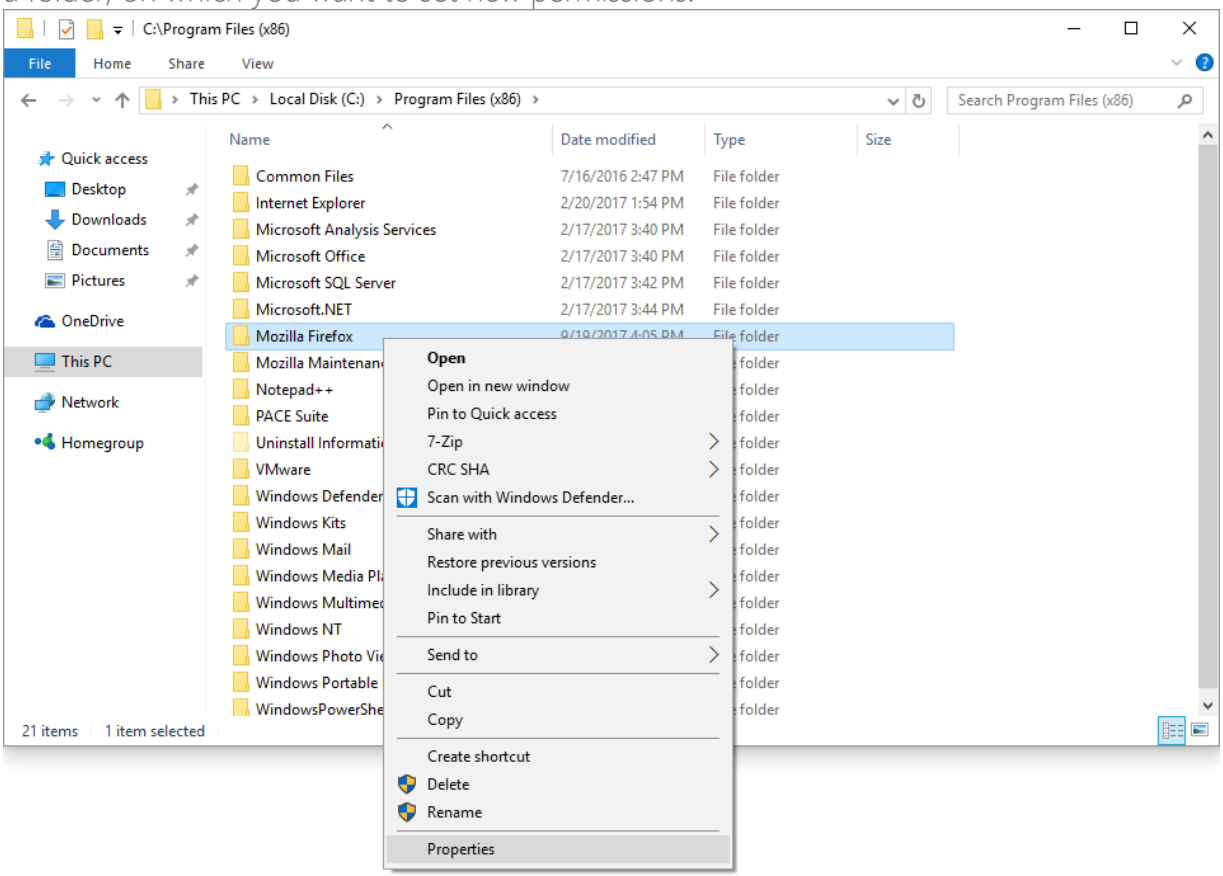


- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.

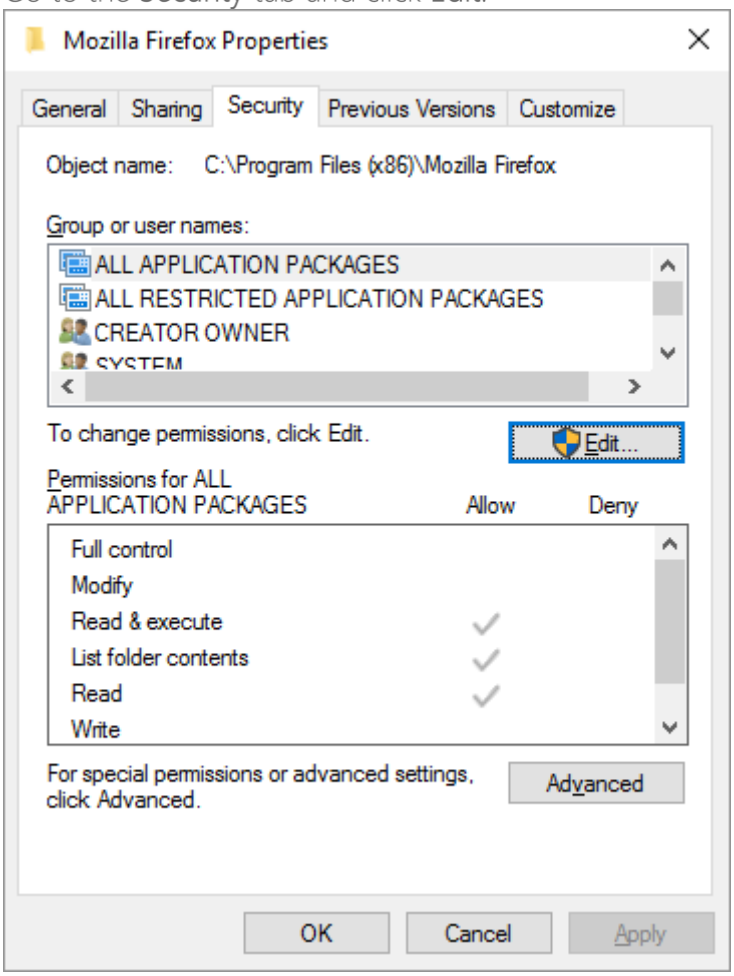



- Now you can make any changes to the file system and registry, which you want to capture and include to the transform (MST) file. For instance, you can create new or copy-paste existing files, import REG file to the system registry, changes permission settings, or launch the installed application in order to capture the necessary application configurations, like disabling updates and so on.
- If you additionally need to change the default permission settings on a folder or a file and include them to the package, select Properties from the Windows Explorer context menu of

a folder, on which you want to set new permissions.

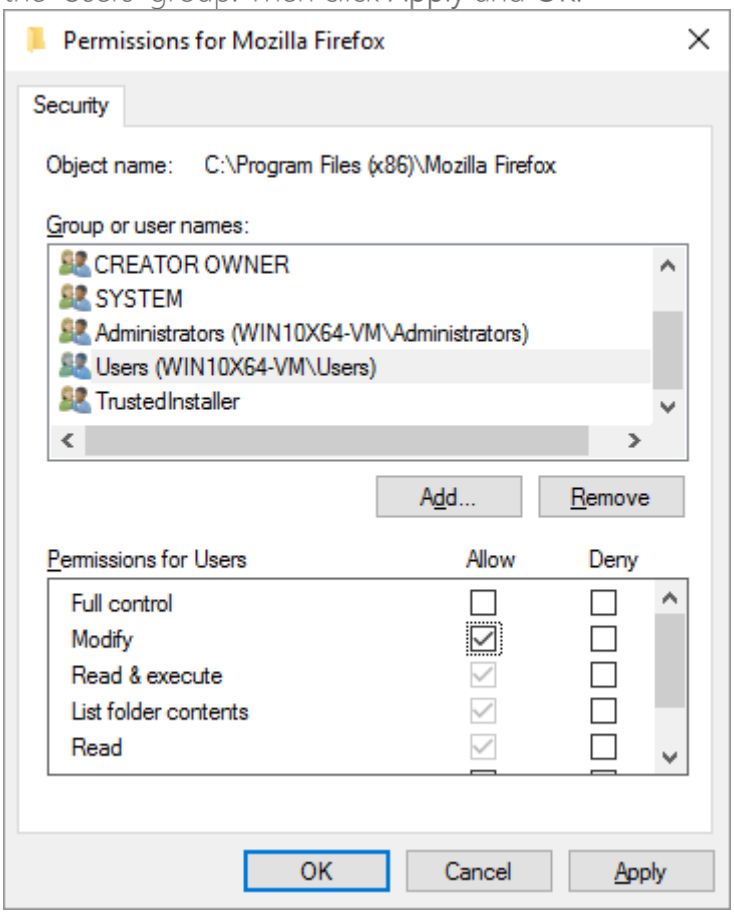


- Go to the Security tab and click Edit.

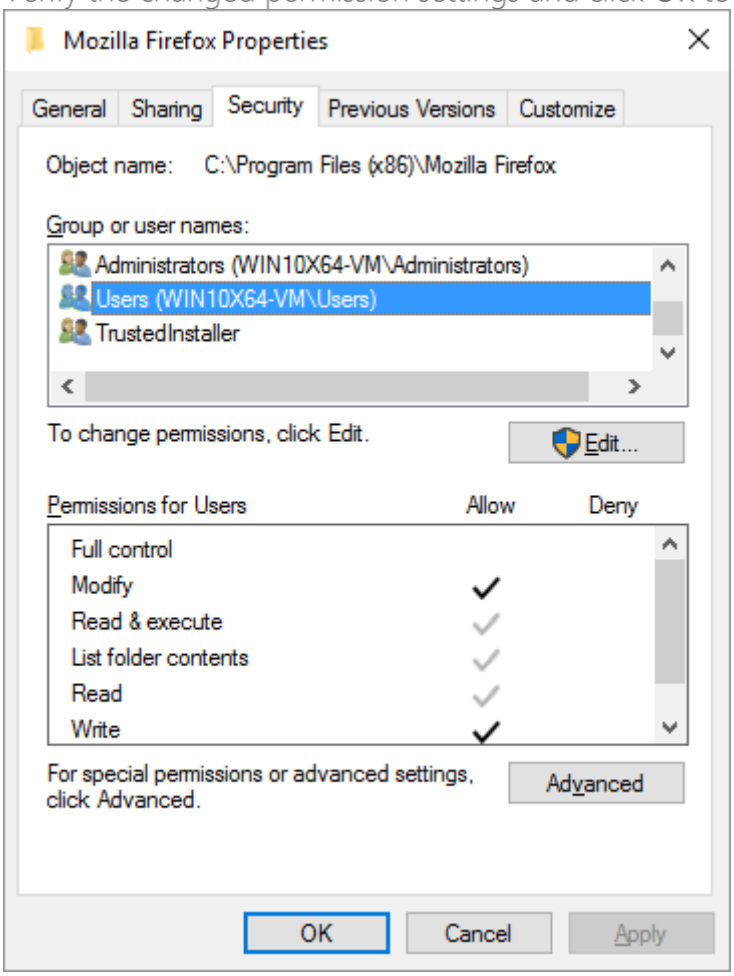



- Select a group or a user, for which you want to change the permissions and then select the necessary access rights. For example, lets set the 'Write' and the 'Modify' access rights for

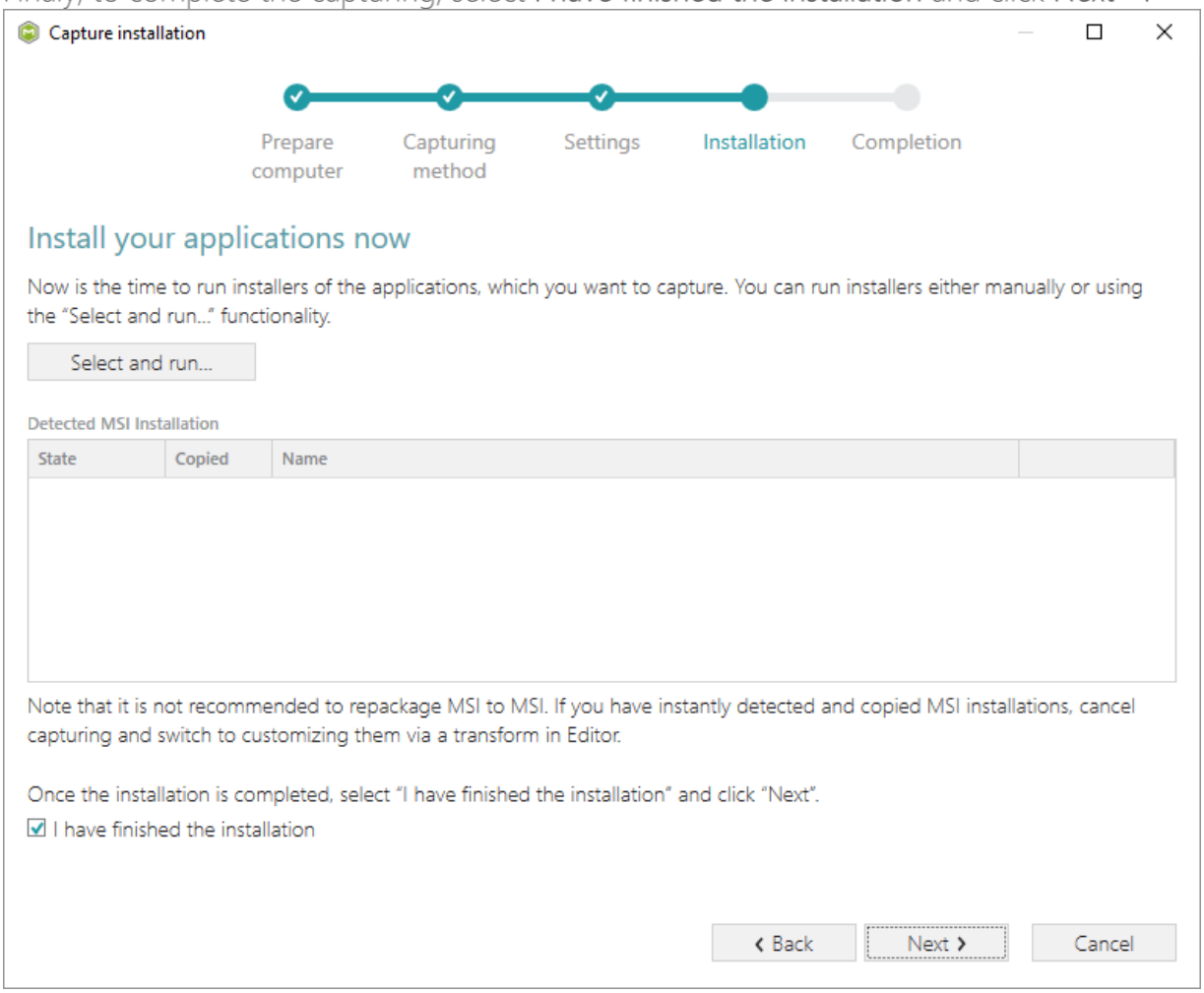
the 'Users' group. Then click Apply and OK.



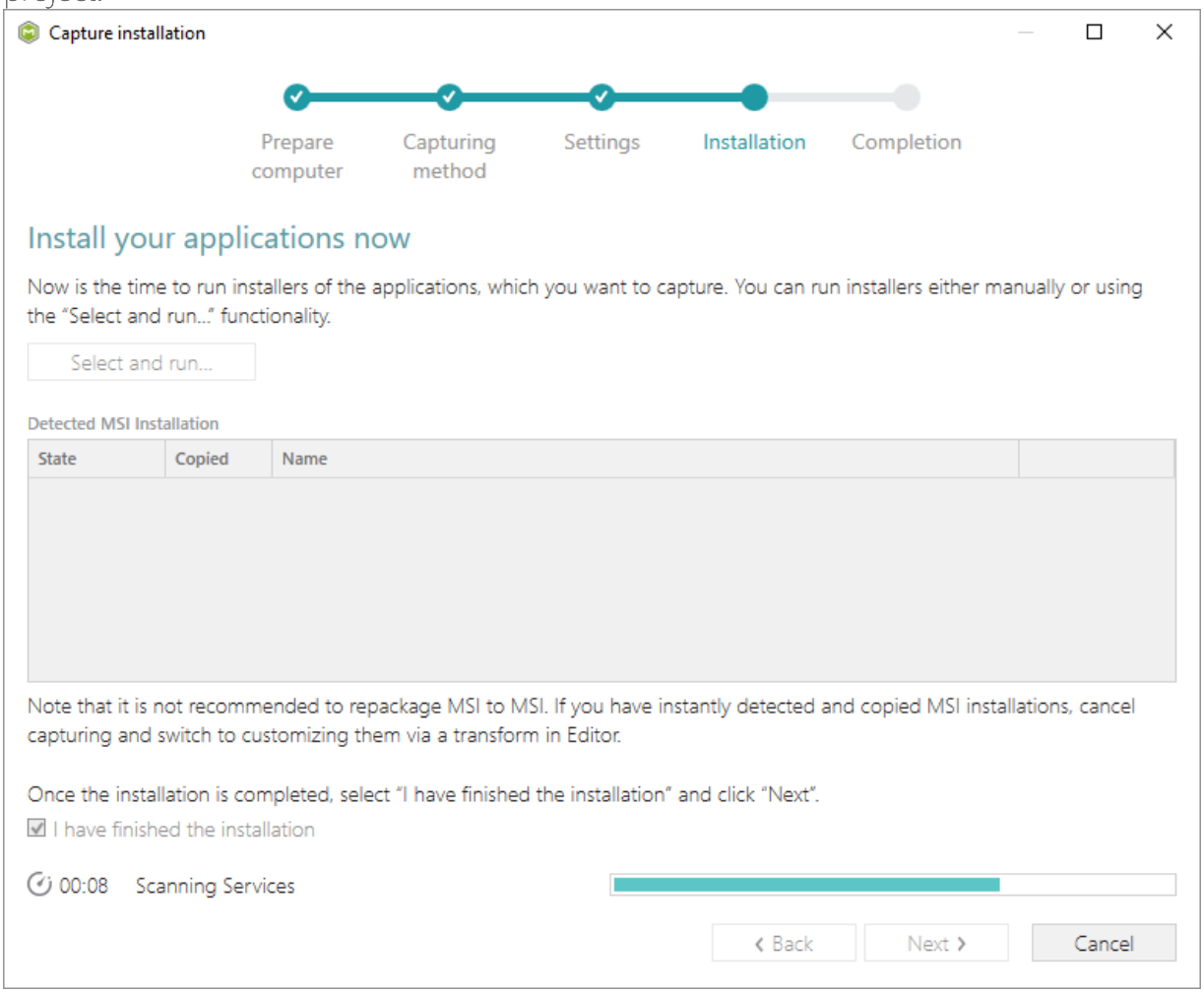
- Verify the changed permission settings and click OK to close the window.



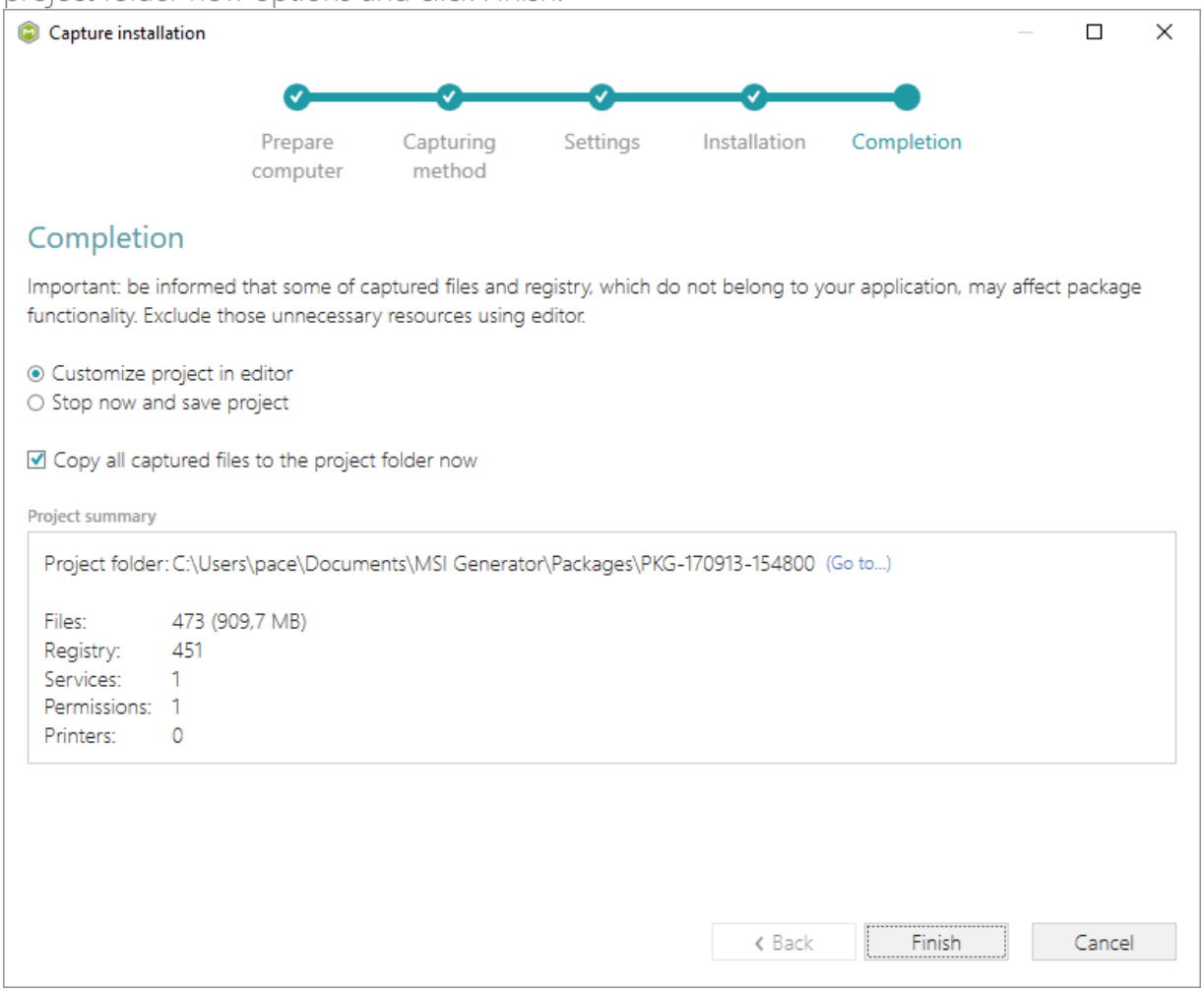
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



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

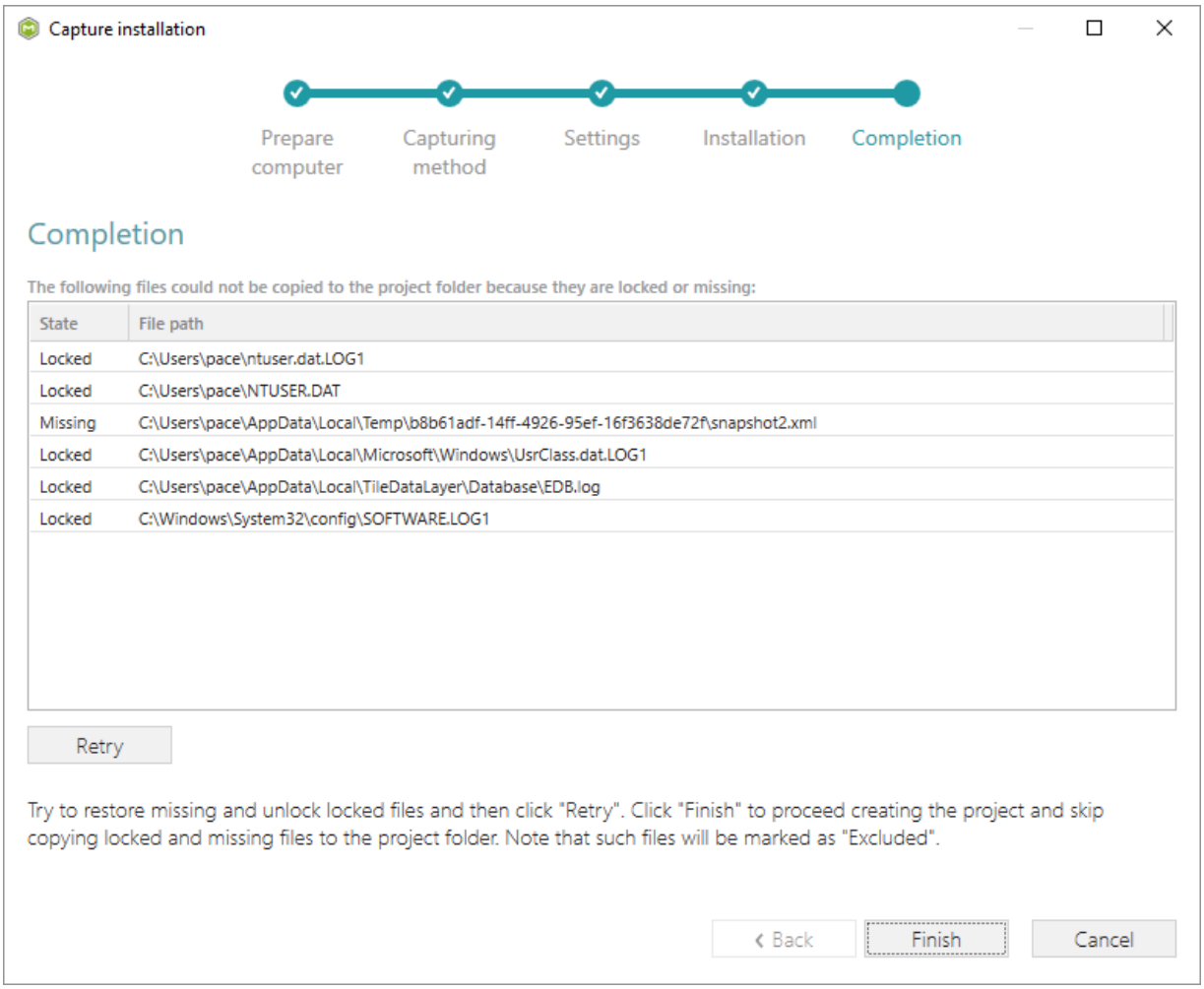


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



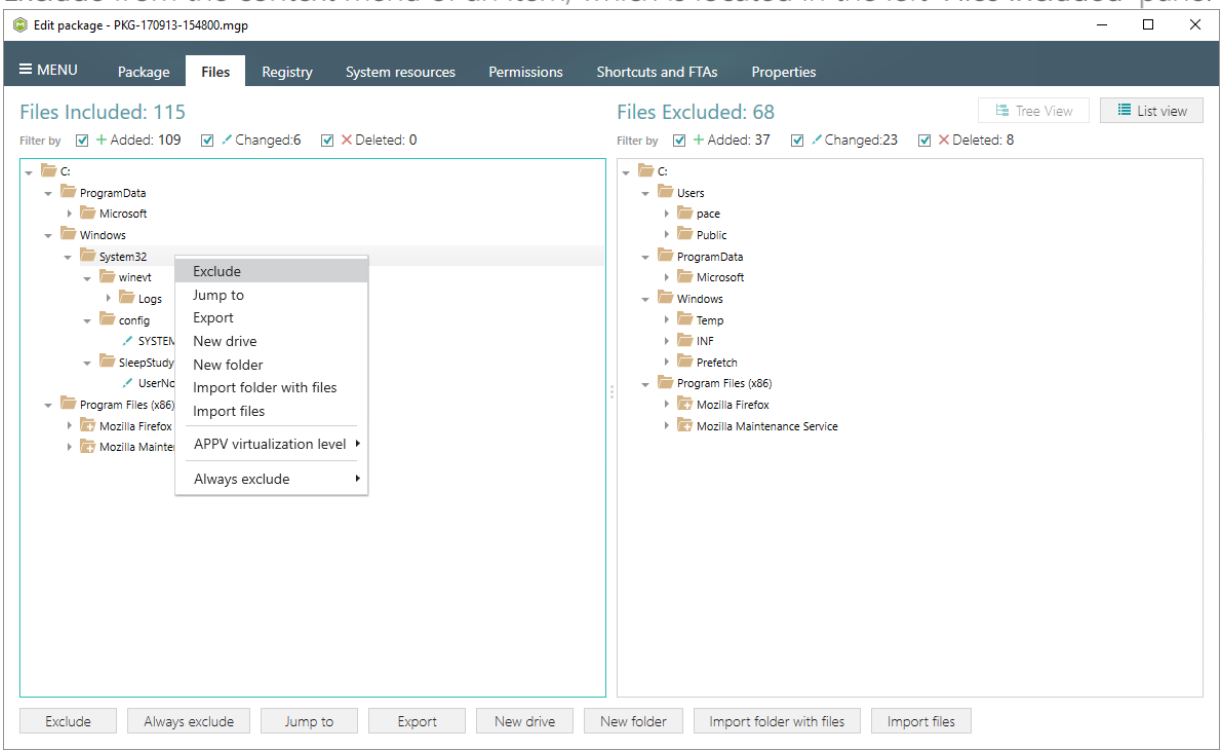

- The following dialog displays captured files, which could not be copied to the project folder because they do not exist anymore or locked by the system or by an application. Try to resolve these issues and then click Retry. Click Finish to skip copying the missing and locked

files and mark them as Excluded resources.

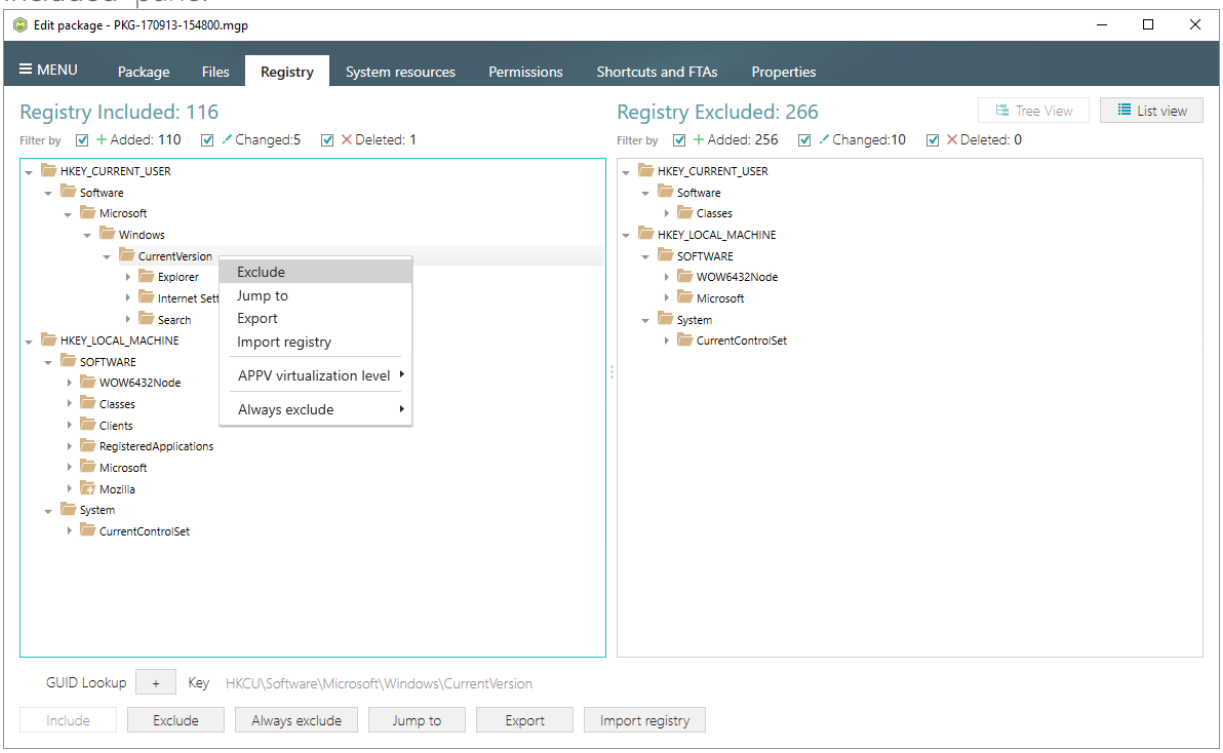



- Once the project is opened in the project editor, review the captured resources at the Files, Registry, System resources, Permissions, and Shortcuts and FTAs tabs and exclude unnecessary ones from the project. Unnecessary resources are files, registry entries, which are usually created or modified in the result of operating system work, and such resources could not be a part of your captured application. Unfortunately, there is no universal rule to discover which of captured files or registry entries should be excluded, so exclude only those ones, which almost 100% do not refer to your captured application (e.g. NOD32 antivirus files couldn't be a part of Firefox application).

- 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.

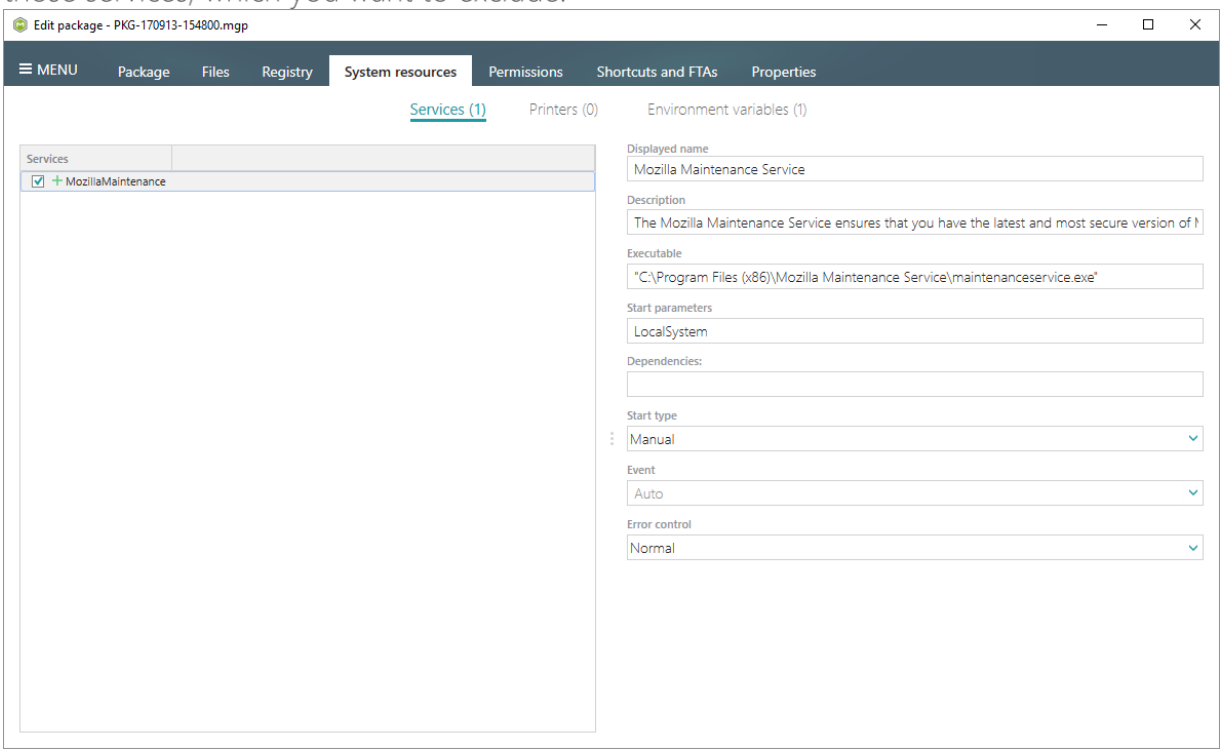



- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.

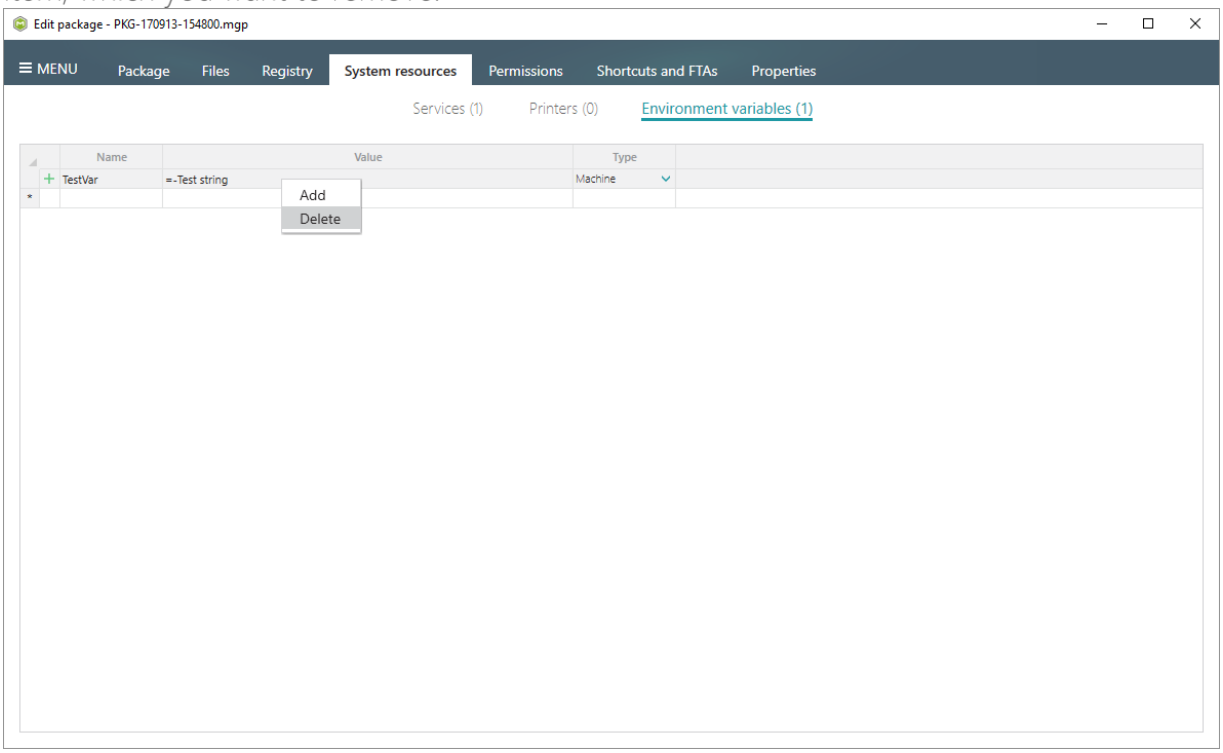



- In order to review and exclude unnecessary services, go to the **System resources** -> **Services** tab, and uncheck the checkbox, located before the service name in the list, for

those services, which you want to exclude.

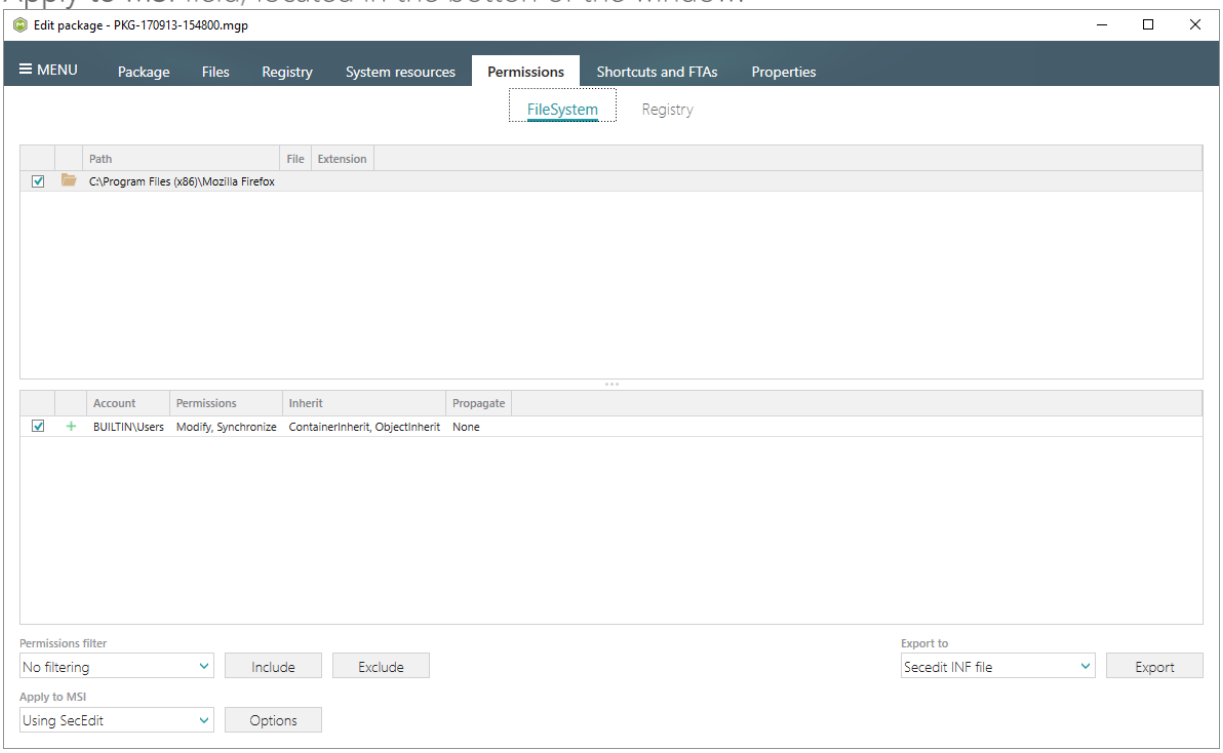



- In order to review and remove unnecessary Environment variables, go to the System resources -> Environment variables tab, and select Delete from the context menu of an item, which you want to remove.

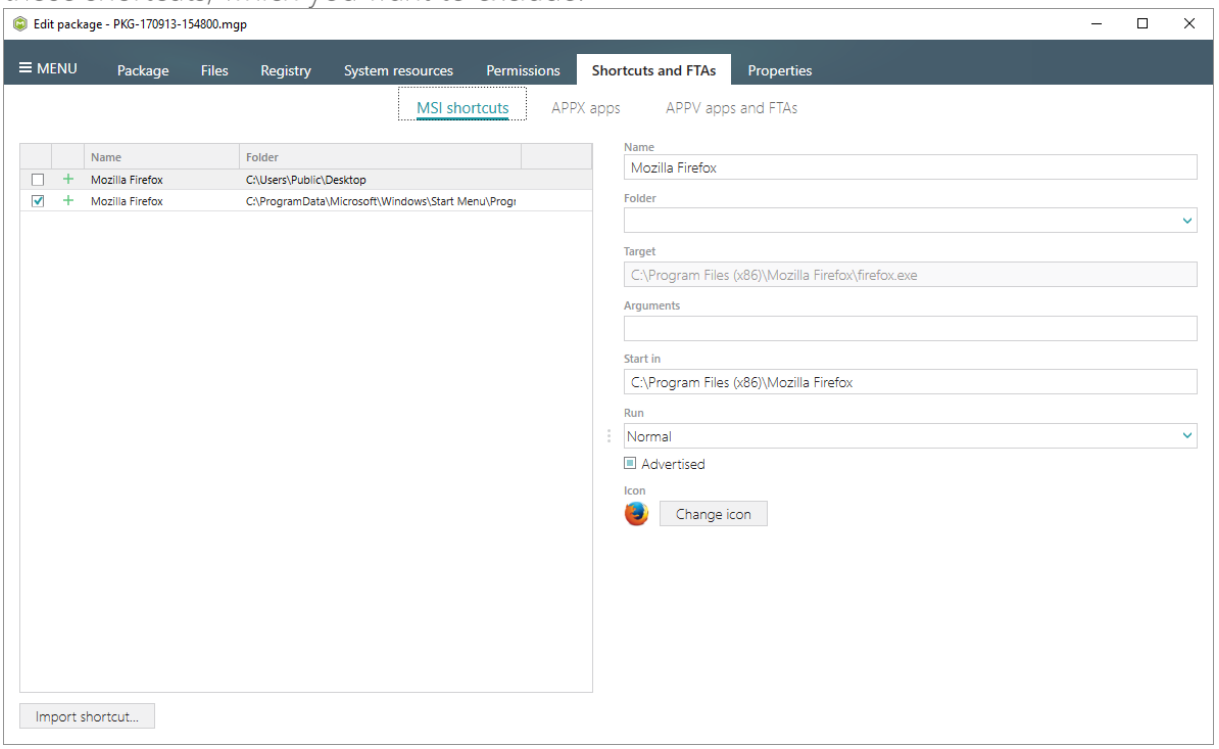



- On the Permissions tab you will find captured permissions for the file system and registry. For excluding unnecessary permission changes, either uncheck the checkbox, located before the path, for which permission changes were detected, or select Do not apply in the

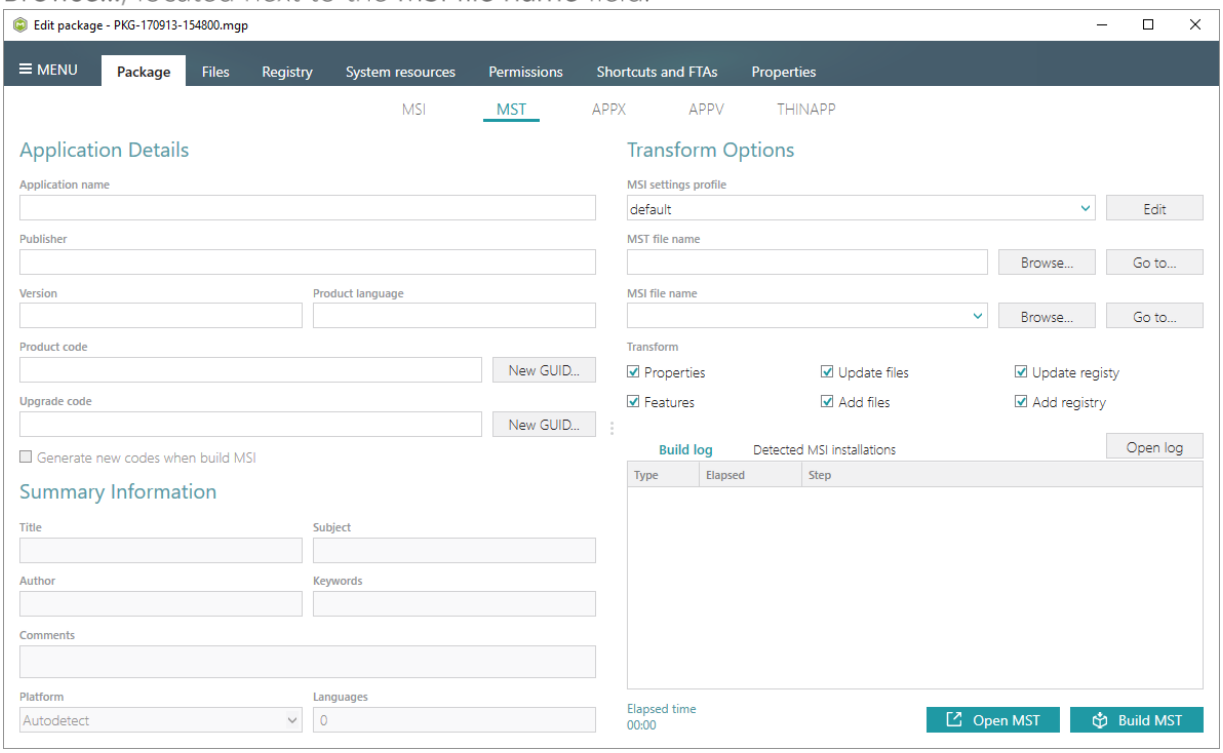
Apply to MSI field, located in the botton of the window.



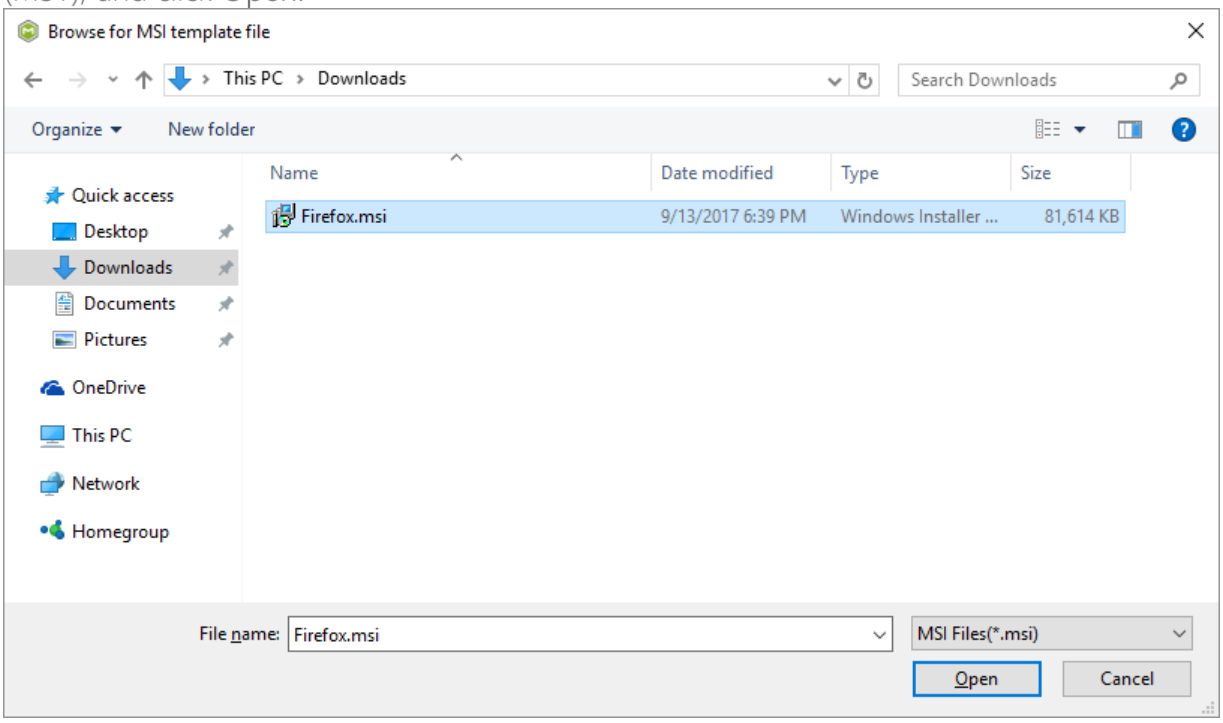

- In order to review and exclude unnecessary shortcuts, go to the Shortcuts and FTAs -> MSI shortcuts tab, and uncheck the checkbox, located before the shortcut name in the list, for those shortcuts, which you want to exclude.



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

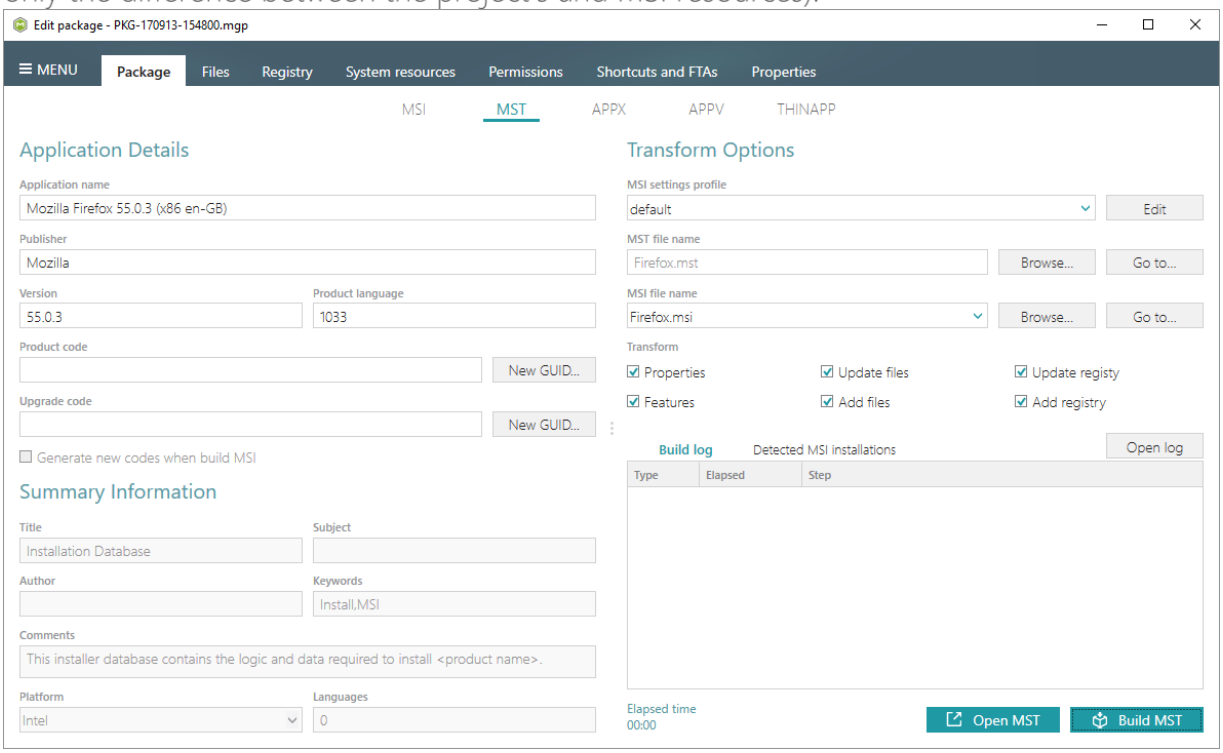



- Choose an MSI package (e.g. Firefox.msi), against to which you want to create a transform (MST), and click **Open**.

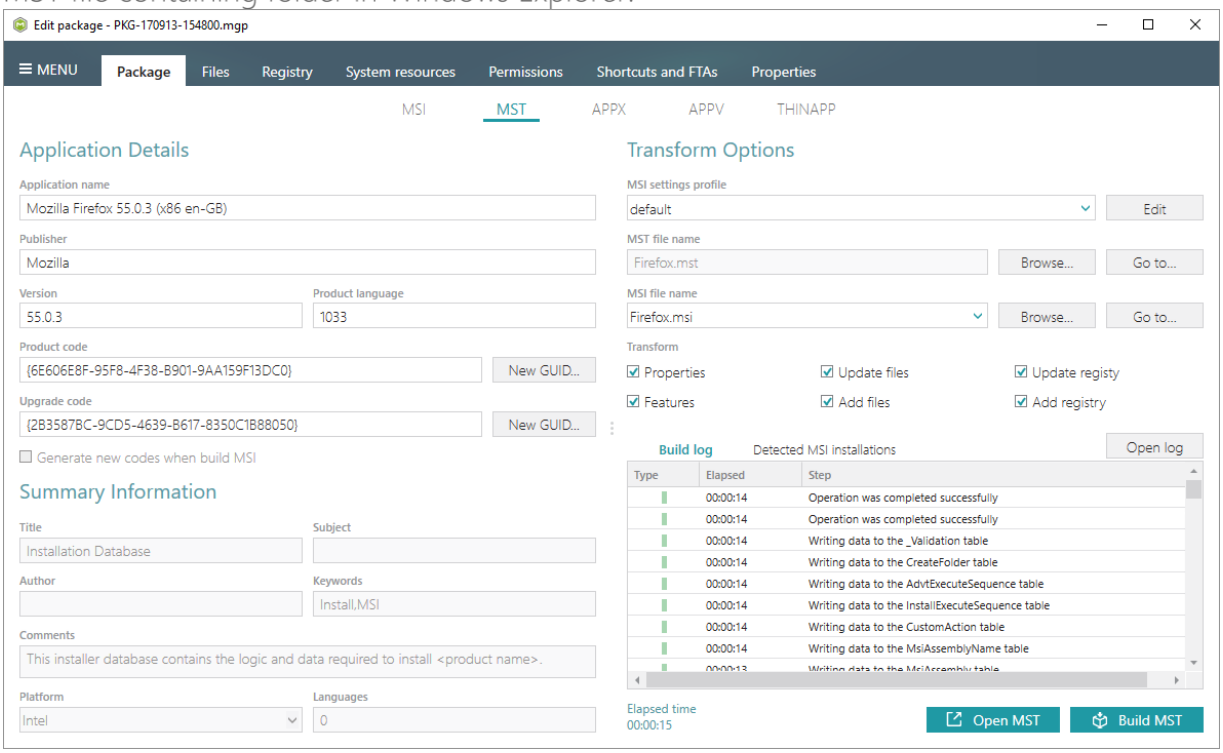



- Update Application details such as name, publisher, version, language and click **Build MST** to generate a transform (MST) file. Note that the generated MST will contain only those resources that do not exist in the selected MSI package (in other words, MST will contain

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:

- Create MSP Based 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 to MSP, described in section 3.3.2
  Save changes, made to the opened MSI database, as a Patch (MSP) package.

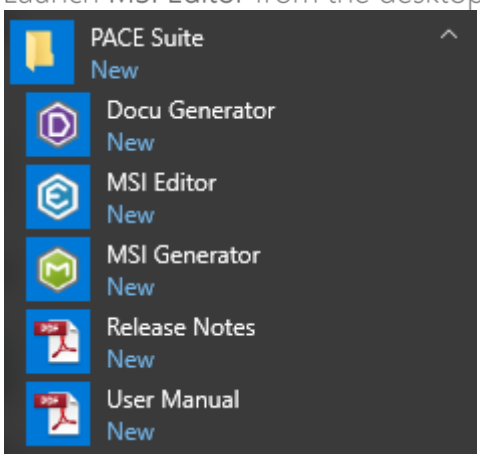
NOTE Adhere to the following restrictions during creating a Patch package:

- Do not move files from one folder to another.
- Do not move files from one cabinet to another.
- Do not change the order of files in a cabinet.
- 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 Create MSP Based On Difference

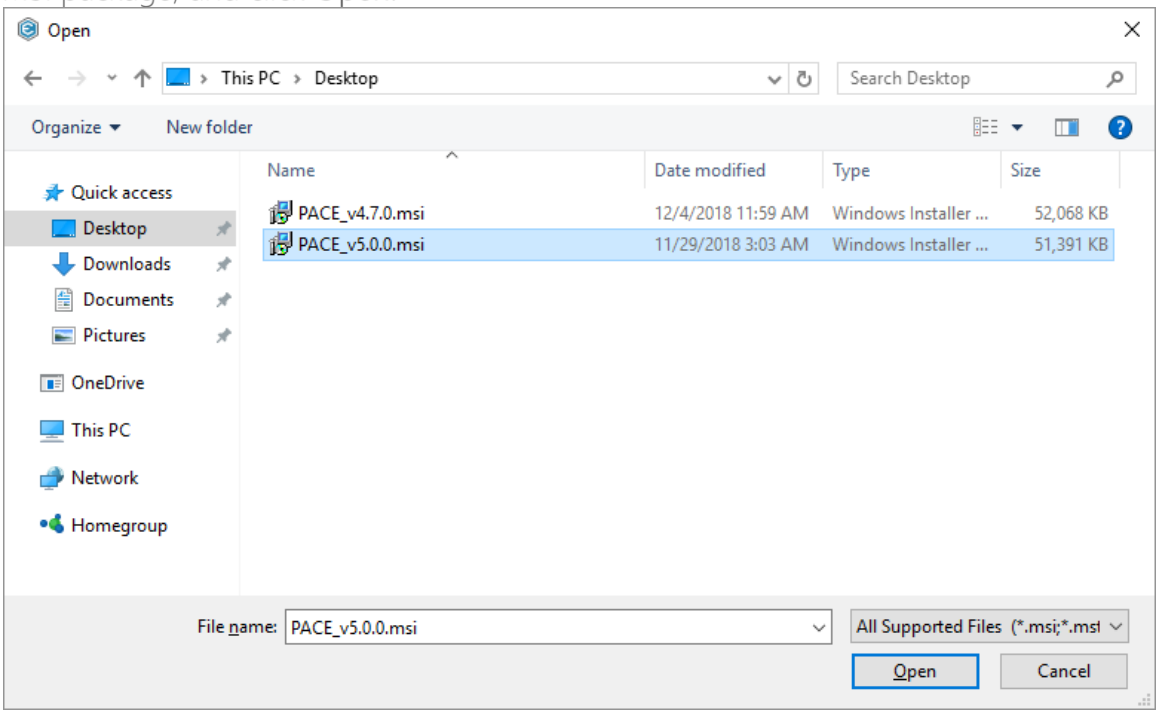
Create a Patch (MSP) package, based on differences between two MSI packages: recent and old ones.

[1]. Launch MSI Editor from the desktop or the start menu shortcut.

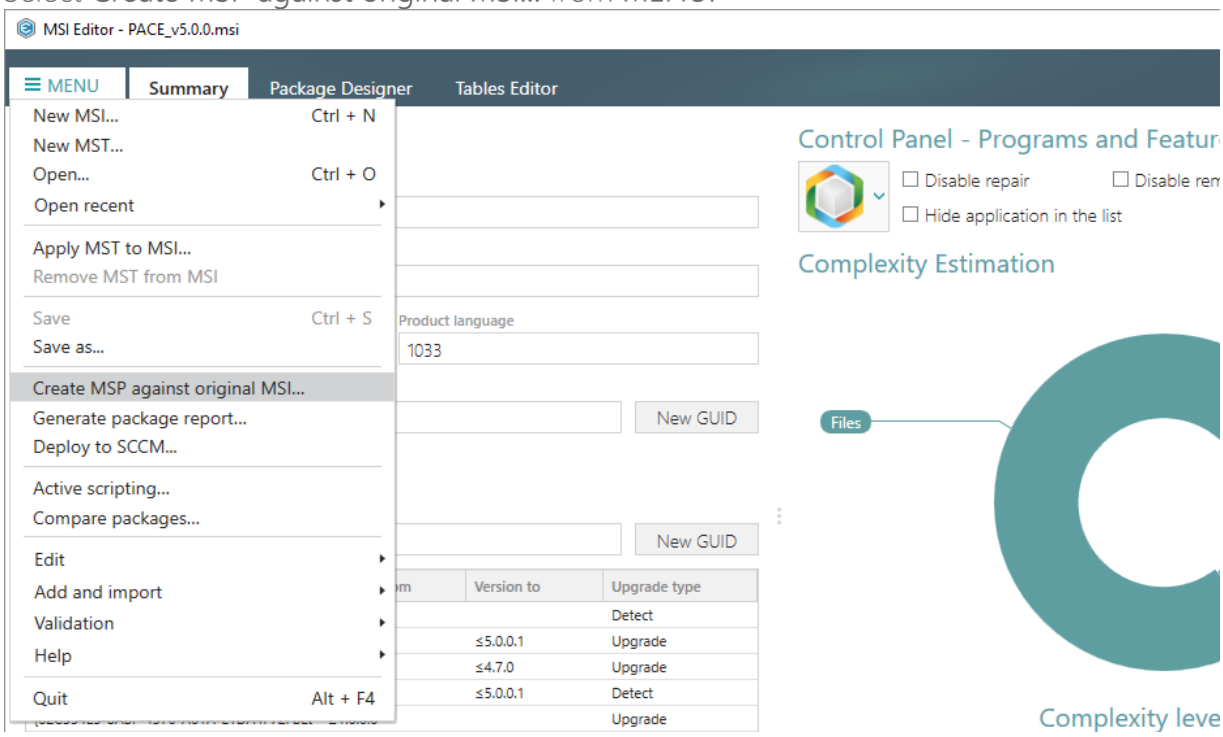


[2]. Select Open... from MENU.

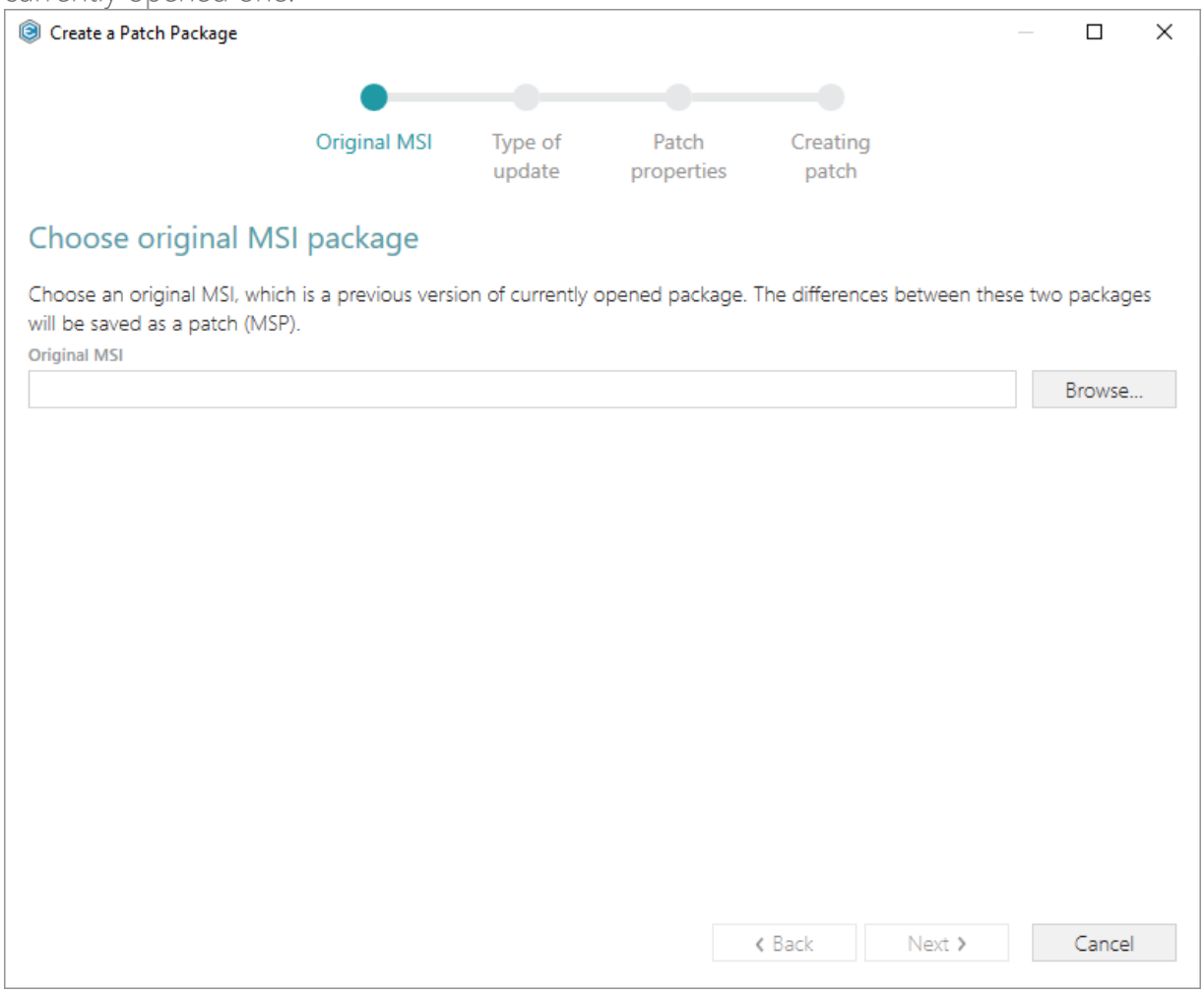
[3]. Choose a recent MSI package, containing new resources in comparison to the previous MSI package, and click Open.



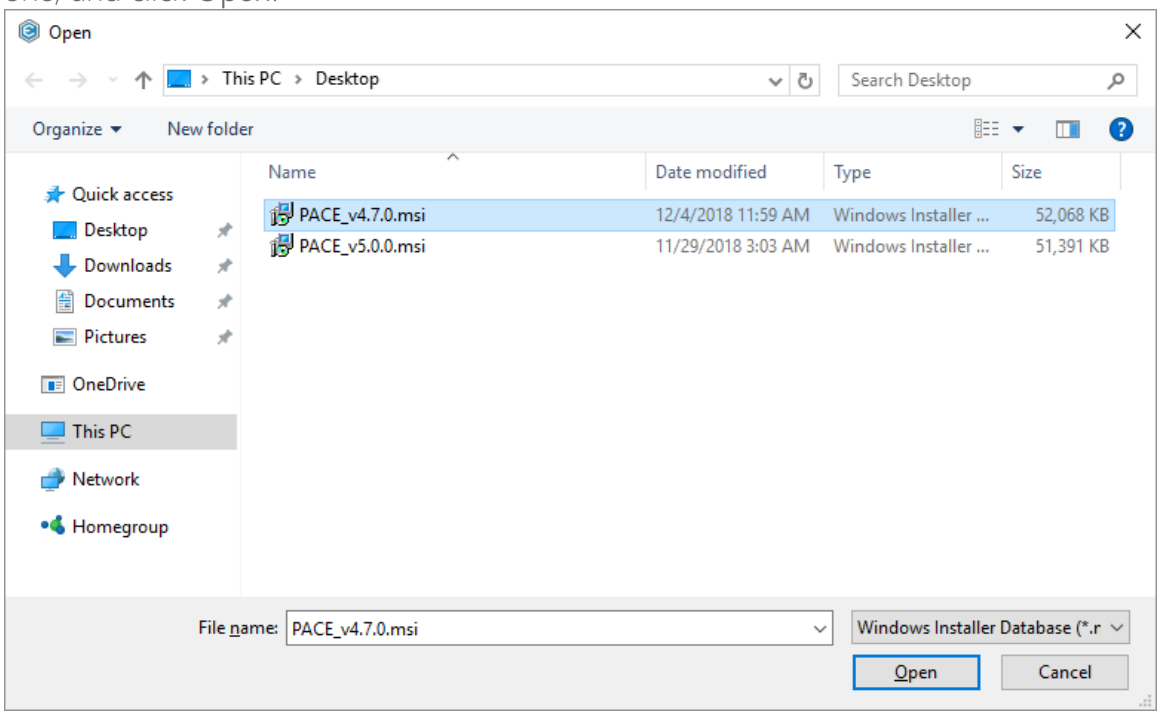
[4]. Select Create MSP against original MSI... from MENU.



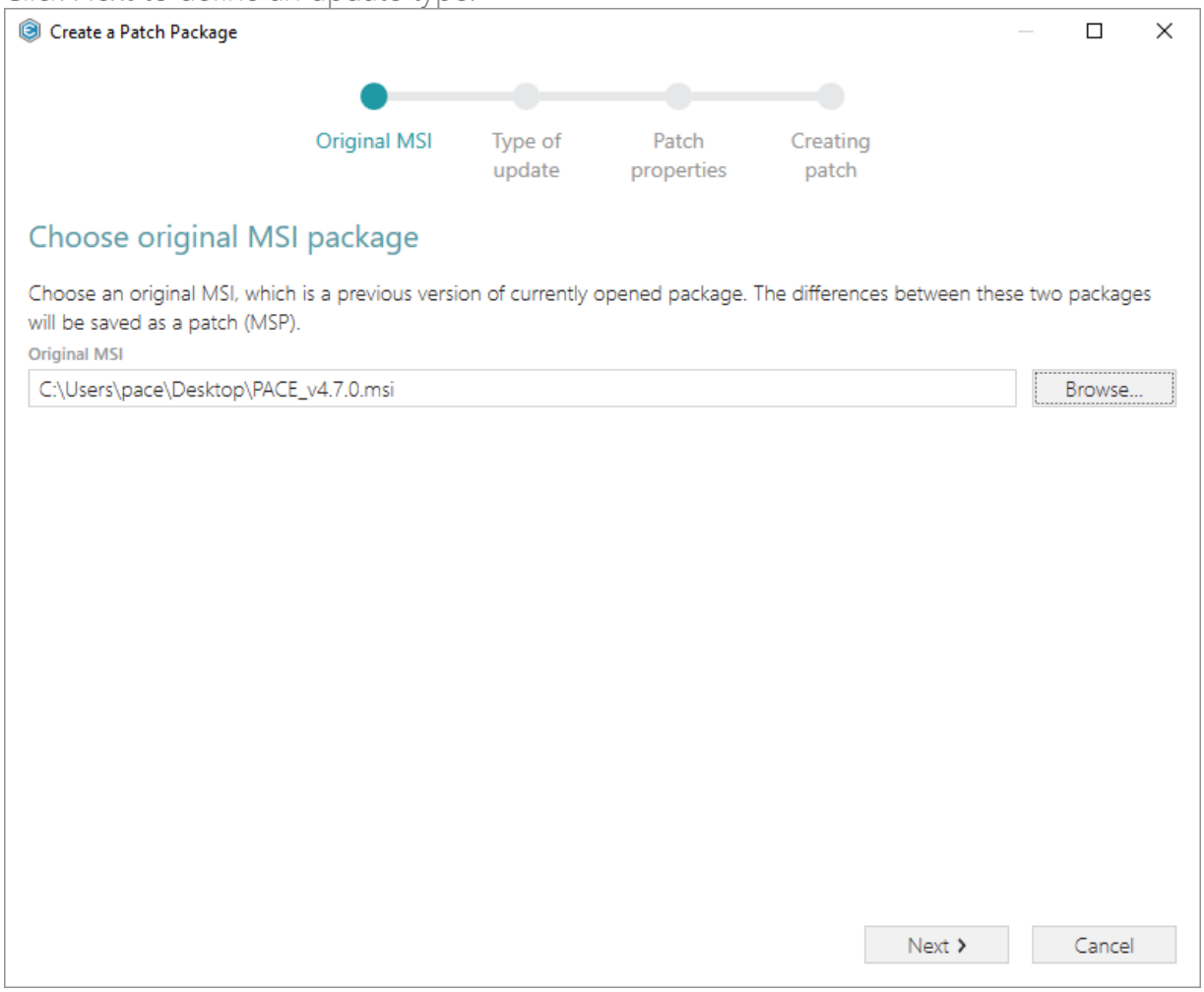

[5]. Click Browse... to choose an original MSI package, which is a previous version of the currently opened one.



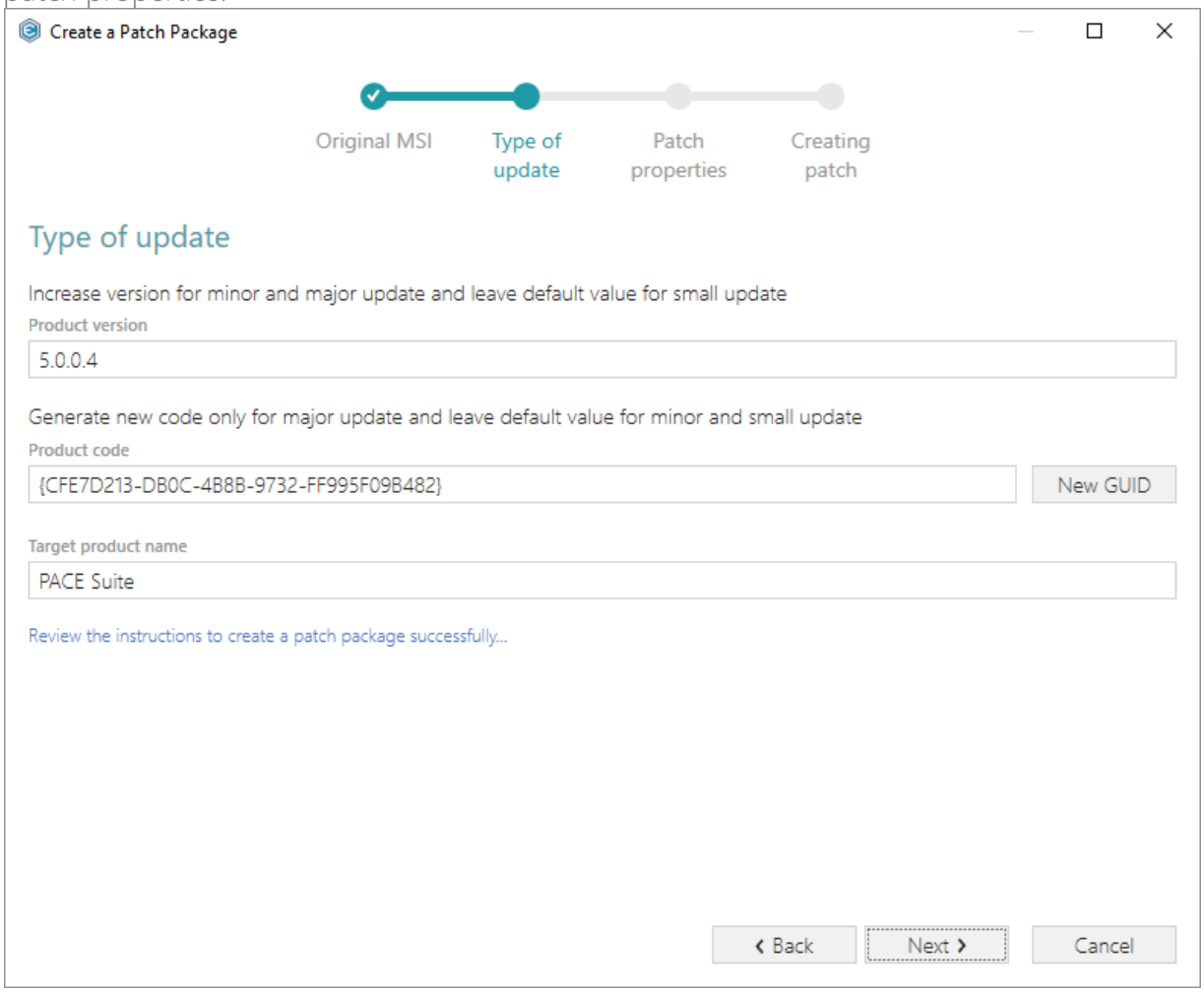
[6]. Choose an original (old) MSI package, which is a previous version of the currently opened one, and click **Open**.



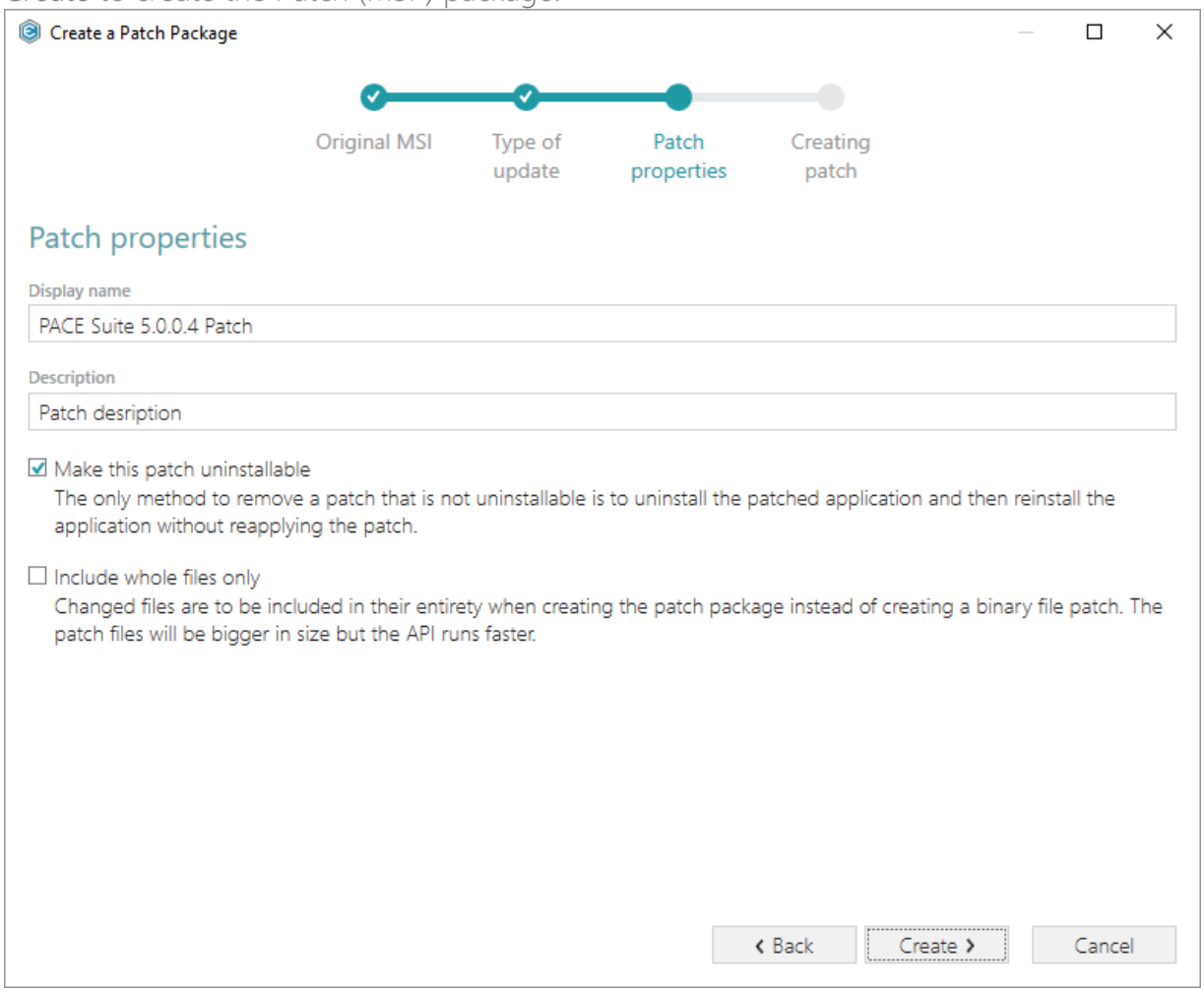

[7]. Click **Next** to define an update type.



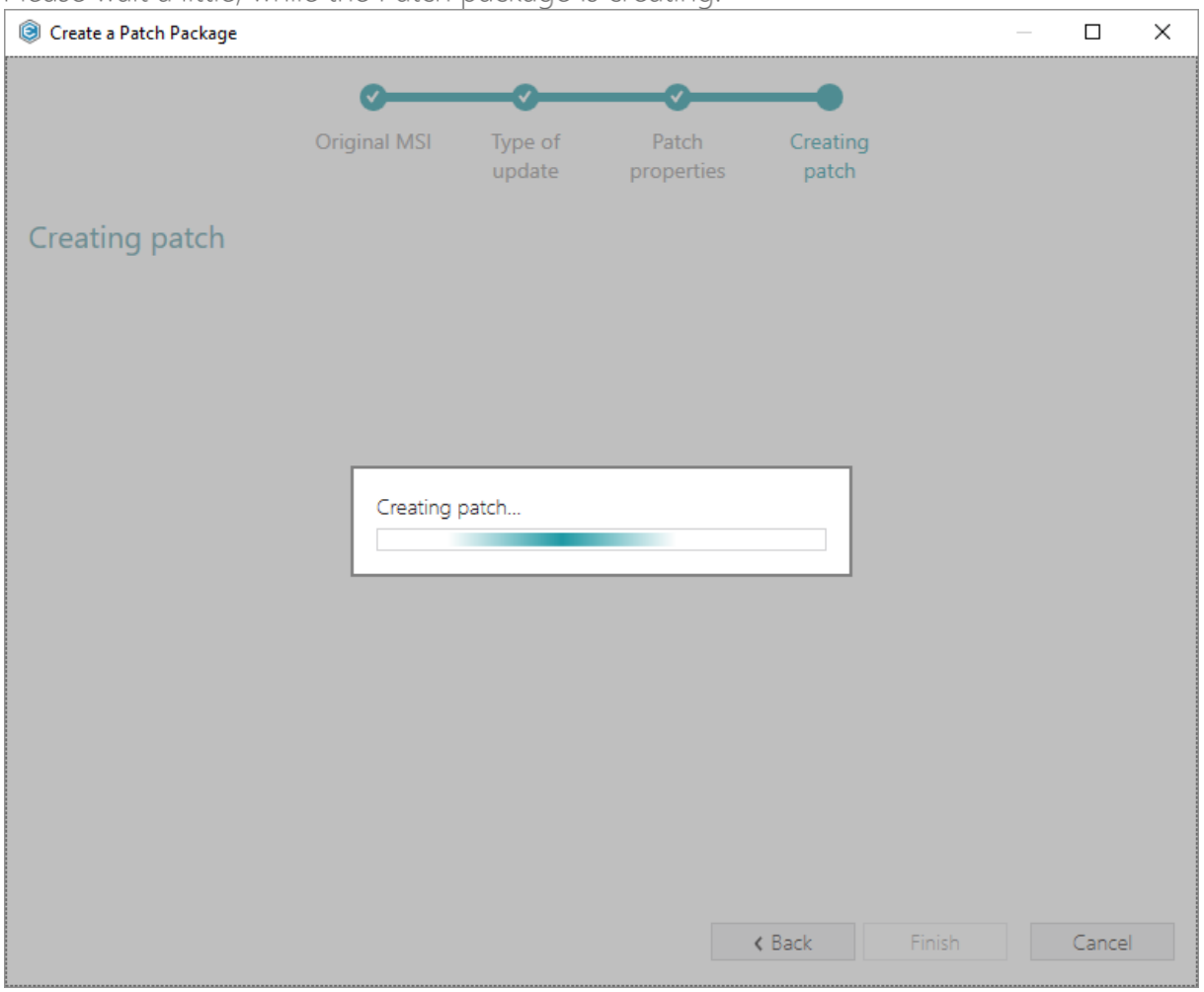
[8]. Increase a Product version value for the minor and major update and leave default one for the small update. Generate new Product code only for the major update and leave default one for the minor and small update. If needed, update a Target product name in order to change the Product Name of the updated application. Click Next to select a patch properties.



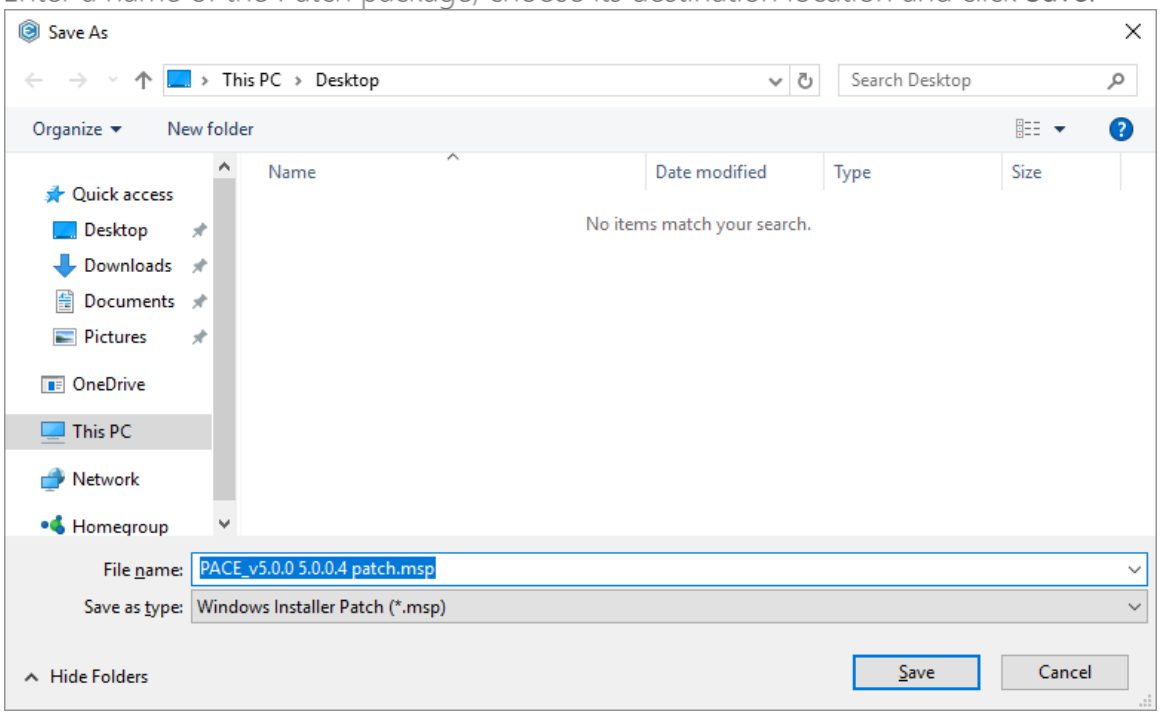
[9]. Update Patch **Display name**, **Description** and select the necessary Patch properties. Click **Create** to create the Patch (MSP) package.



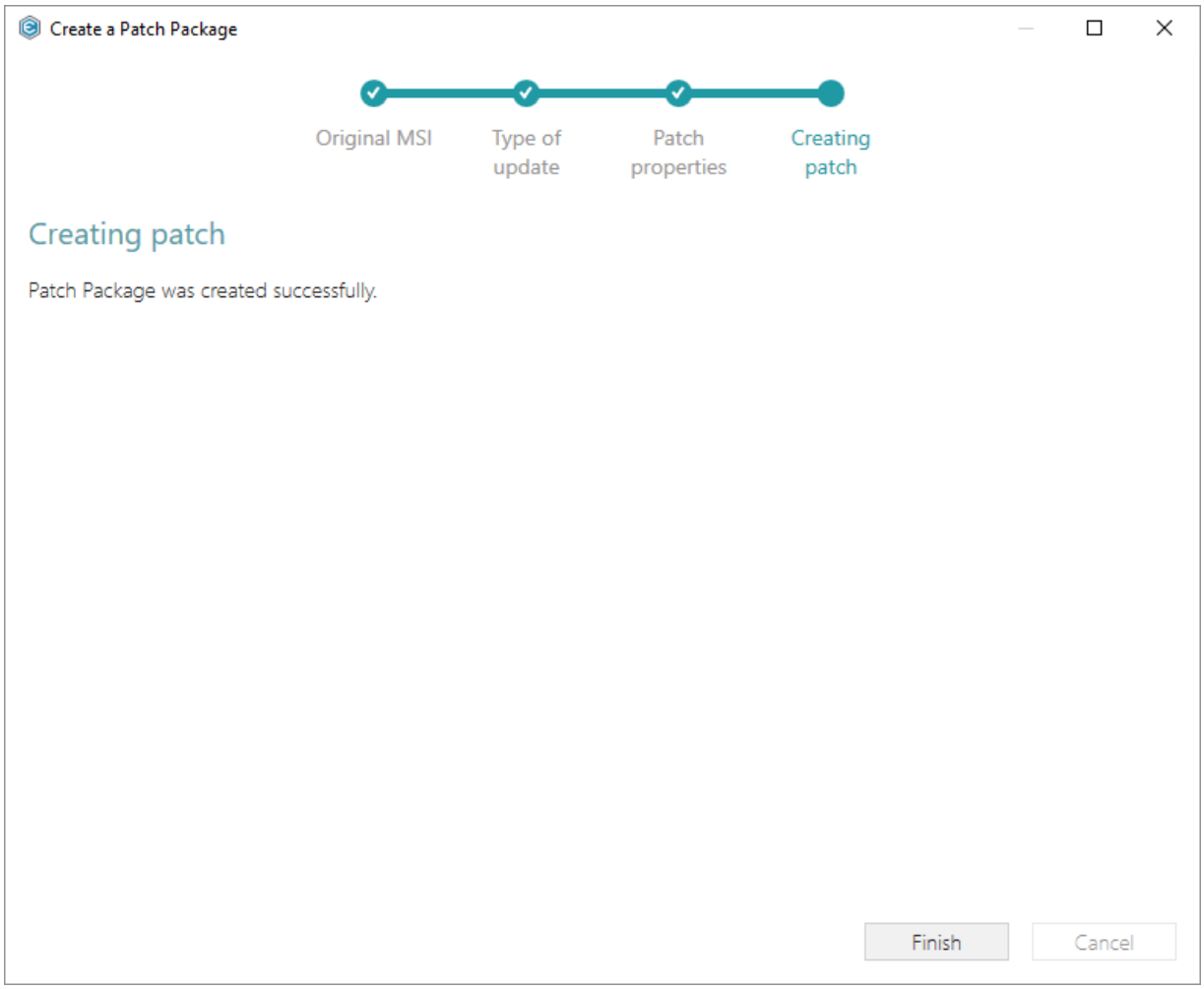
[10]. Please wait a little, while the Patch package is creating.




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



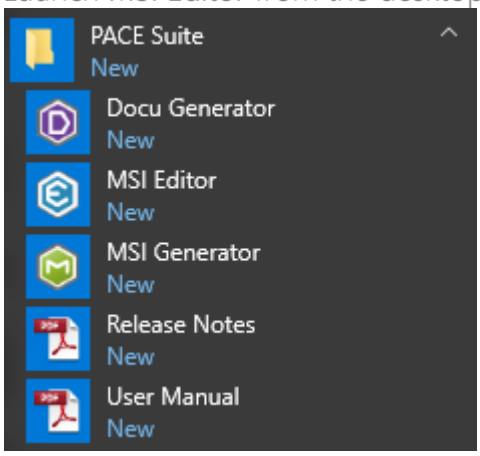
[12]. Click **Finish** to close the wizard.




## 3.3.2 Save Changes to MSP

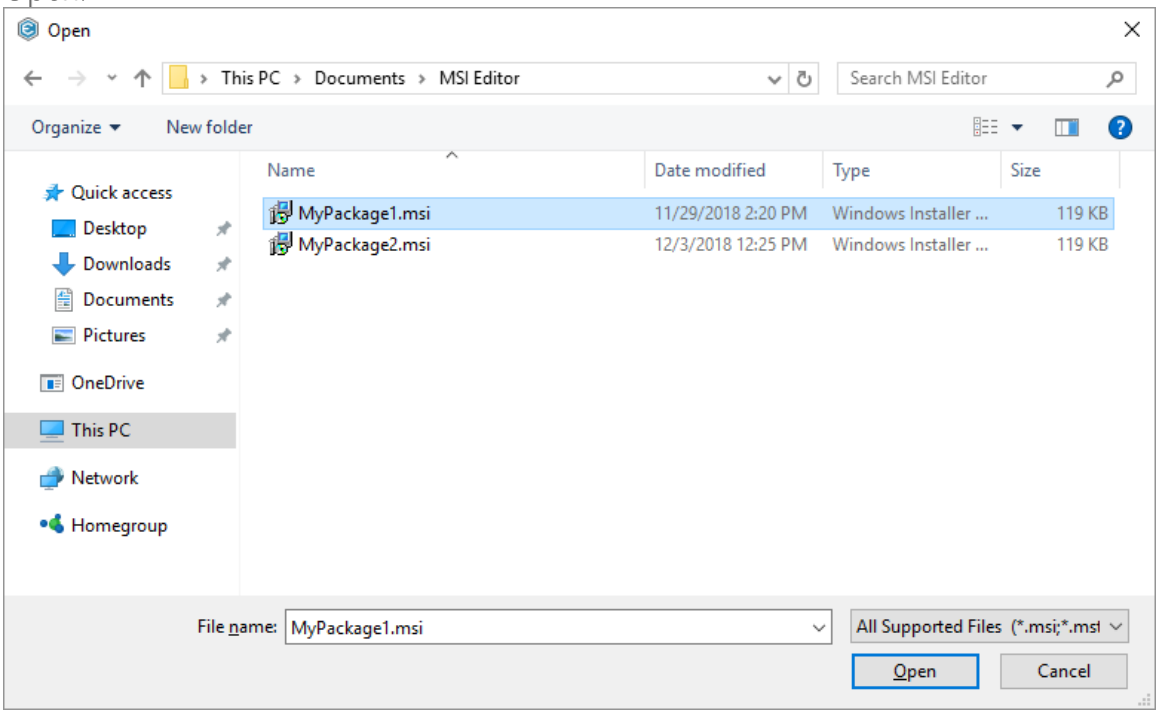
Save changes, made to the opened MSI database, as a Patch (MSP) package.

[1]. Launch **MSI Editor** from the desktop or the start menu shortcut.



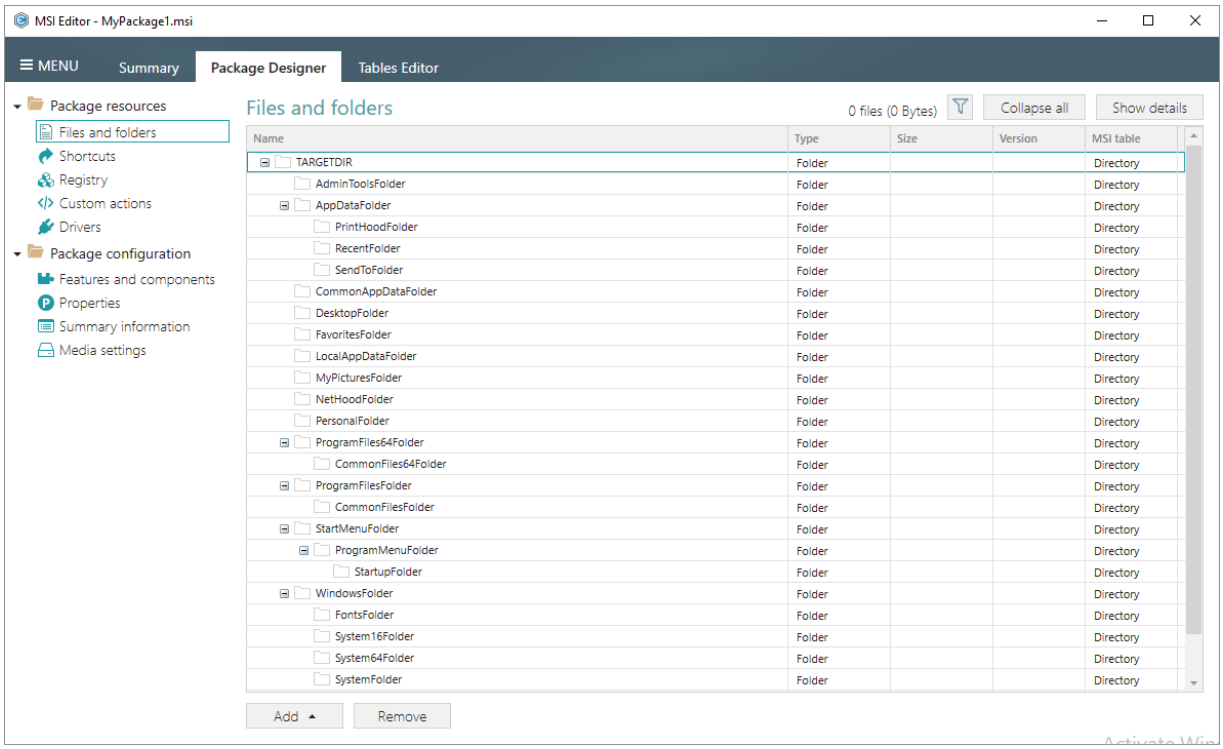
[2]. Select Open... from MENU.

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

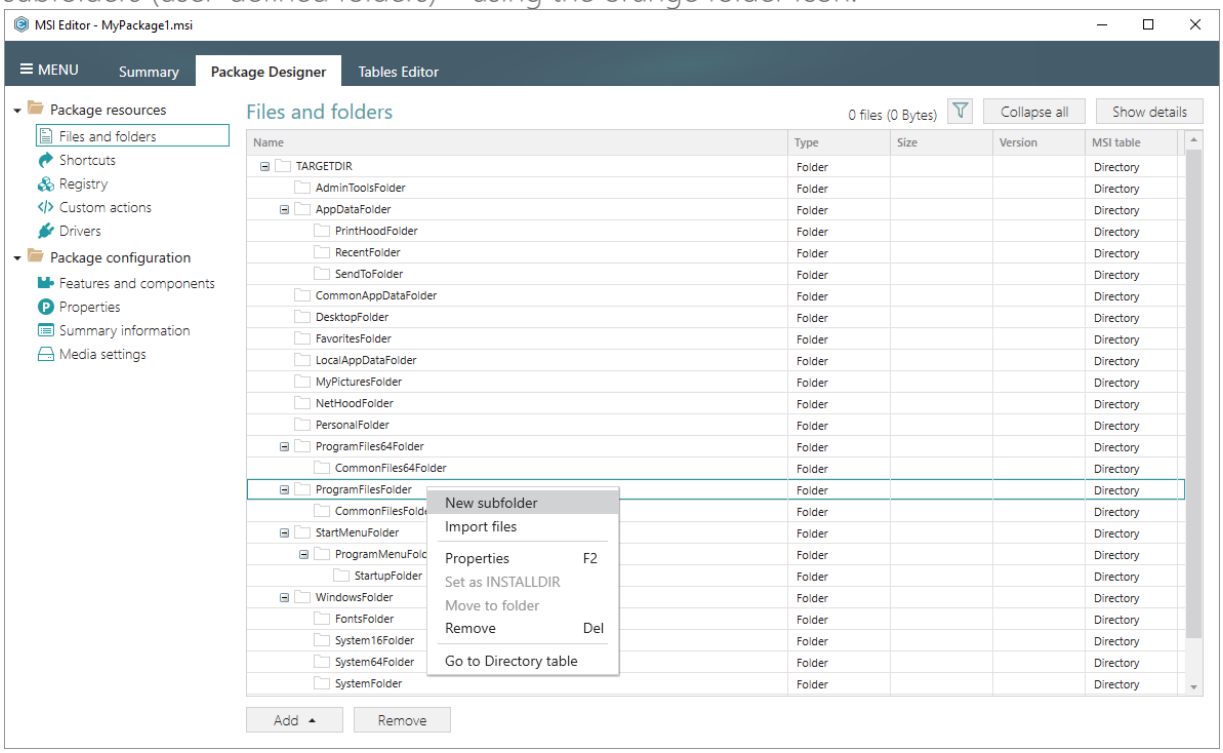



[4]. Now you can start adding the necessary resources, which you want to include to the Patch (MSP) package, using MSI Editor. The steps from [5] to [13] describe how to create a new folder and import files to this folder; the steps from [14] to [22] describe how to create a new registry key, value, and import registry entries from the REG files; the steps from [23] to [31] describe how to create a new and import existing shortcuts; and the steps [32]-[38] describe how to save all these changes to the Patch (MSP) package.

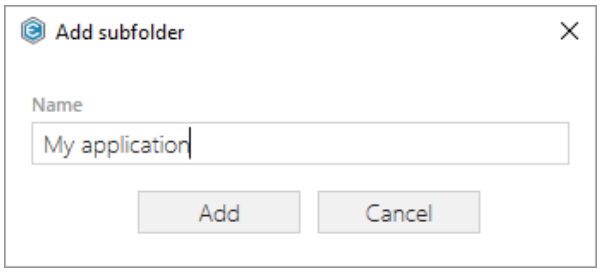
[5]. In order to create a new folder and import files to this folder, go to the **Package Designer –> Files and folders** tab.



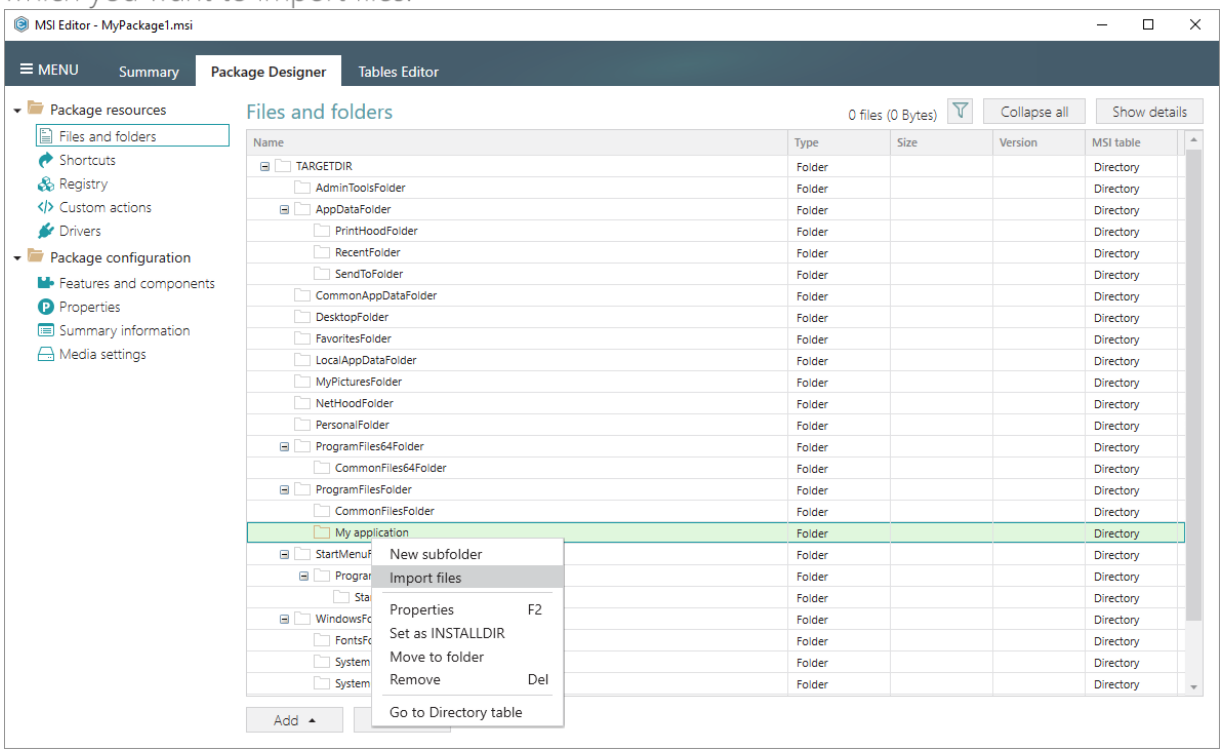

[6]. Select **New subfolder** from the context menu of a folder, where you want to create a new subfolder. Note that system folders are displayed using the grey folder icon and the subfolders (user-defined folders) – using the orange folder icon.



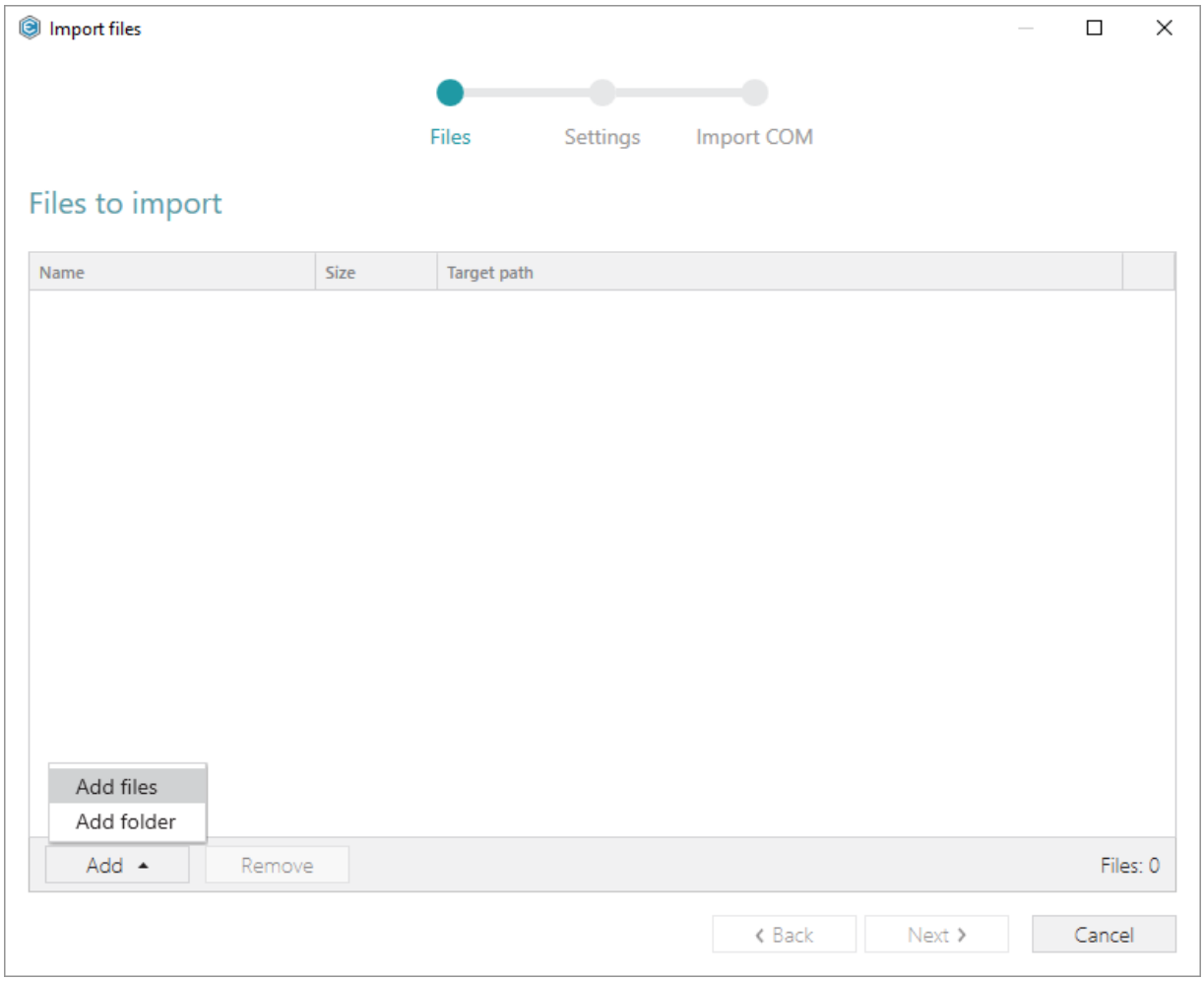
[7]. Enter a folder name and click **Add**.



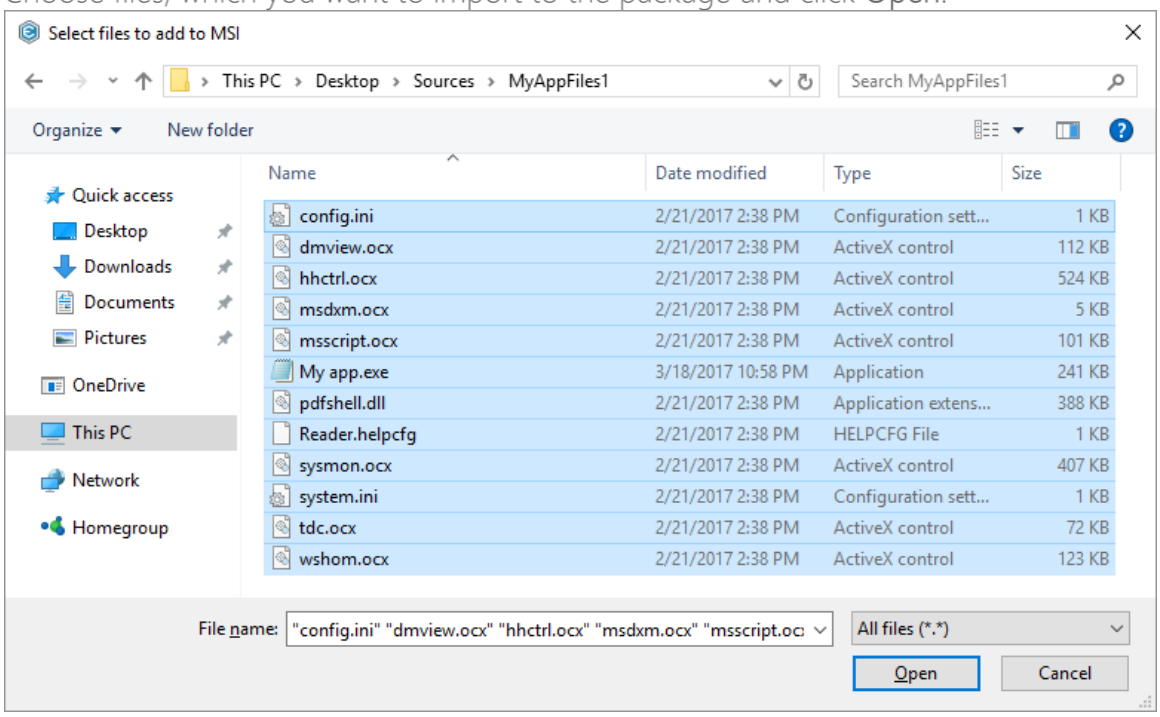

[8]. For adding files to your package, select **Import files** from the context menu of a folder, to which you want to import files.



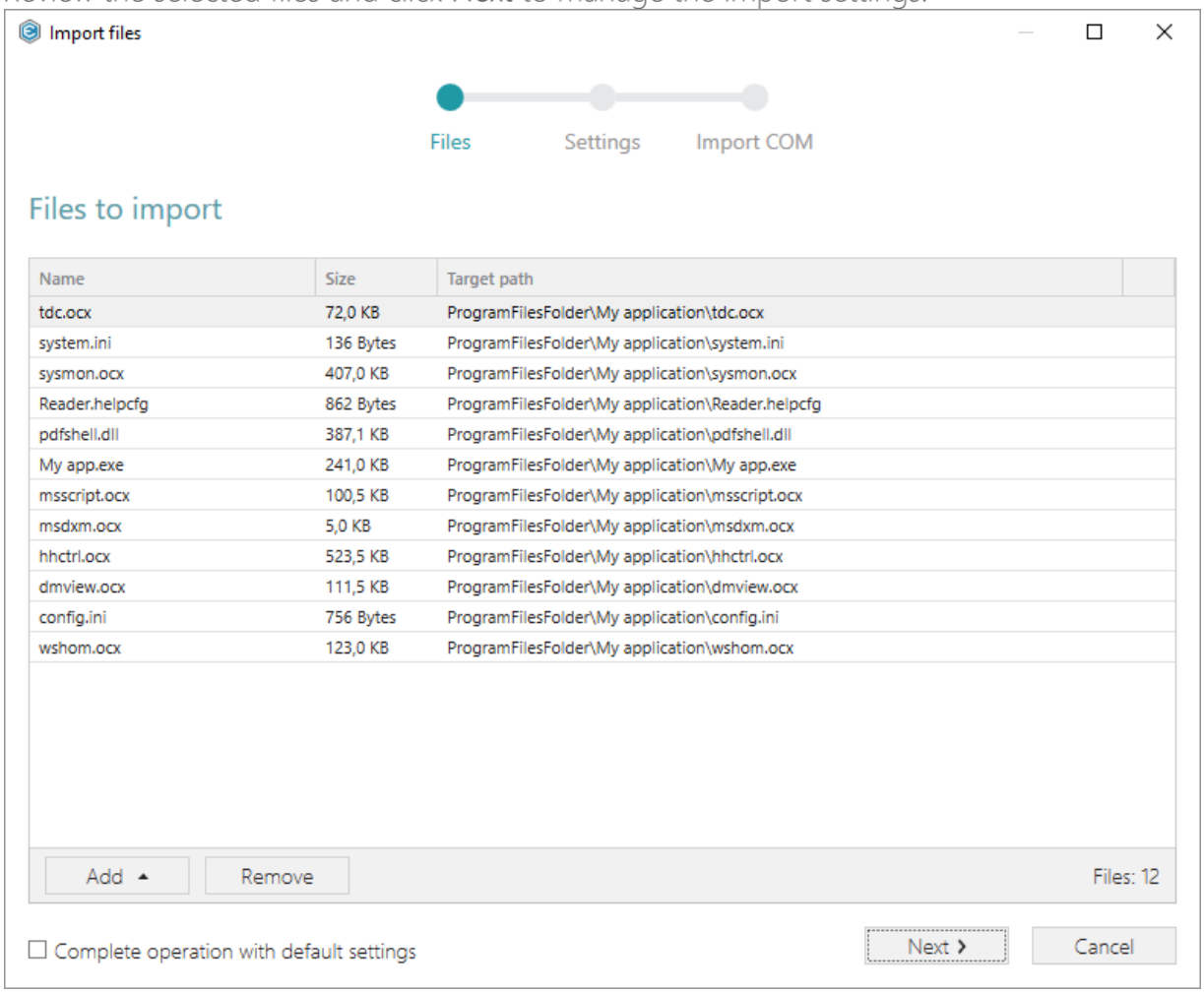
[9]. Click **Add** -> **Add files** to choose files in a folder or **Add** -> **Add folder** to choose a folder with subfolders and files.



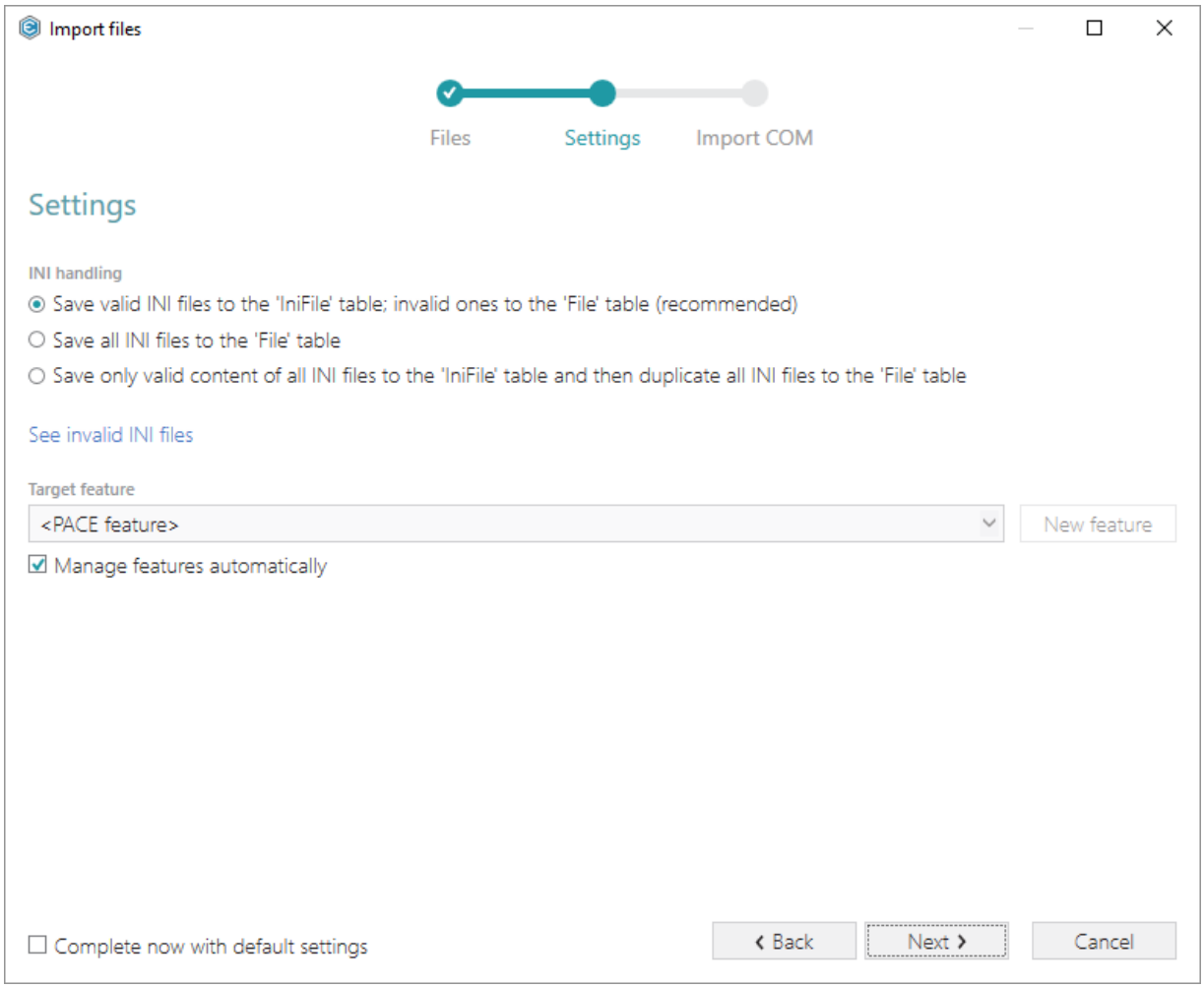

[10]. Choose files, which you want to import to the package and click **Open**.



[11]. Review the selected files and click **Next** to manage the import settings.




[12]. At this step, you can manage how to save INI files, select an MSI Feature, to which components with files will be assigned. Below, in the tables, you will find the detailed description of these settings. Note that files compression types and other Media settings are represented on the separate **Media settings** tab of the **Package Designer**. Choose the desired settings and click **Next** to continue.




## INI handling

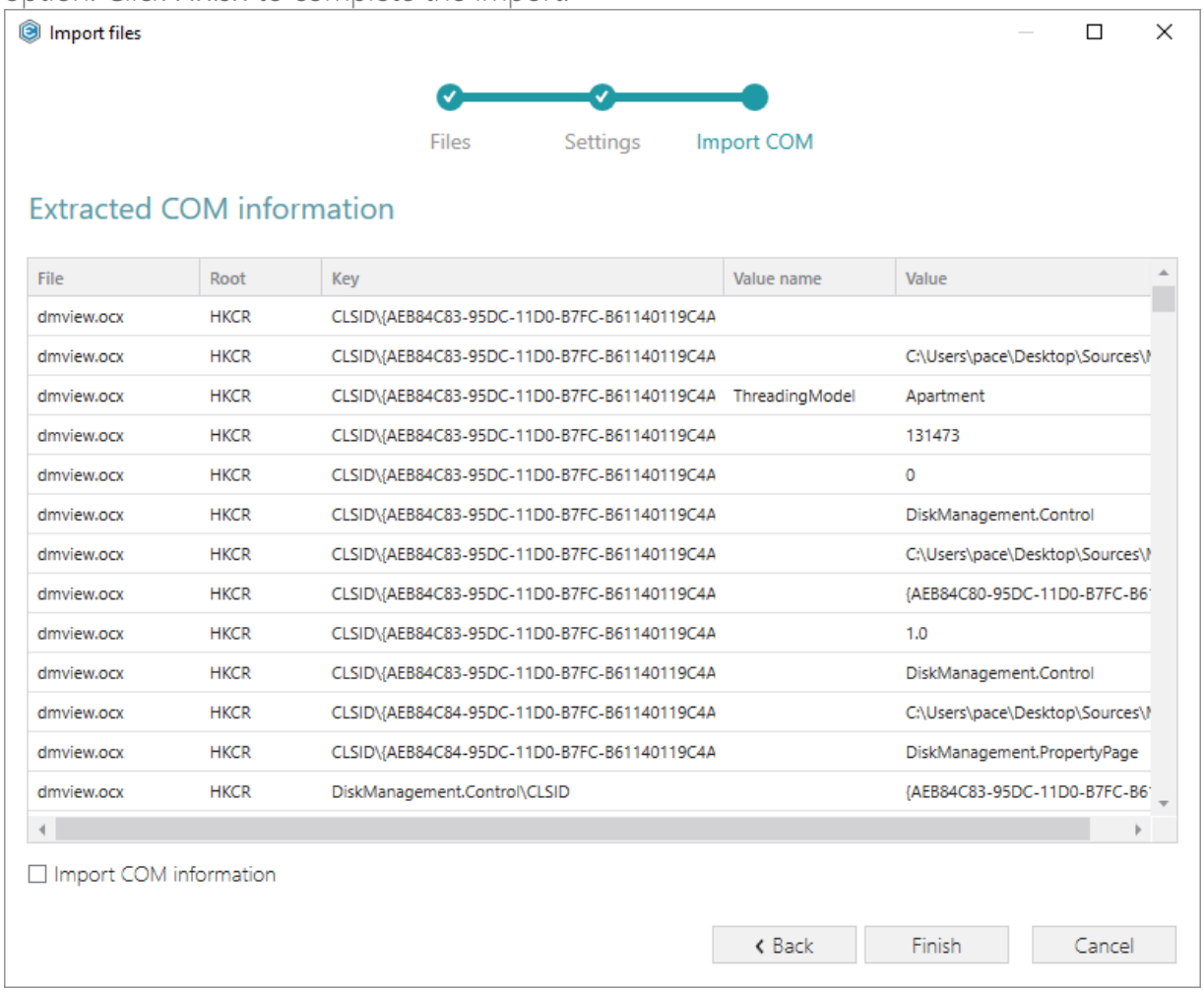
| | |
|---|---|
| Save valid INI files to the 'IniFile' table; invalid ones to the 'File' table (recommended) | INI files, which comply with the INI file format, will be saved to the 'IniFile' table.<br>All other INI files, which contain unsupported data, will be saved to the 'File' table as binary files. |
| 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.<br>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

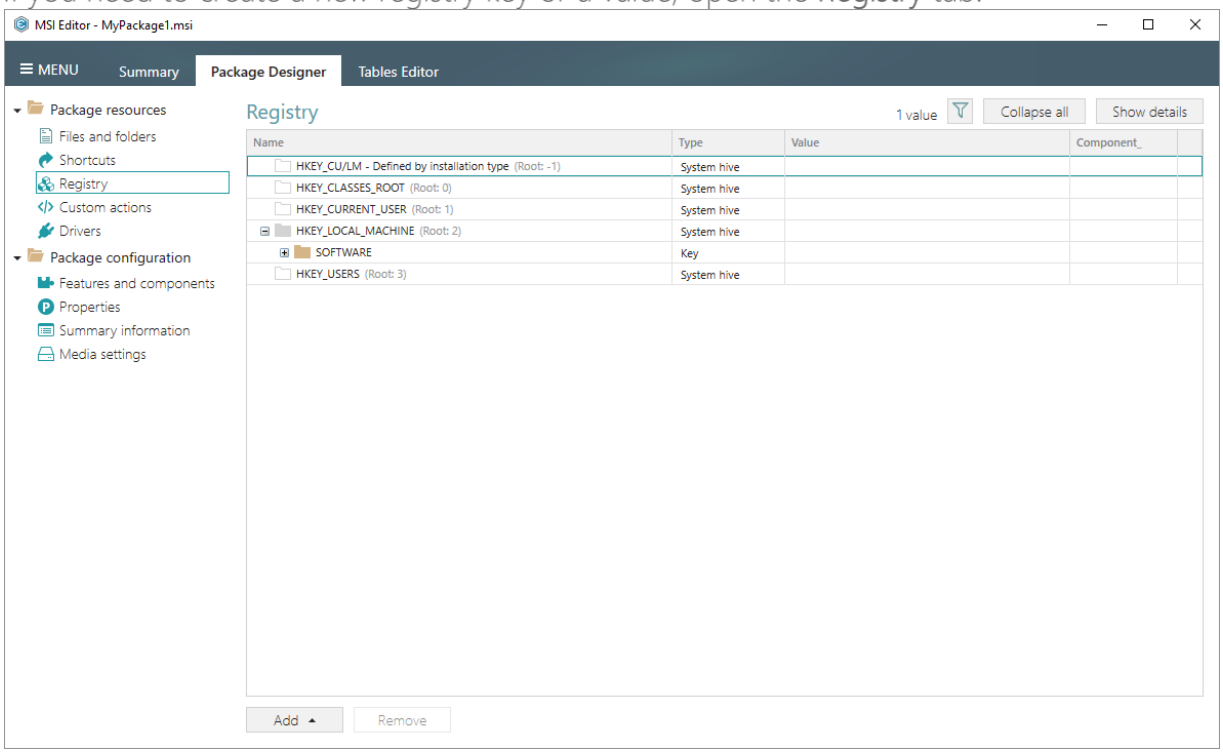
| | |
|---|---|
| Manage features automatically | The specific PACE features will be created (or used if they exist) for the components, containing the imported files.<br>Feature structure is based on the packaging best practices and depends on the destination location of files. |

| | |
|---|---|
| <a feature, selected from the list> | Components with files will be assigned to a feature, selected from the list. |

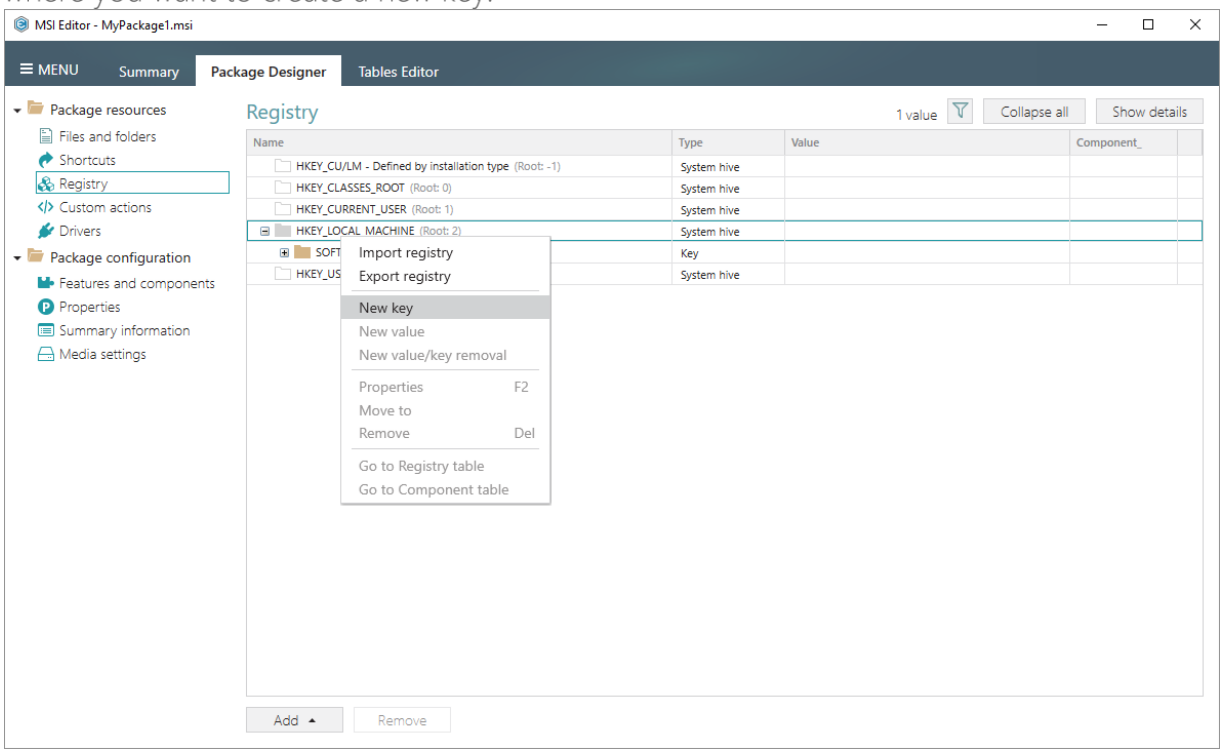
[13]. On the last step you can review COM information, extracted from the selected files. For importing this COM information along with files, select the Import COM information option. Click Finish to complete the import.



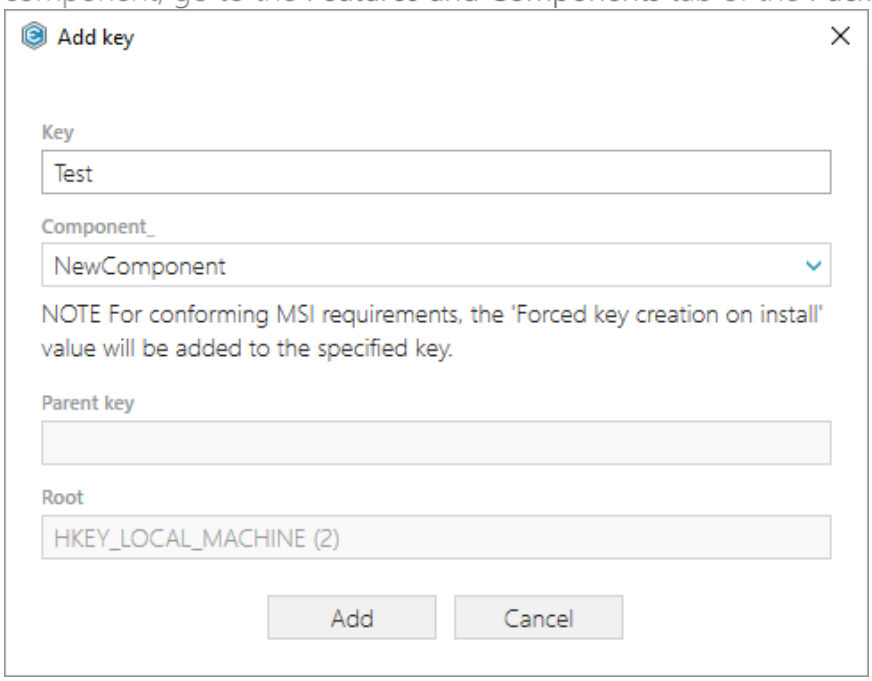
[14]. If you need to create a new registry key or a value, open the **Registry** tab.



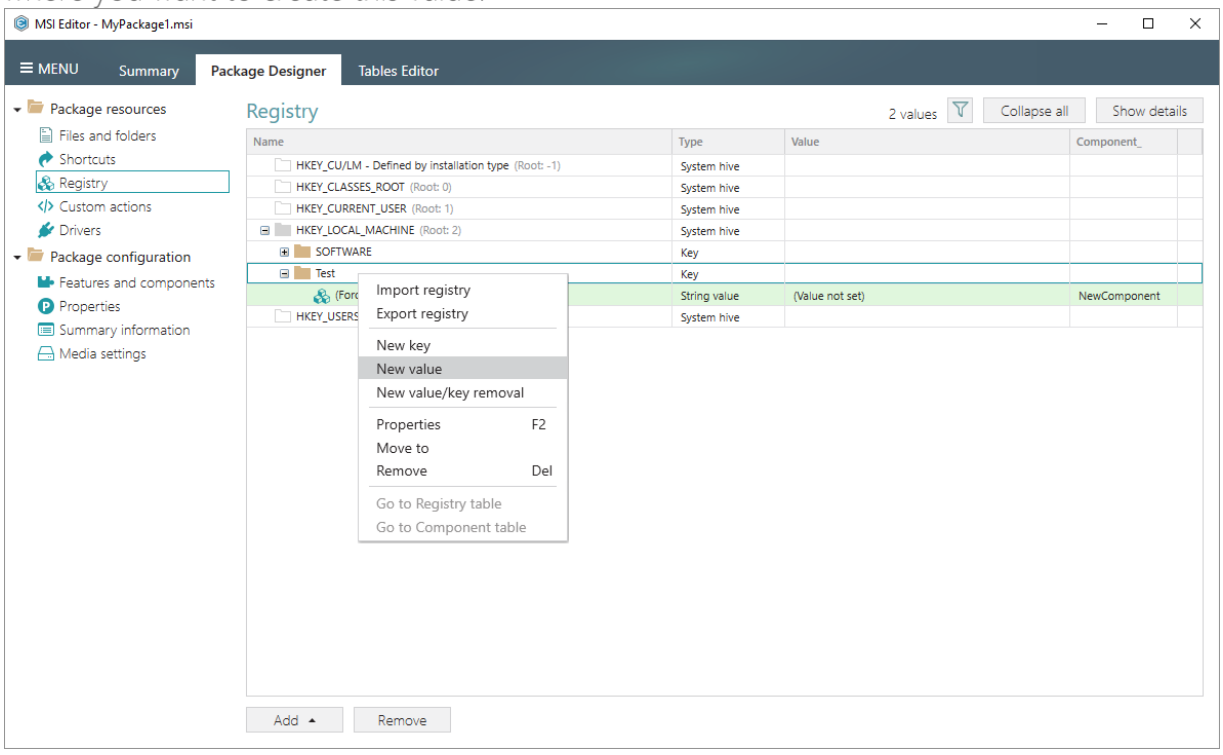

[15]. Select **New key** from the context menu of a root hive (e.g. HKEY_LOCAL_MACHINE), where you want to create a new key.



[16]. 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**.




[17]. In order to create a new registry value, select **New value** from the context menu of a key, where you want to create this value.

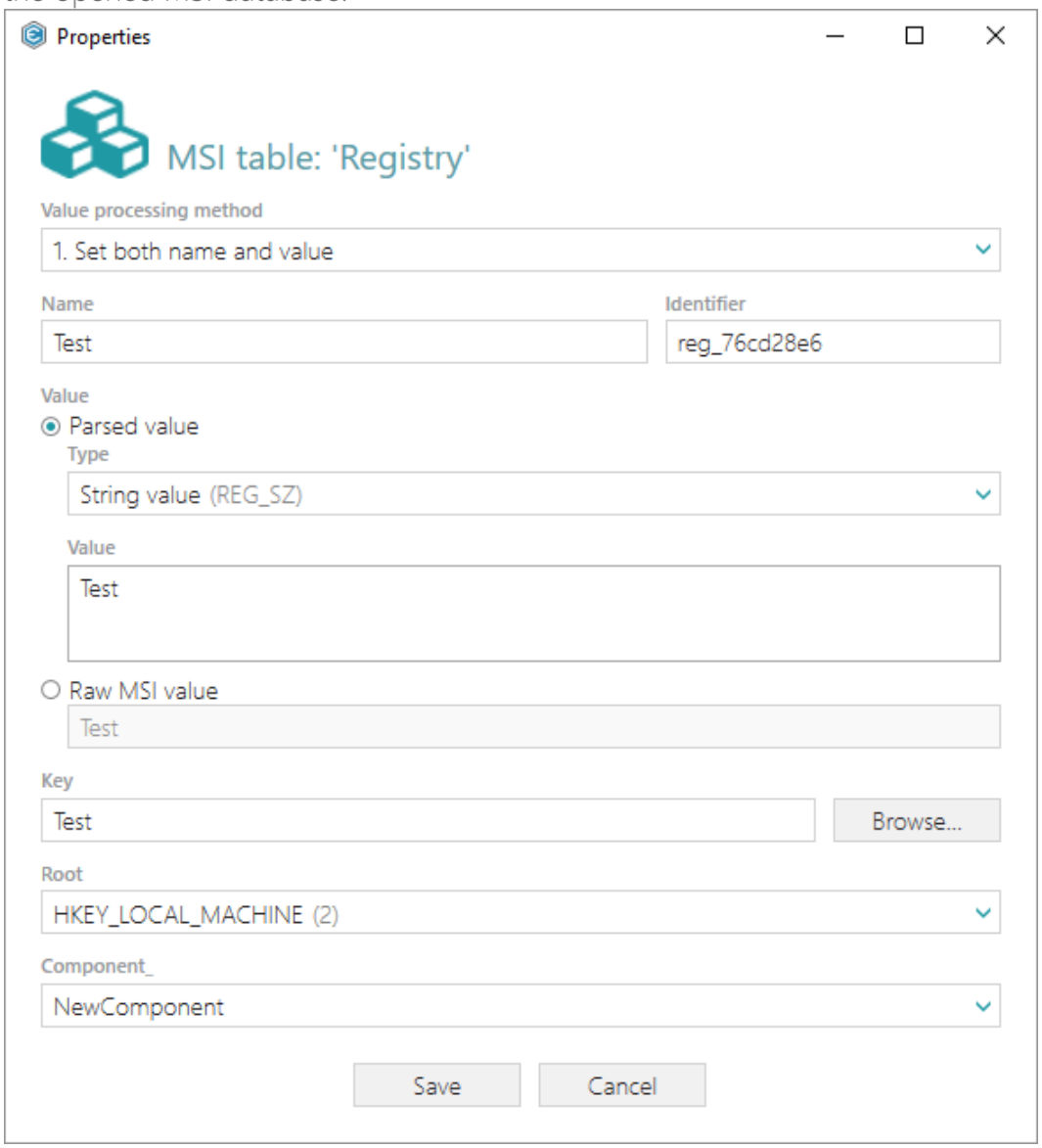



[18]. For creating an ordinary registry value, specify a name and its value, choose a component from the list, to which this value will be assigned, and click **Save**.

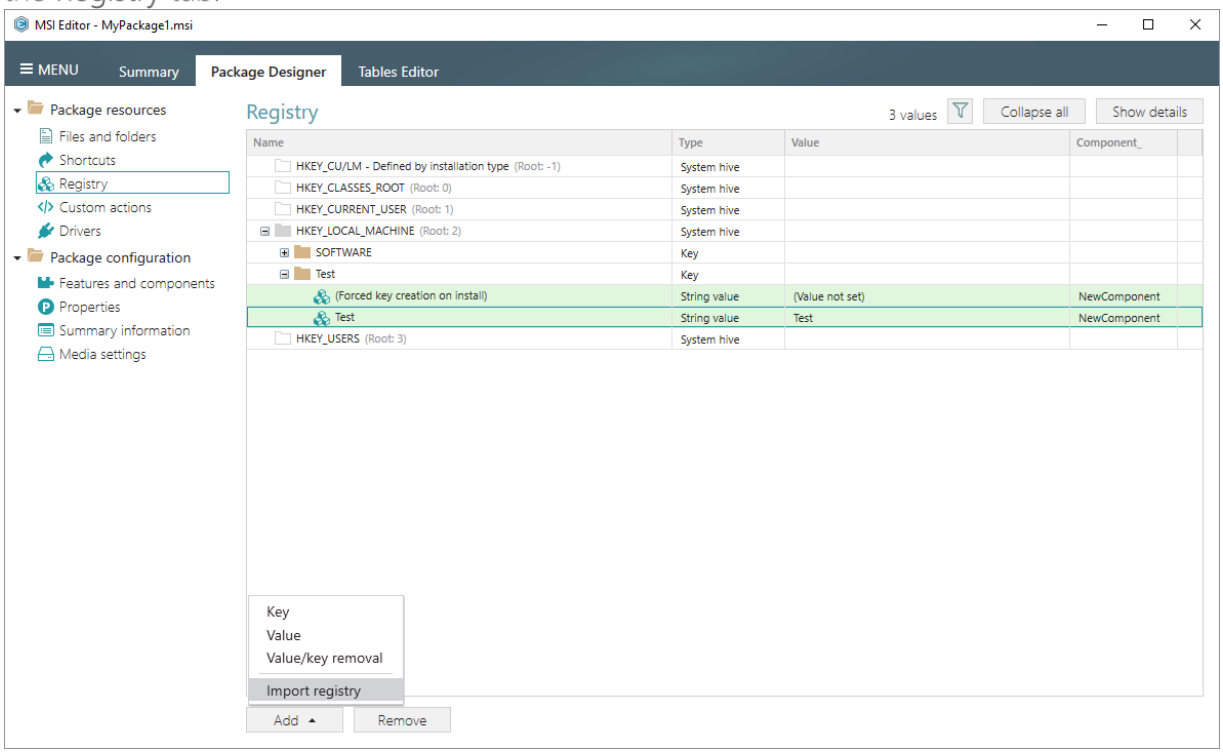
For creating a Default key value, or securing a key creation and/or removal, select a respective option from the **Value processing method** field.

The **Parsed value** option serves easier creation of a registry value of the proper value type.

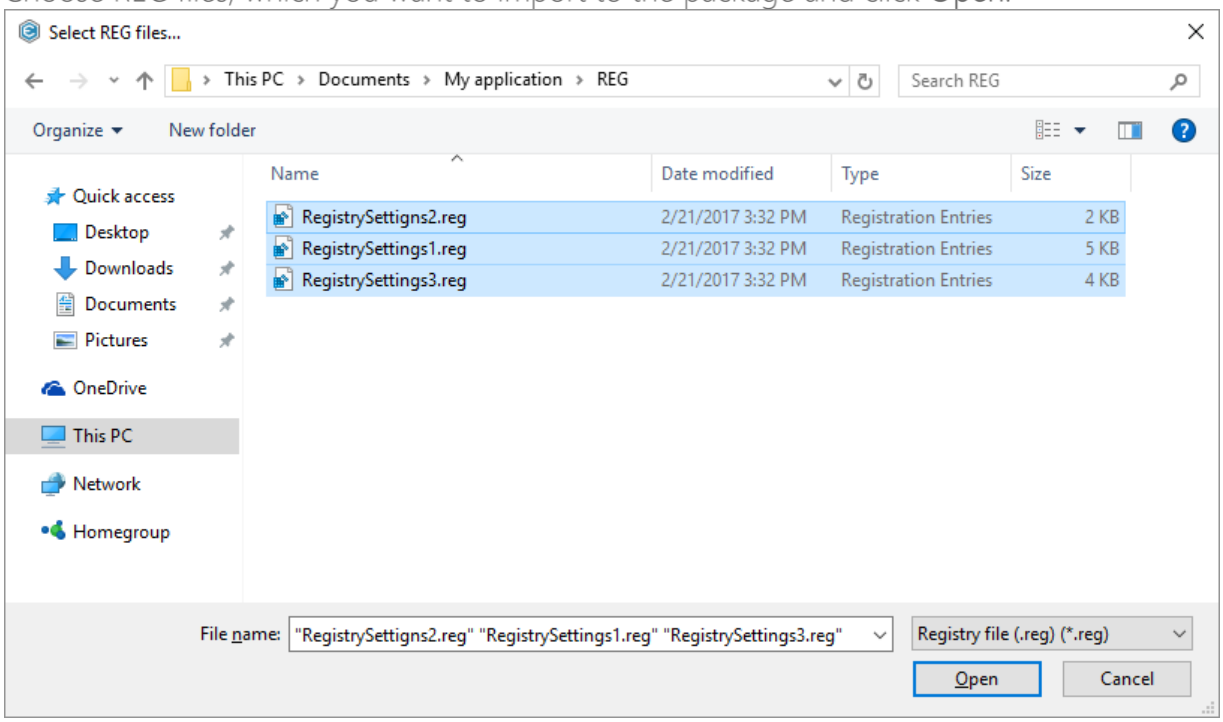
However, the **Raw MSI value** option gives ability to manage what exactly will be saved to the opened MSI database.



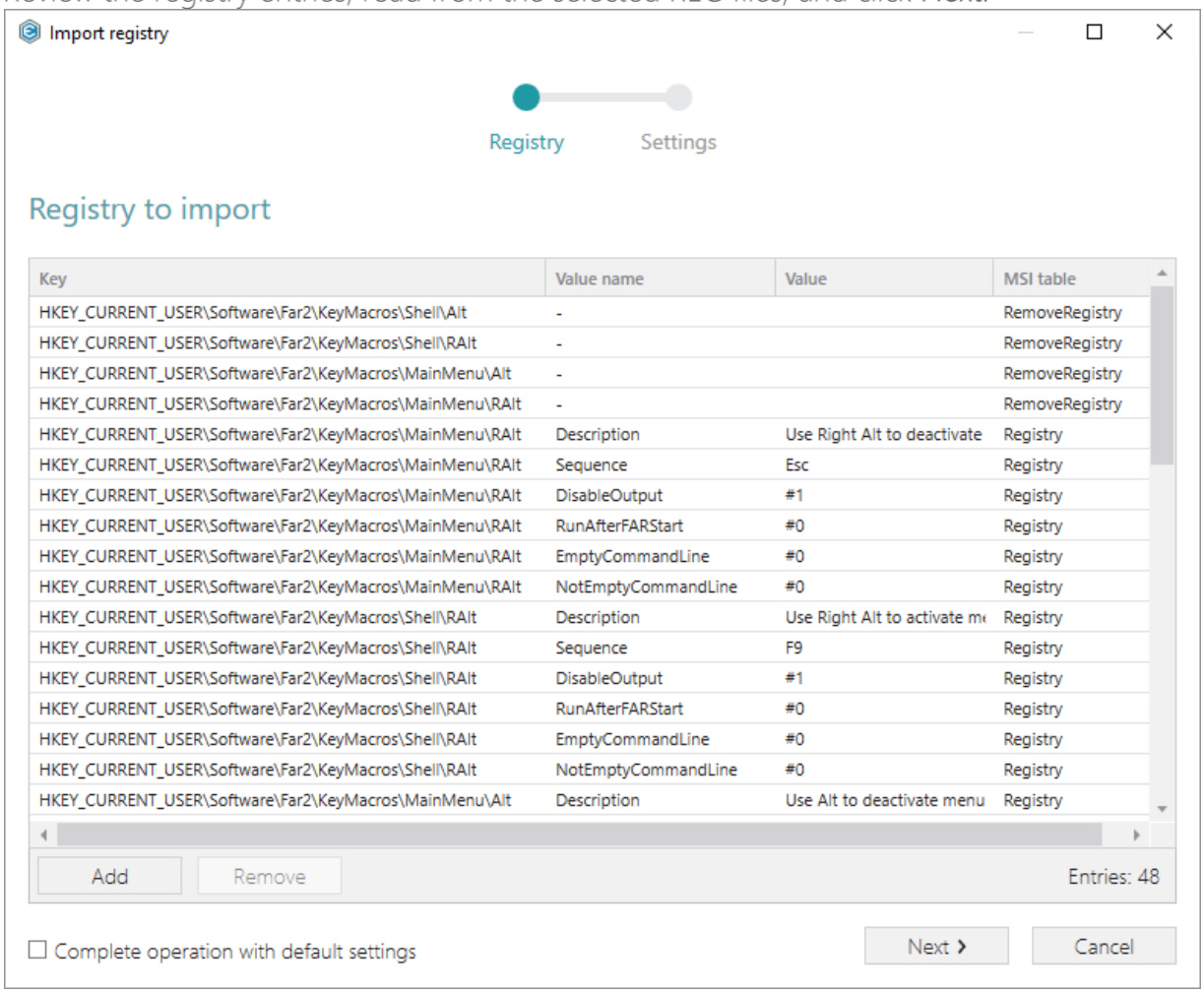
[19]. For importing registry from the REG file, select **Add** -> **Import registry** from the toolbox of the **Registry** tab.




[20]. Choose REG files, which you want to import to the package and click **Open**.

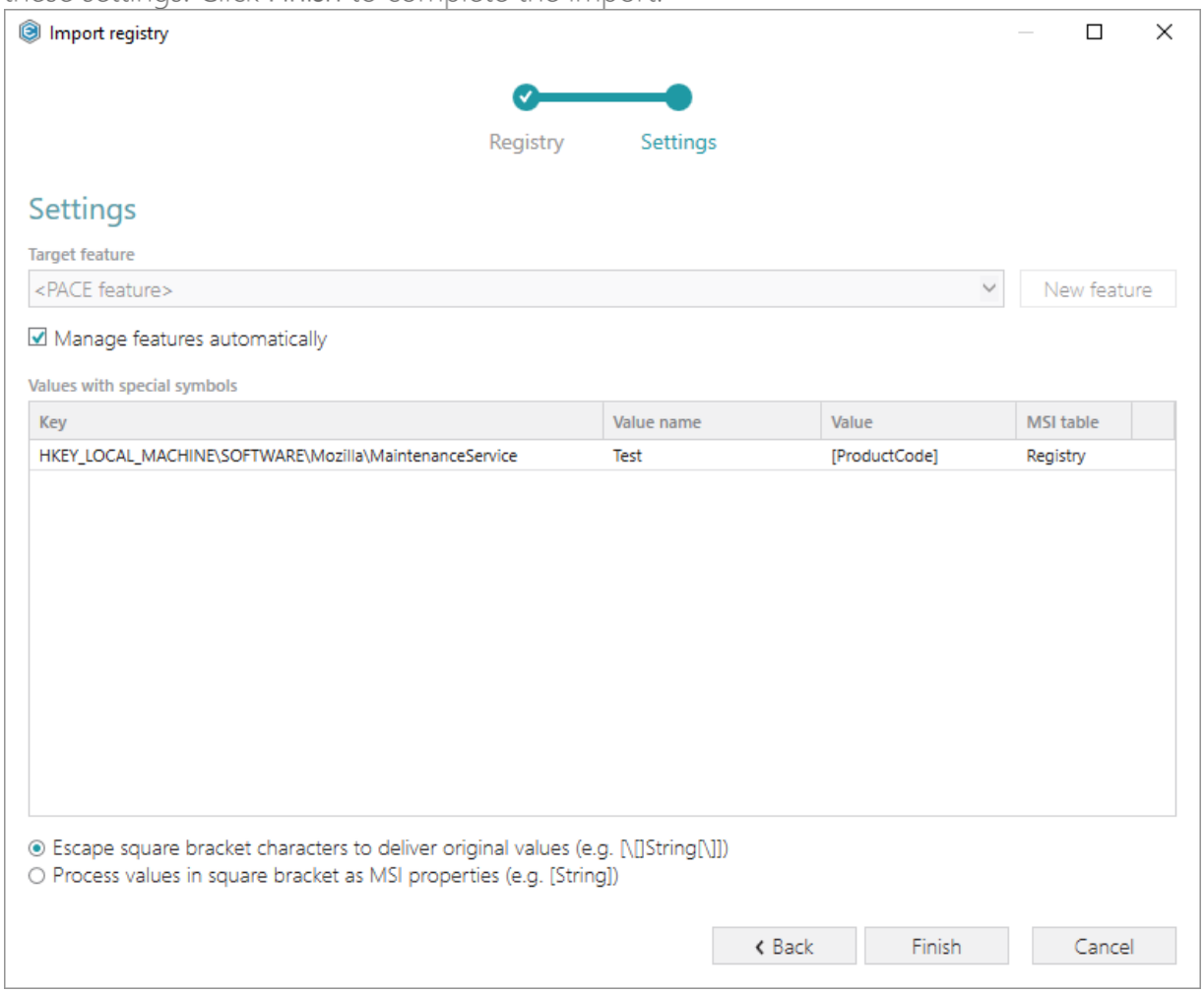


[21]. Review the registry entries, read from the selected REG files, and click Next.




[22]. 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

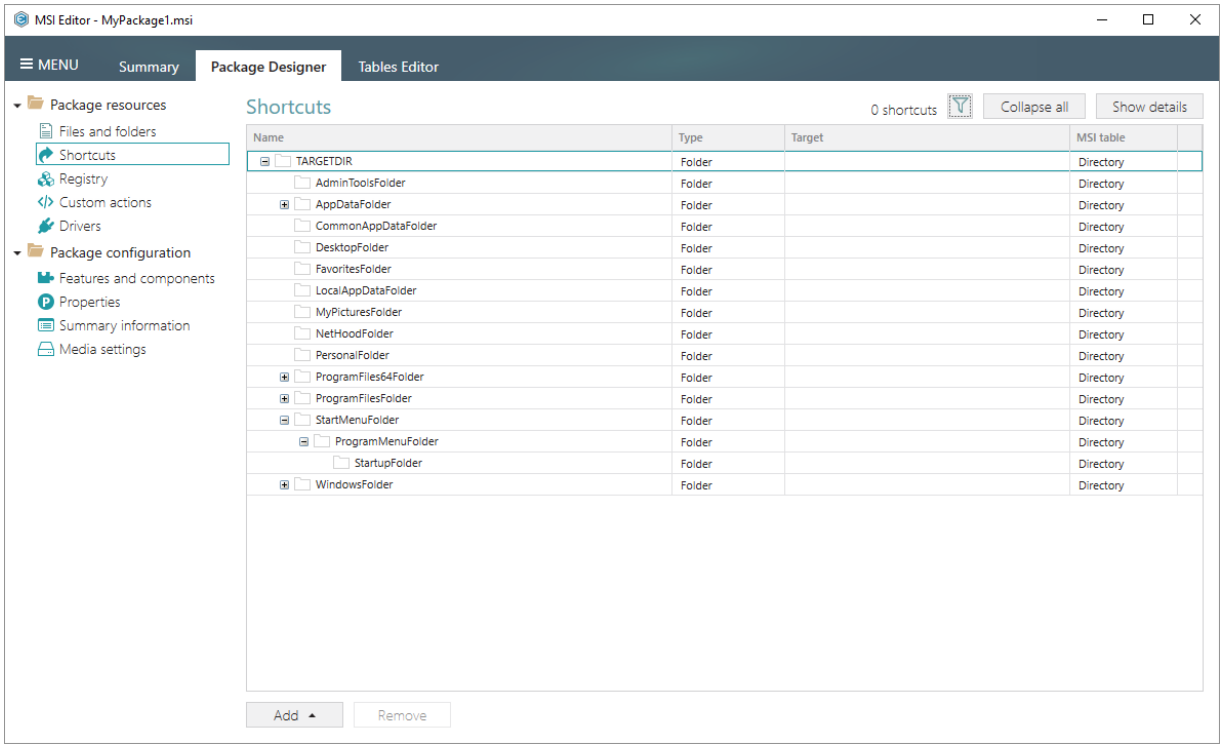
| | |
|---|---|
| Manage features automatically | The specific PACE features will be created (or used if they exist) for the components, containing the imported registry. Feature structure is based on the packaging best practices and depends on the registry key location. |
| <a feature, selected from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

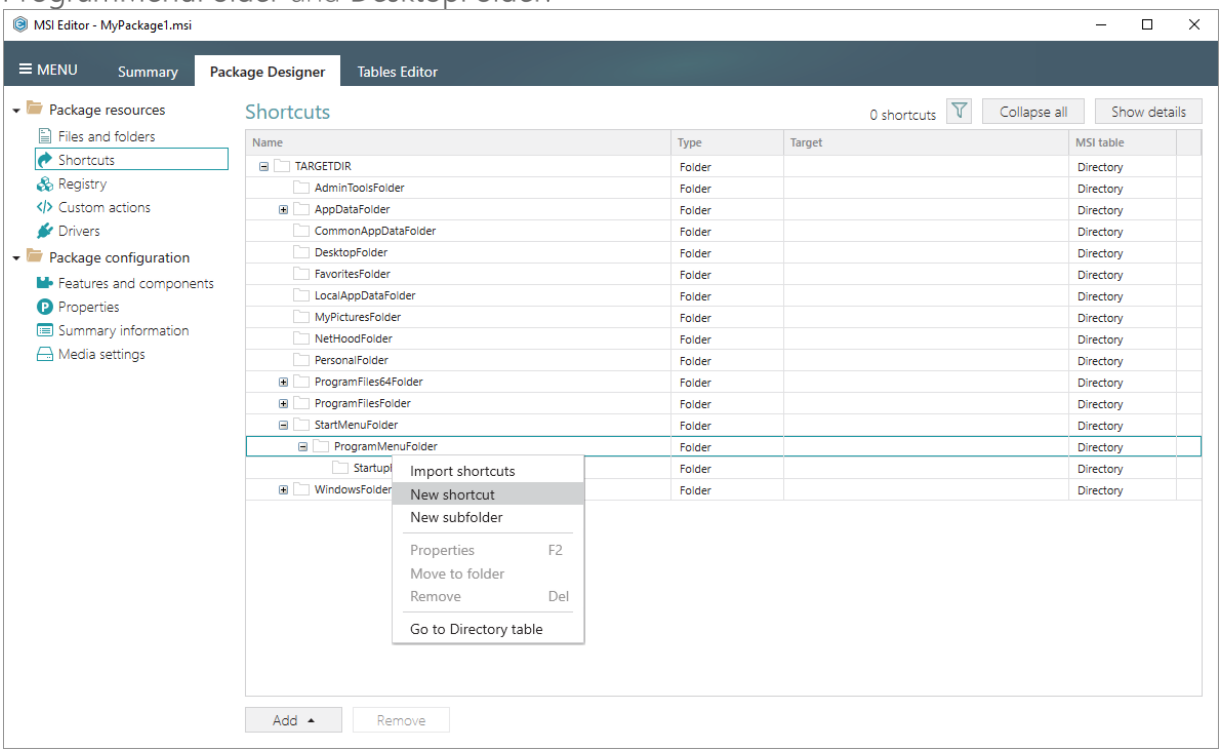
| | |
|---|---|
| Process values in square brackets as MSI properties | Values in square brackets, displayed in the list, will be saved to MSI as MSI properties. Thus, each MSI property will be replaced with its value during the MSI installation. If a property does not exist in MSI during the installation,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

[23]. If you need to create a new or import the existing shortcuts to the package, go to the Shortcuts tab.



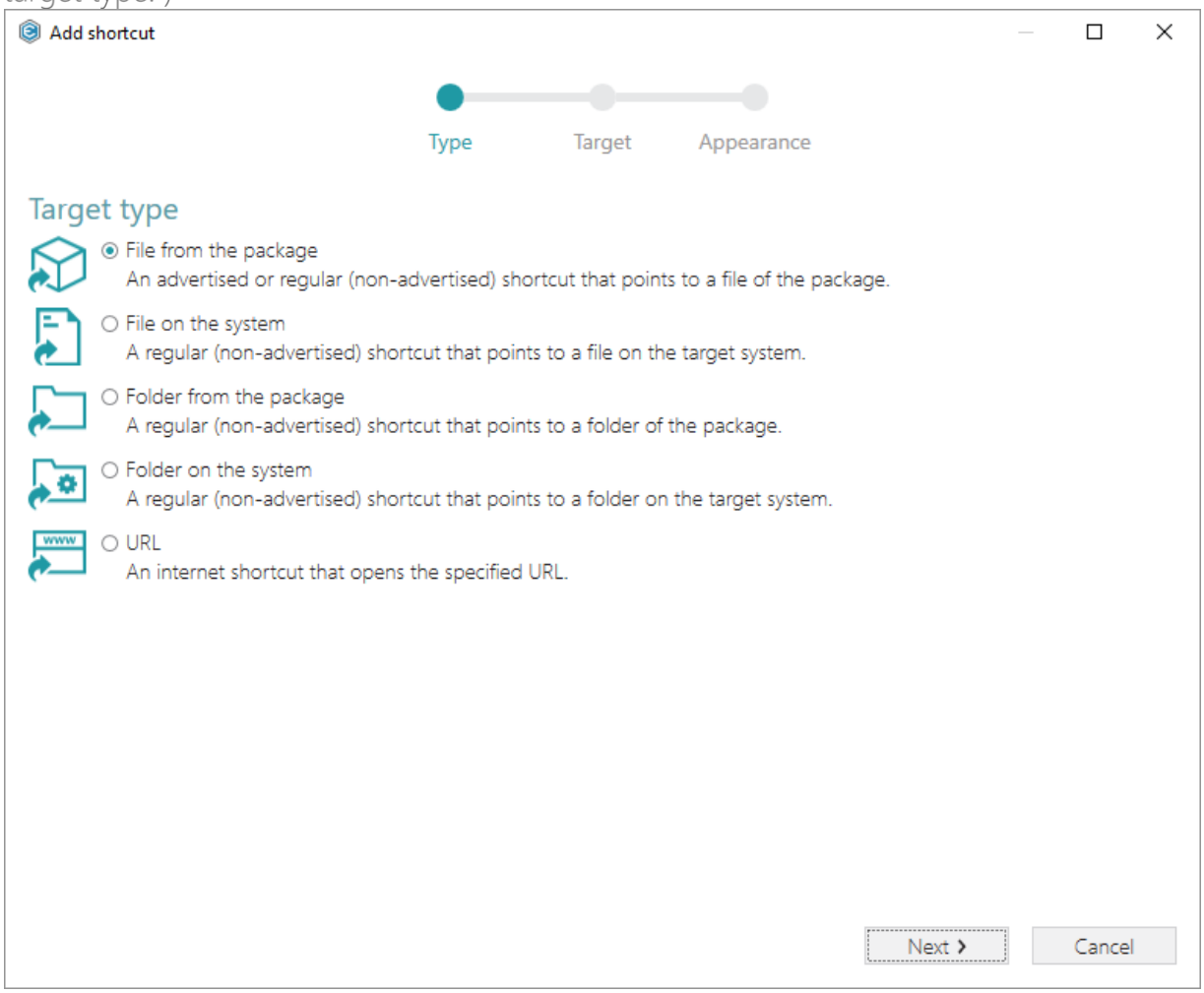

[24]. In order to add a new shortcut, select New shortcut from the context menu of a folder, where you want to create this shortcut. Usually, application shortcuts are installed to the

ProgramMenuFolder and DesktopFolder.

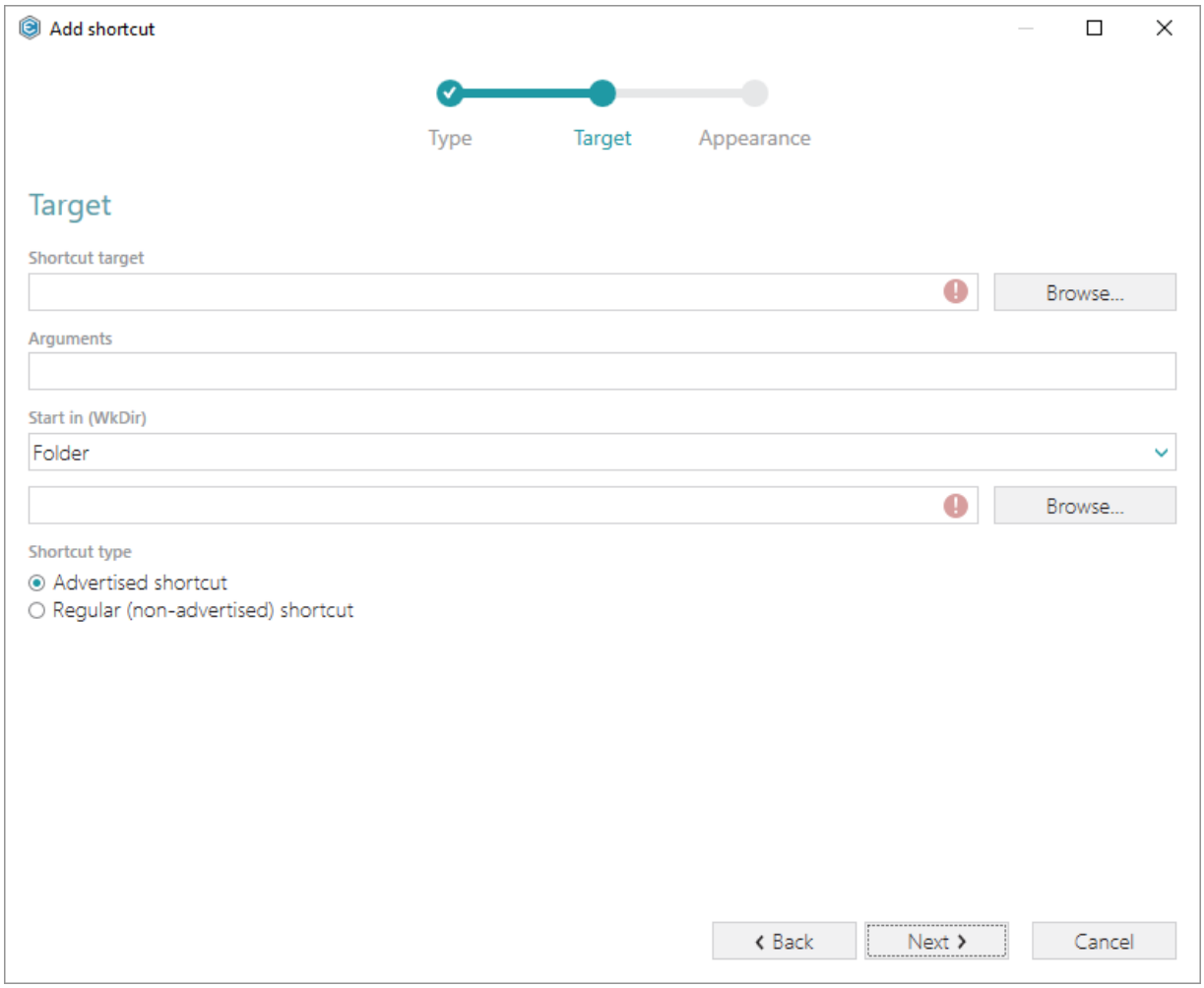



[25]. Select a type of the shortcut target and click Next.
(The following instructions describe creation of the shortcut of the 'File from the package'

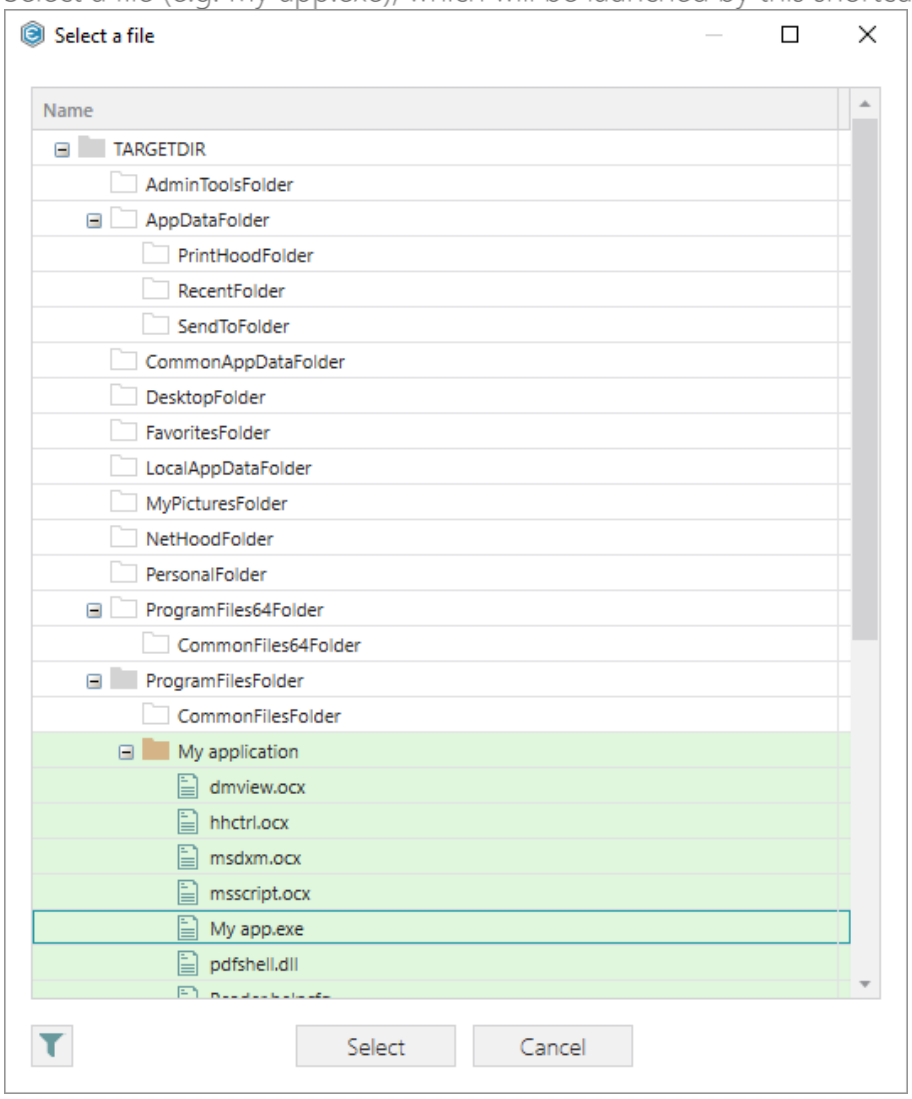
target type. )


Add shortcut
Type
Target
Appearance
Target type
File from the package
An advertised or regular (non-advertised) shortcut that points to a file of the package.
File on the system
A regular (non-advertised) shortcut that points to a file on the target system.
Folder from the package
A regular (non-advertised) shortcut that points to a folder of the package.
Folder on the system
A regular (non-advertised) shortcut that points to a folder on the target system.
URL
An internet shortcut that opens the specified URL.
Next ›
Cancel

[26]. On the second step, click **Browse**, located next to the **Shorcut target** field, to set a target of the shortcut.

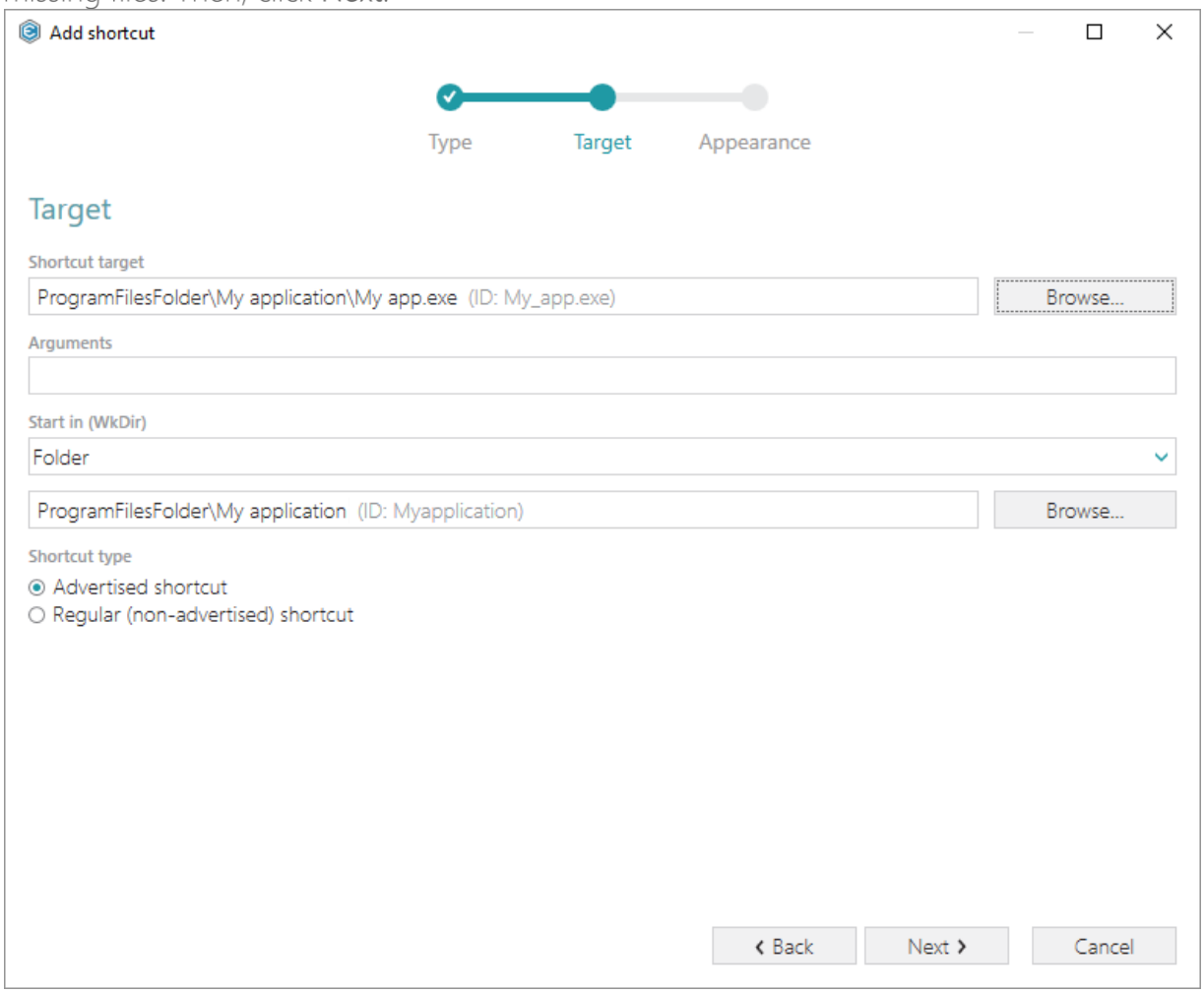


[27]. Select a file (e.g. My app.exe), which will be launched by this shortcut, and click **Select**.

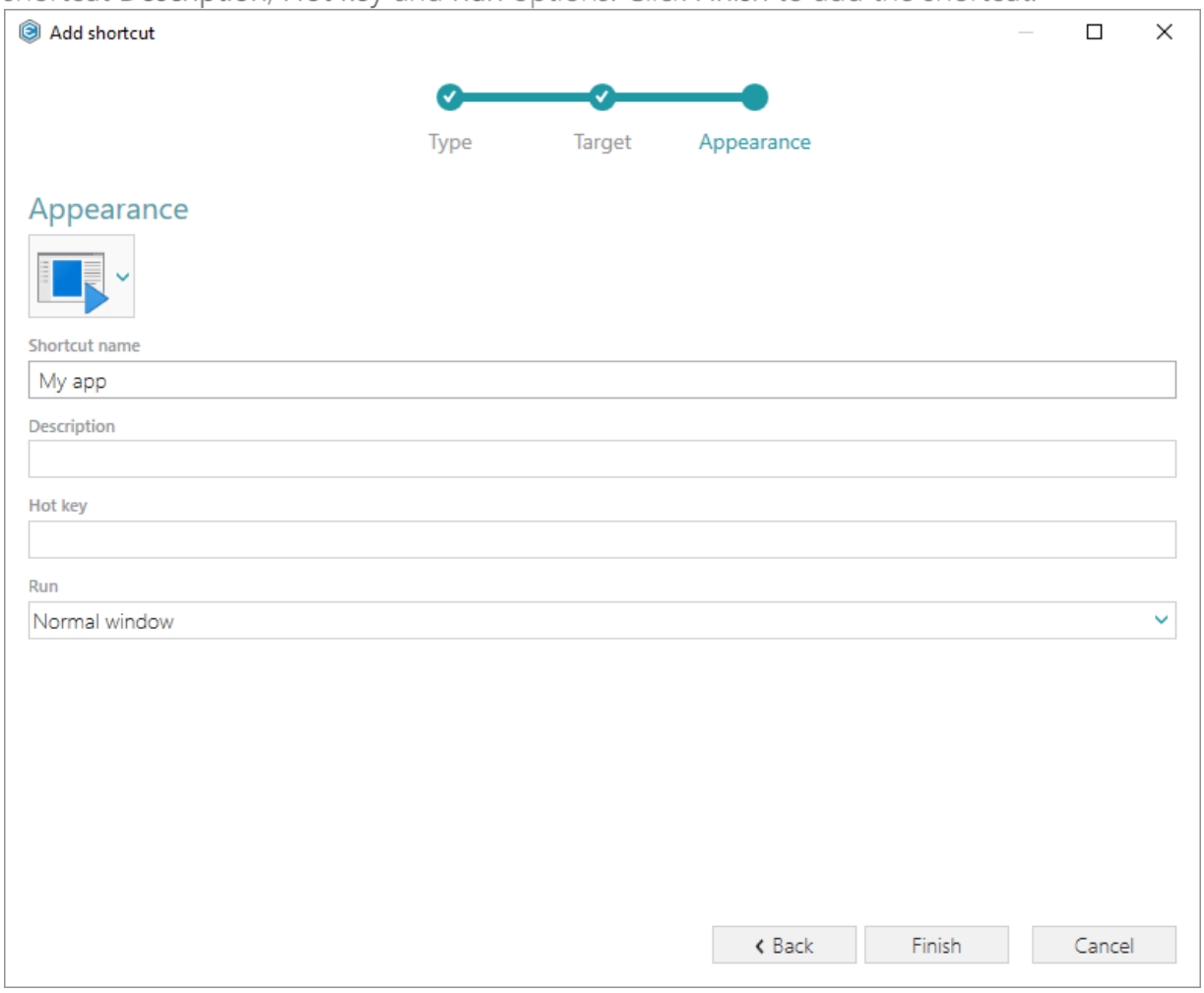



[28]. If needed, enter a shortcut **Arguments**, change the **Start in** path (select a folder or a property), and select a **Shortcut type**. Note that only the Advertised shortcut supports advertisement and repair. This means that if the executable to which the shortcut is pointing is not there then windows installer will repair the application and replace the

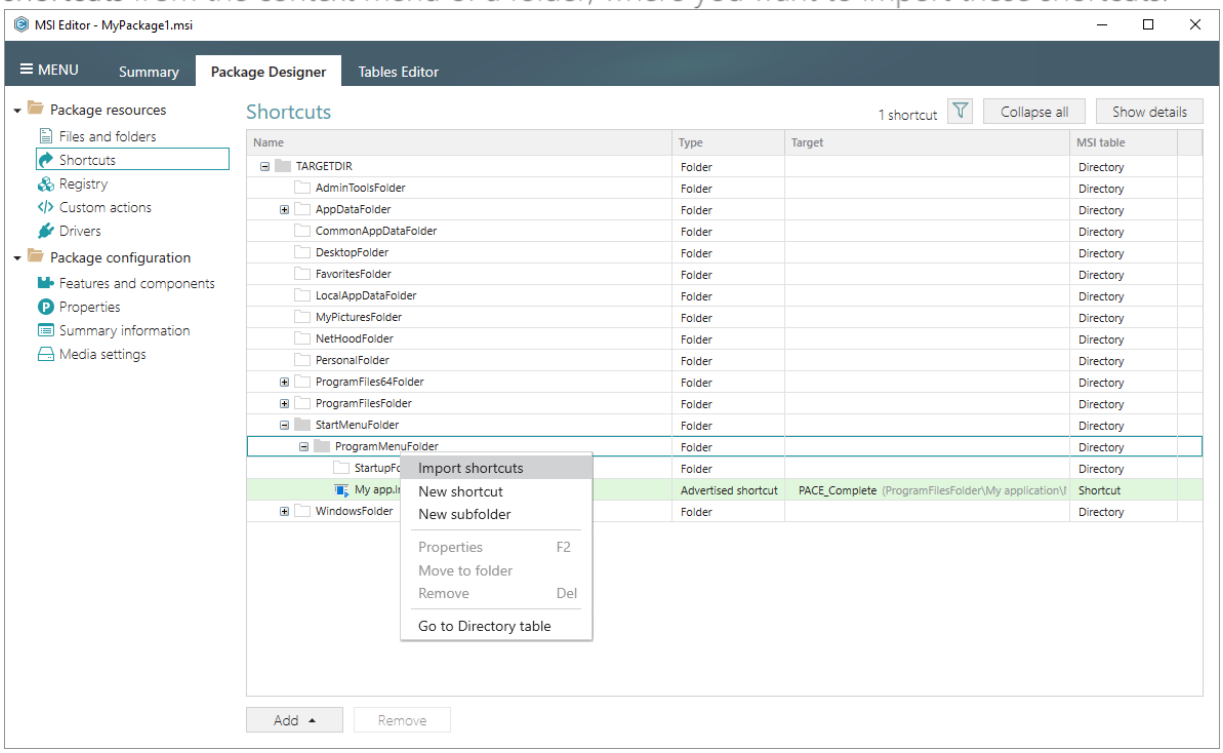
missing files. Then, click **Next**.



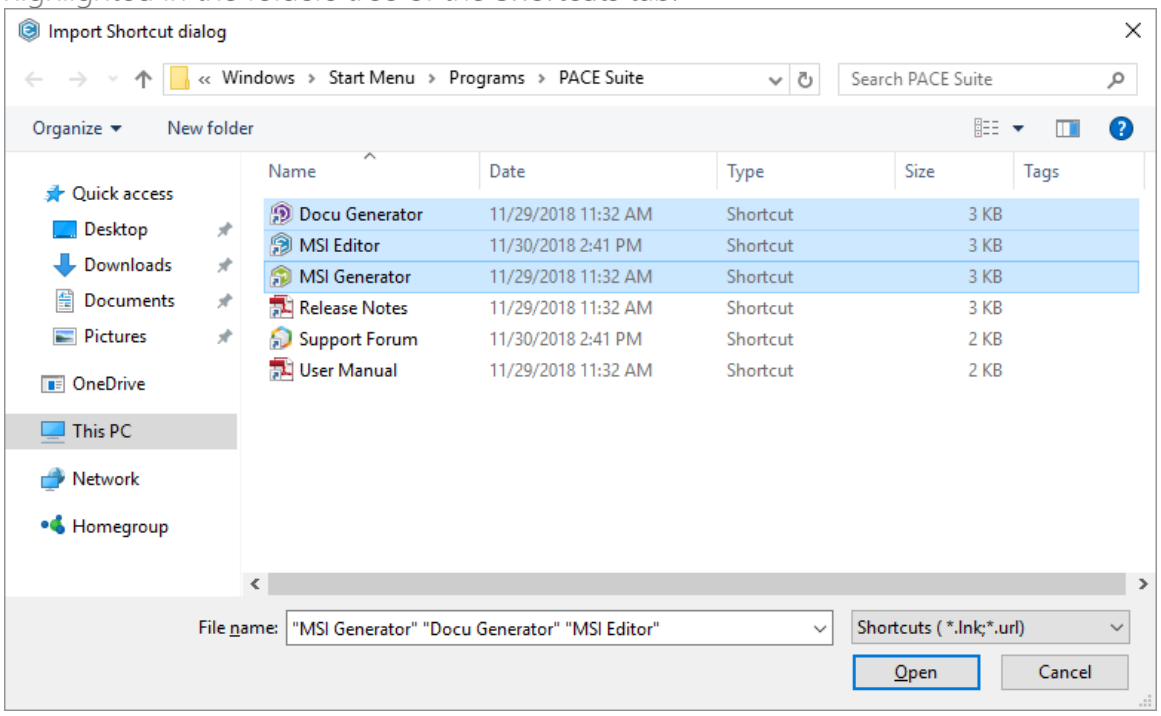
[29]. On the last step, enter a **Shortcut name**, choose a shortcut icon and, if needed, specify the shortcut **Description**, **Hot key** and **Run** options. Click **Finish** to add the shortcut.



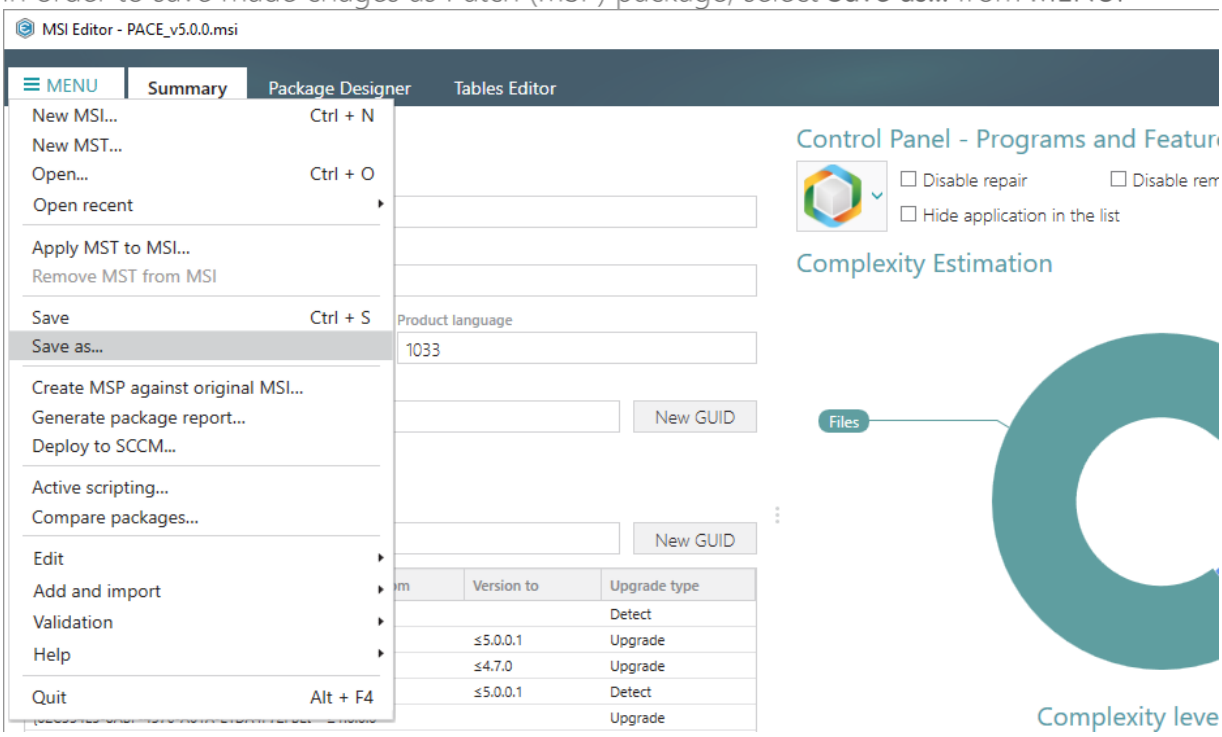
[30]. In order to import existing shortcuts from the file system to the package, select **Import shortcuts** from the context menu of a folder, where you want to import these shortcuts.



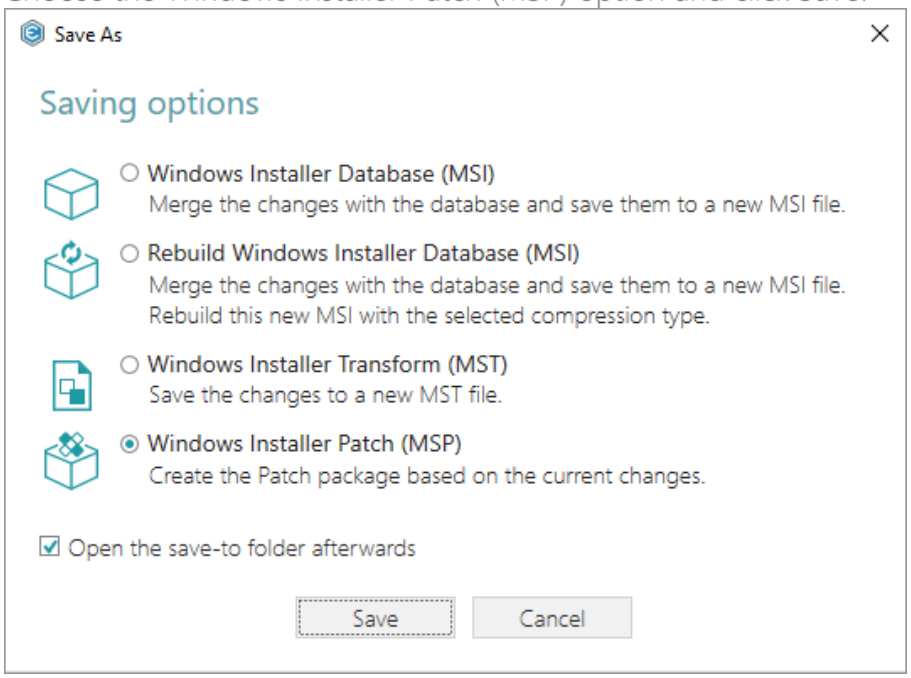

[31]. Choose shortcuts you would like to import and click **Open**. The imported shortcuts will be highlighted in the folders tree of the **Shortcuts** tab.



[32]. In order to save made chages as Patch (MSP) package, select Save as... from MENU.

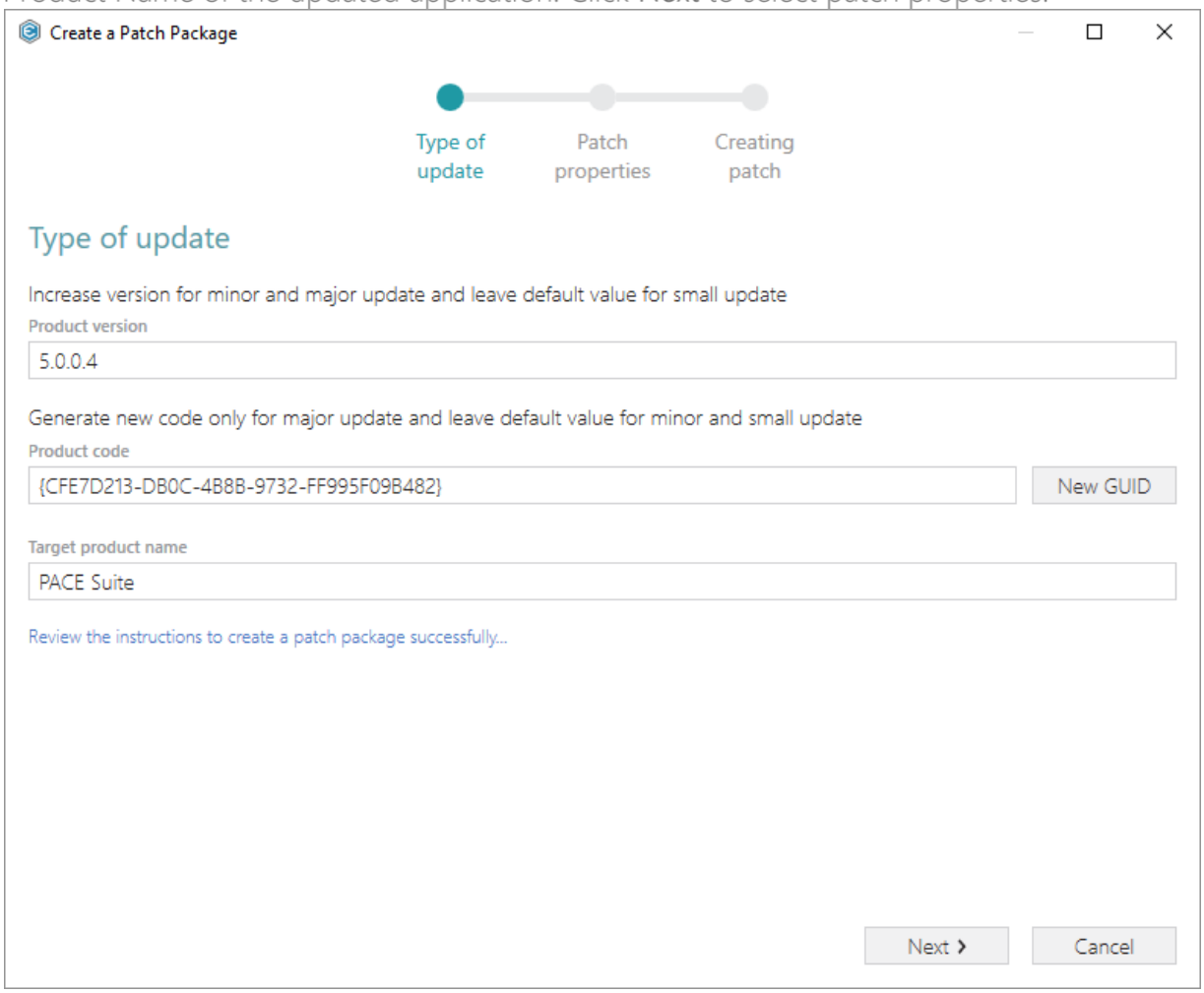



[33]. Choose the Windows Installer Patch (MSP) option and click Save.

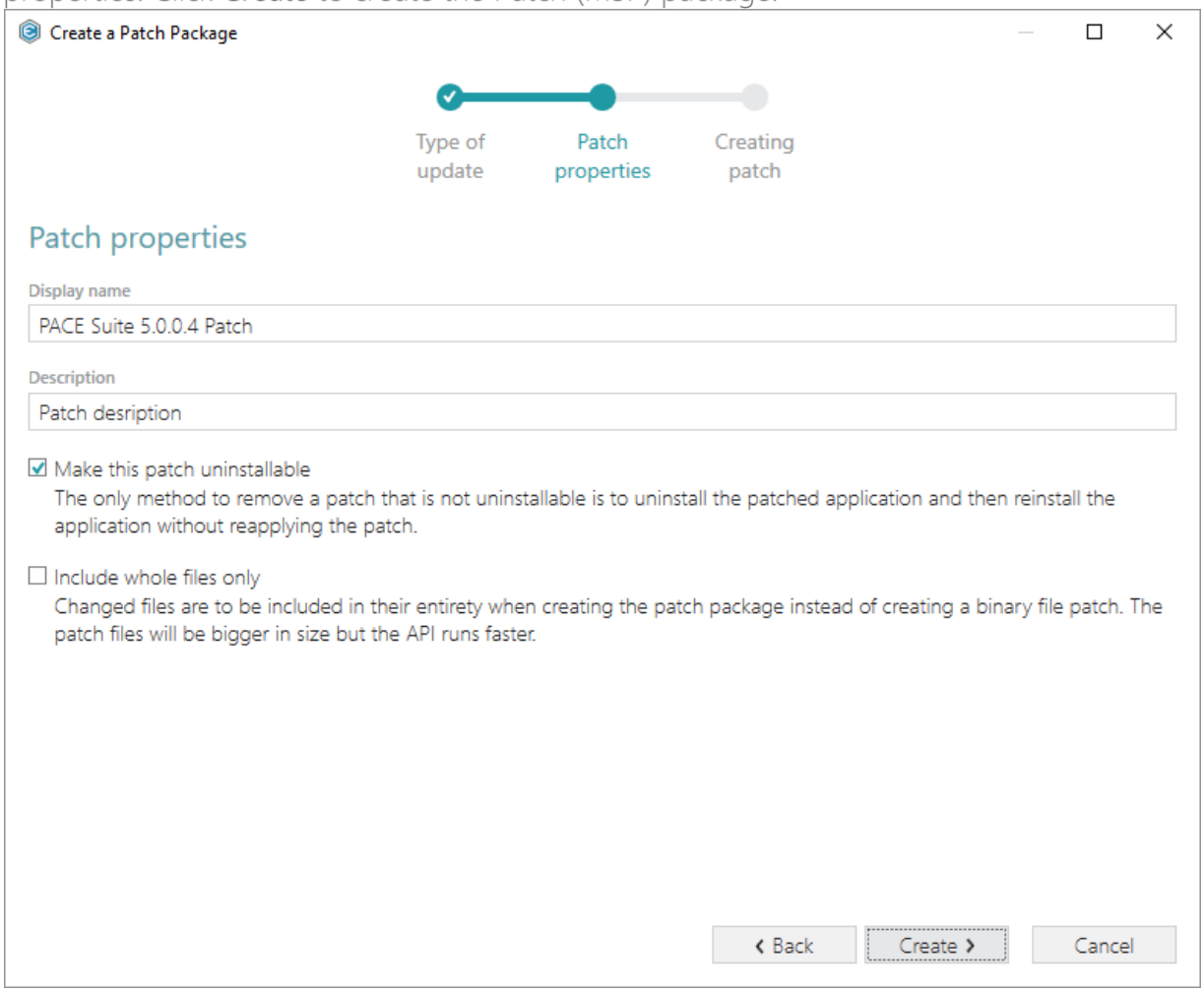



[34]. Increase a Product version for the minor and major update and leave default for the small update. Generate new Product code only for the major update and leave default for the minor and small update. If needed, update a Target product name in order to change the

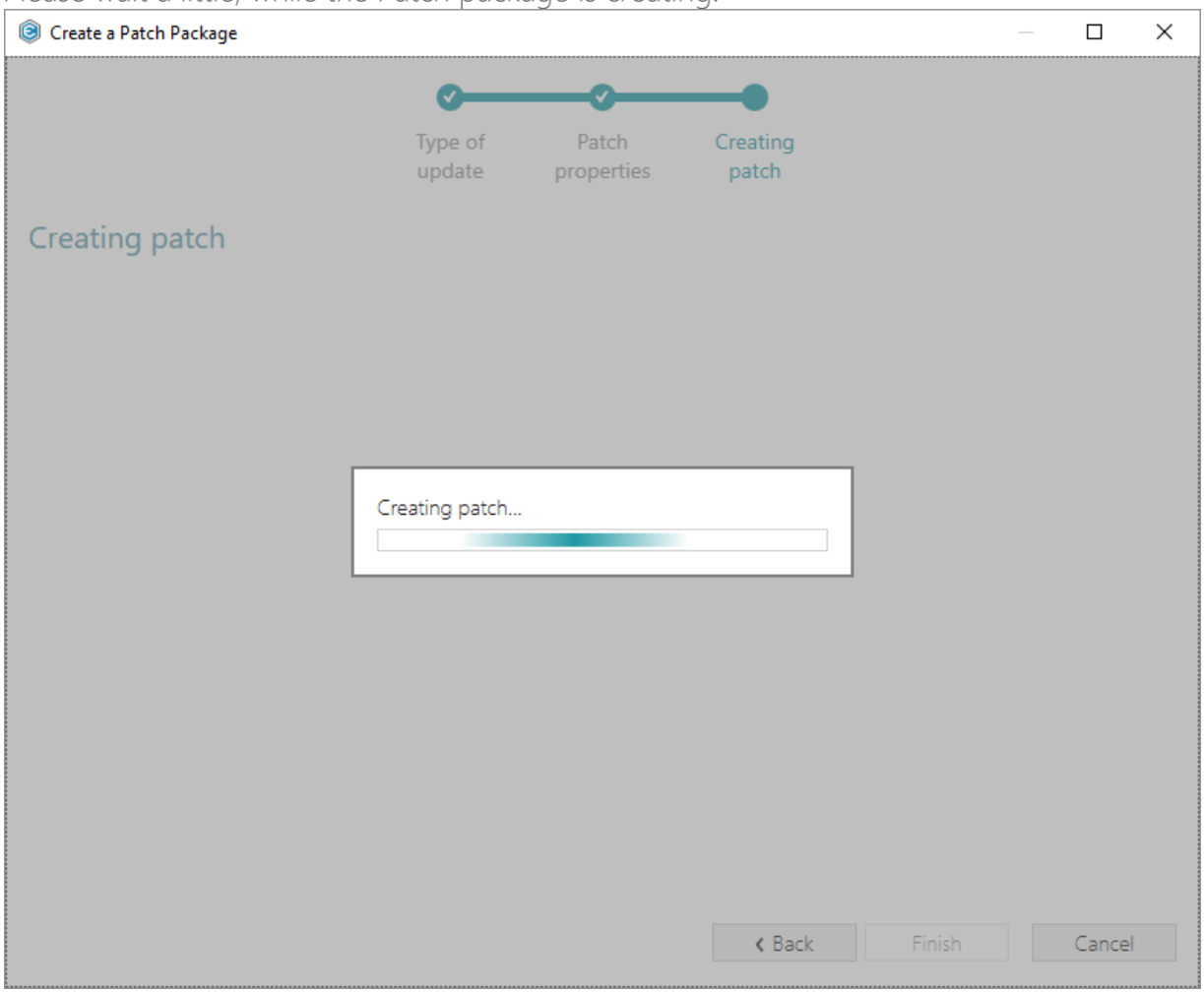
Product Name of the updated application. Click **Next** to select patch properties.



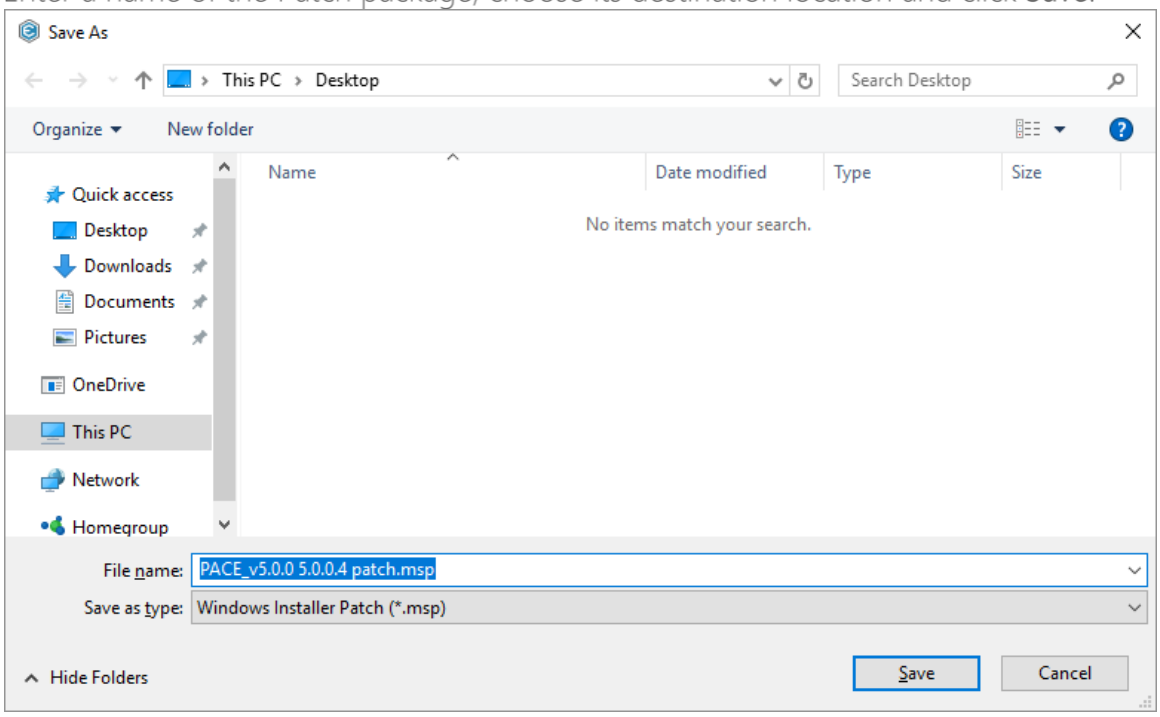
[35]. Update **Display name** and **Description** of the Patch package and select the necessary properties. Click **Create** to create the Patch (MSP) package.



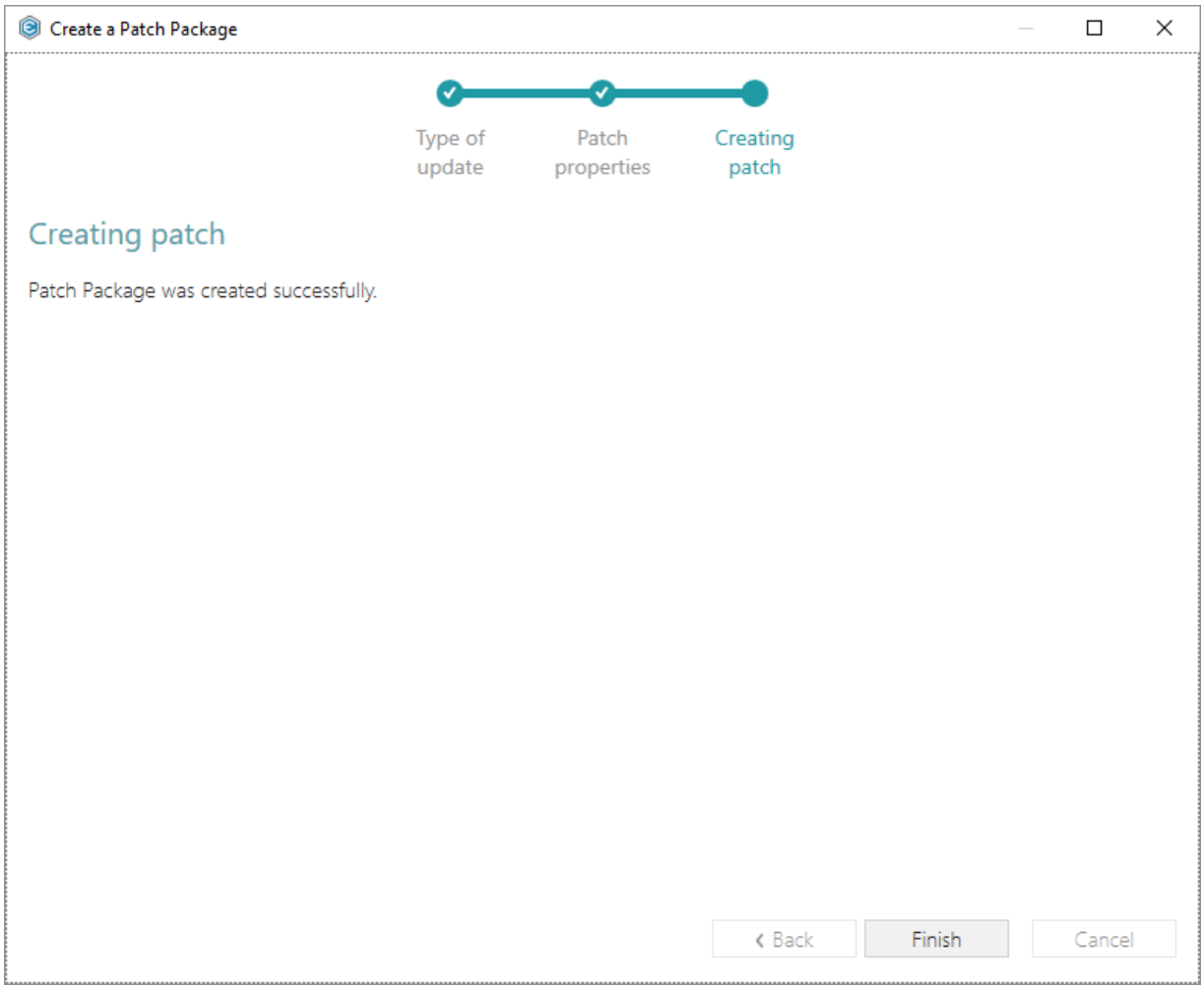
[36]. Please wait a little, while the Patch package is creating.




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



[38]. Click **Finish** to close the wizard.



## 3.4 Create APPX

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

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

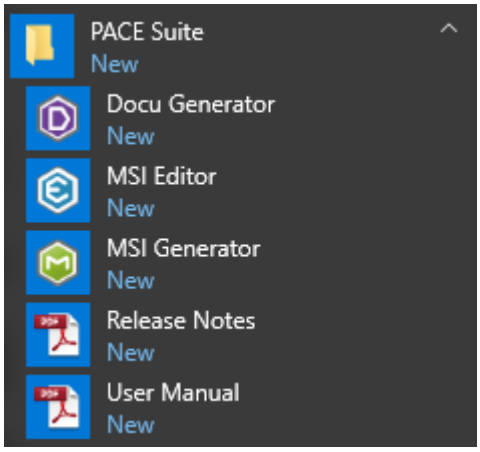
Choose a scenario that better suits your needs:

- Create APPX from Scratch, described in section 3.4.1
  Create a new empty project in MSI Generator, add folders, files, registry entries and APPX apps (shortcuts) to this project and then generate the Universal Windows Platfrorm app package (APPX) from this project.
- Convert (Repackage) EXE, MSI to APPX, described in section 3.4.2
  Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into Universal Windows Platfrorm app package (APPX) using MSI Generator.

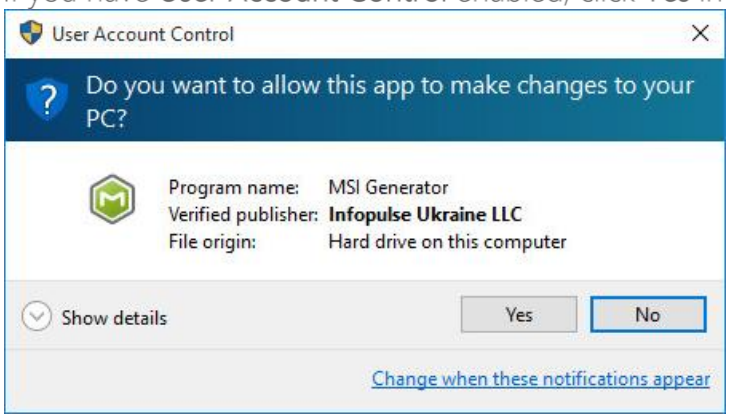
### 3.4.1 Create APPX from Scratch

Create a new empty project in MSI Generator, add folders, files, registry entries and APPX apps (shortcuts) to this project and then generate the Universal Windows Platfrorm app package (APPX) from this project.

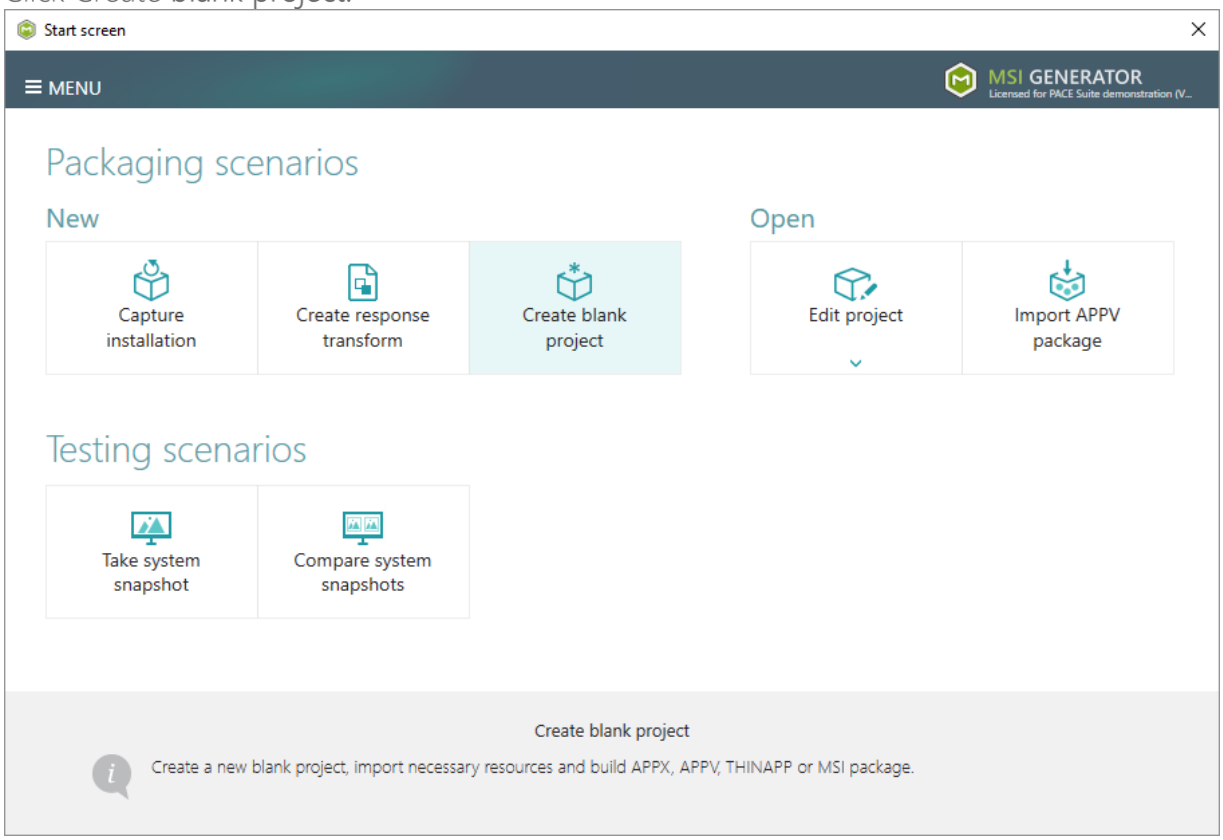
[1]. Launch MSI Generator from the desktop or the start menu shortcut.



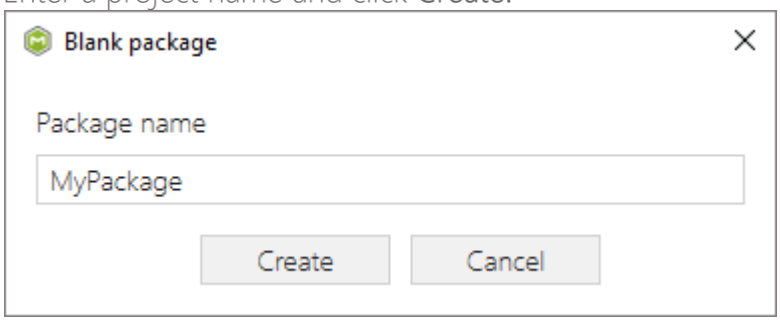
[2]. If you have User Account Control enabled, click Yes in the opened window.




[3]. Click Create blank project.



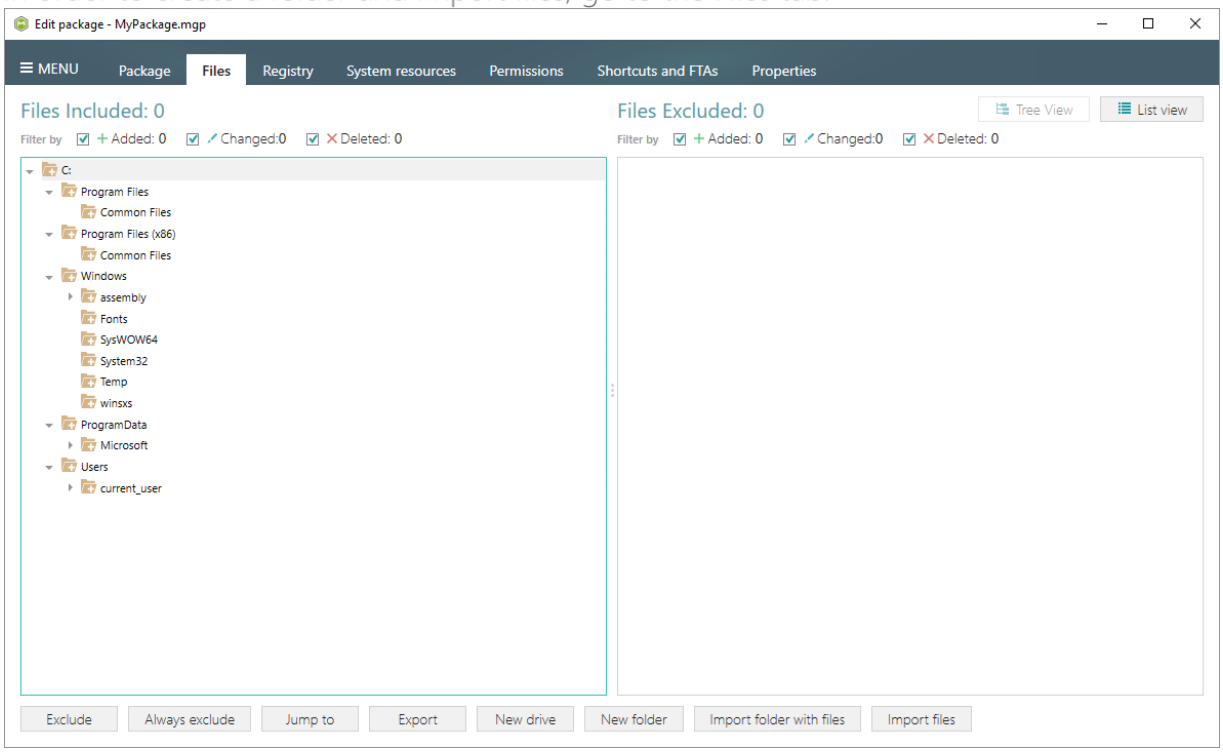

[4]. Enter a project name and click Create.



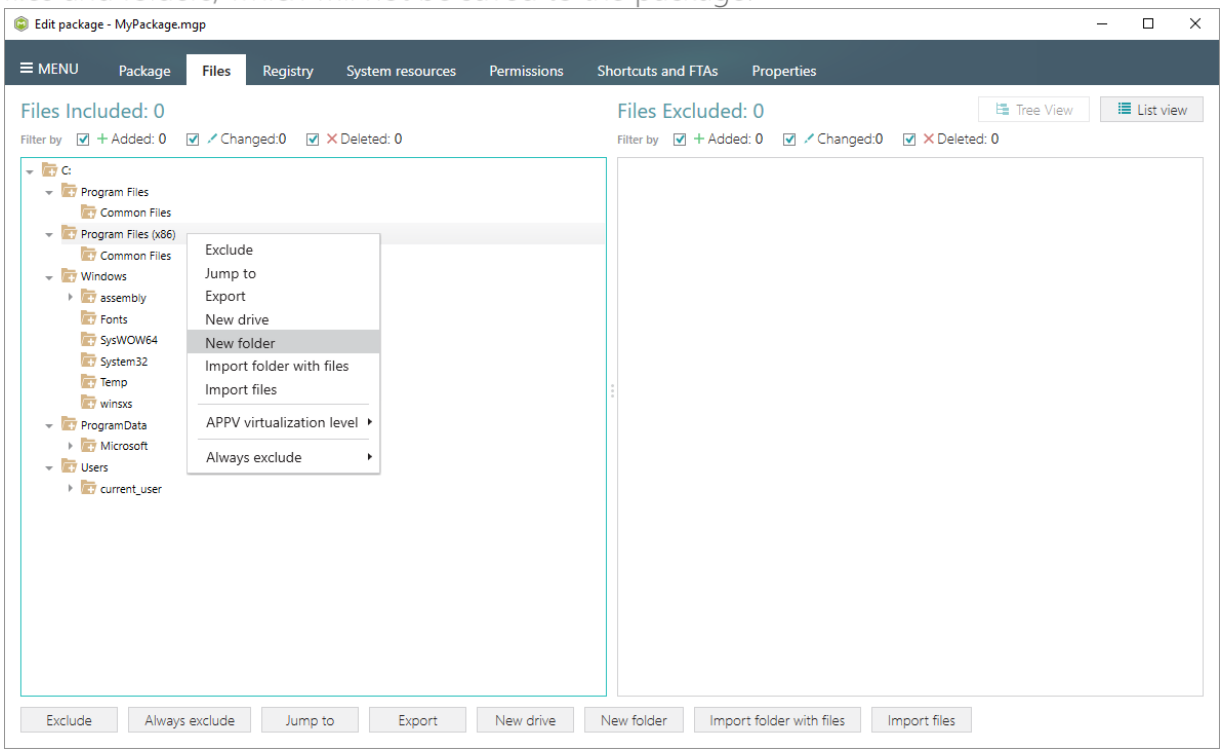

[5]. Once the newly created project is opened automatically, you can start adding the necessary resources using MSI Generator. The steps from [6] to [10] describe how to create a new folder and import files to this folder; the steps from [11] to [13] describe how to import registry entries from the REG file; the steps from [14] to [16] describe how to add

APPX app (shortcut) and its logos; and two last steps [17] and [19] describe how to build APPX package from the prepared project.

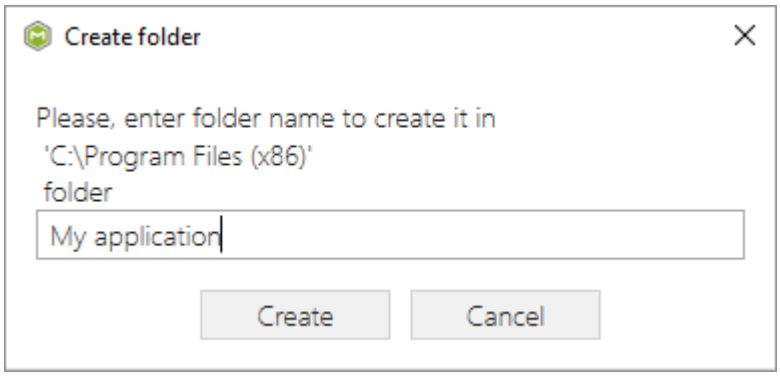
[6]. In order to create a folder and import files, go to the **Files** tab.



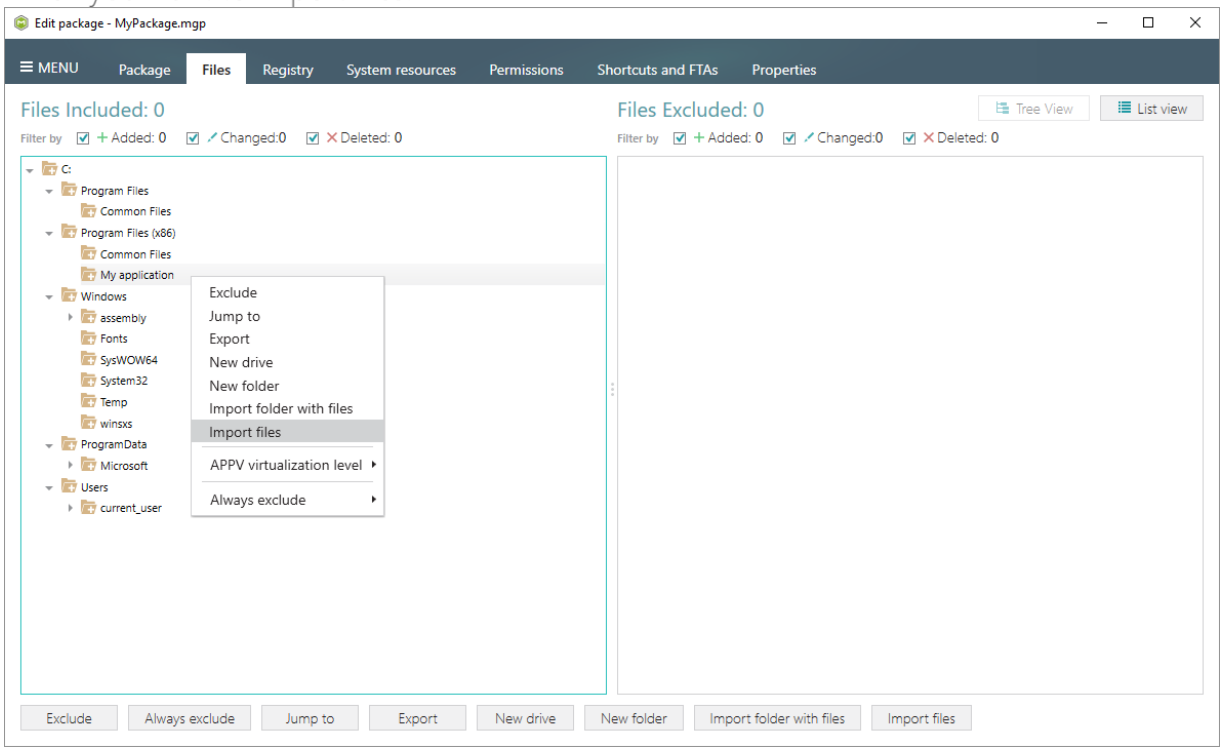

[7]. In the left pane, select **New folder** from the context menu of a folder, where you want to create a new empty folder. Note that the left '**Files Included**' pane displays files and folders, which will be a part of your package, and the right '**Files Excluded**' pane displays files and folders, which will not be saved to the package.



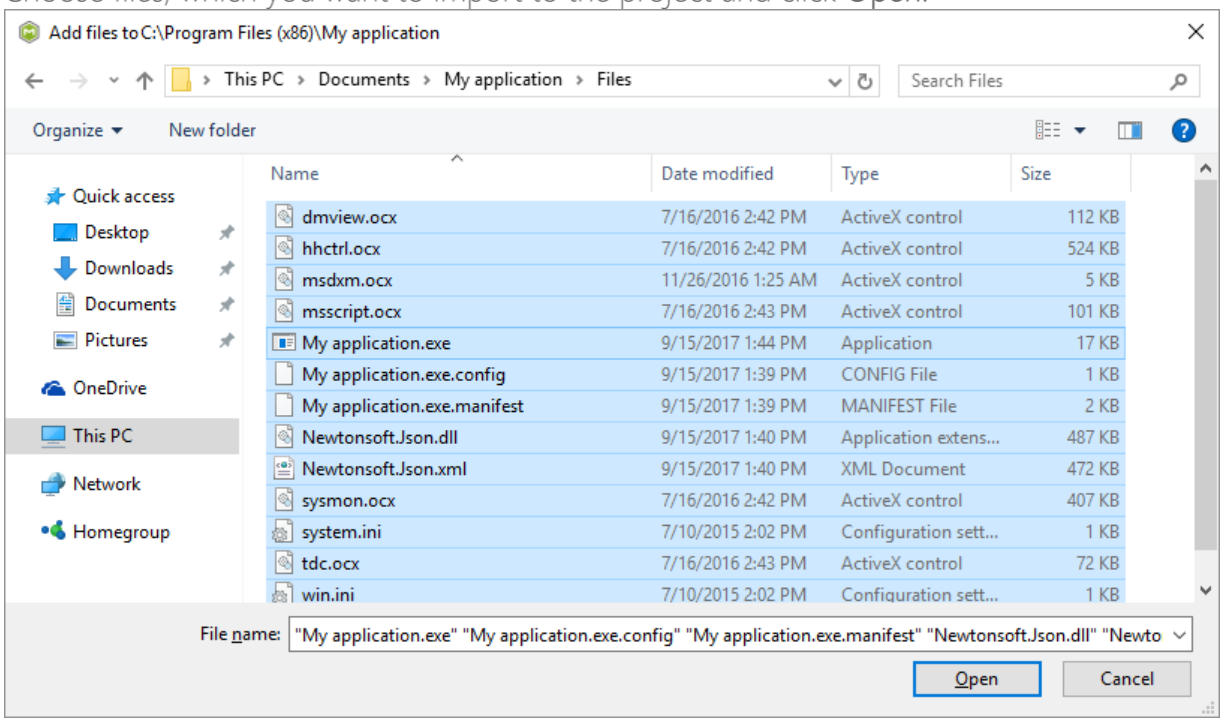
[8]. Enter a folder name and click **Create**.



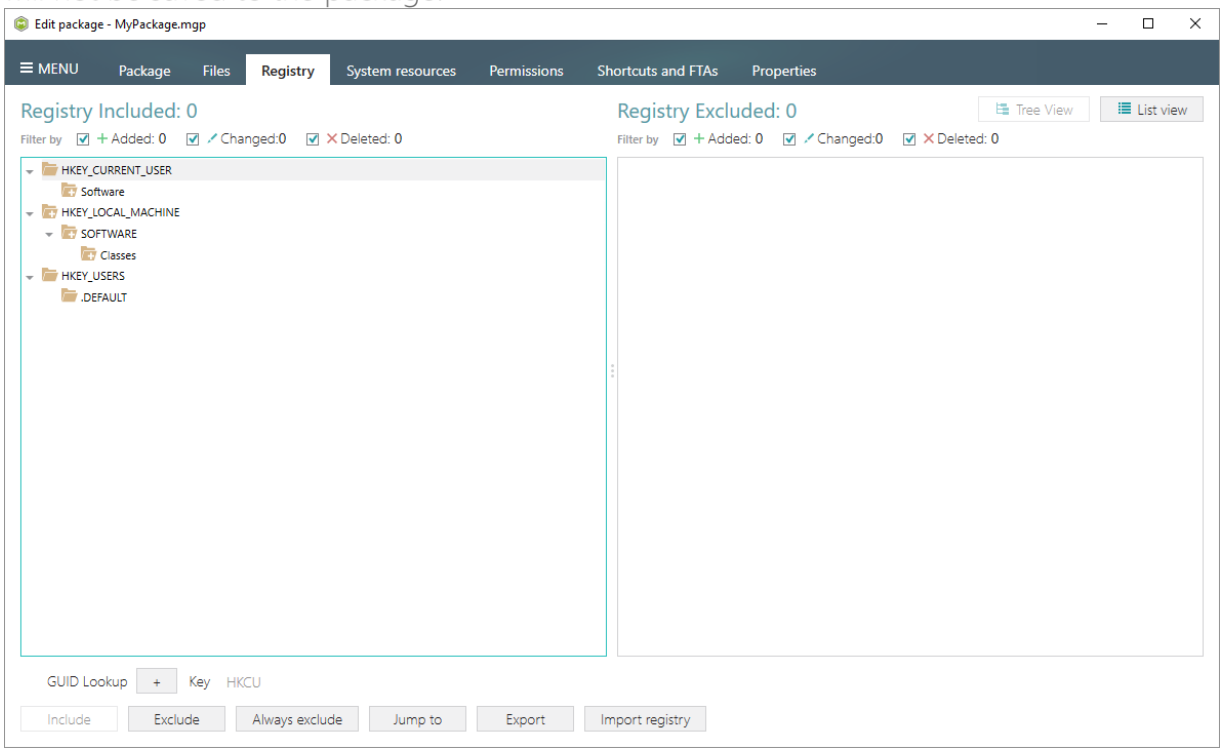

[9]. For adding files to your project, select **Import files** from the context menu of a folder, to which you want to import files.



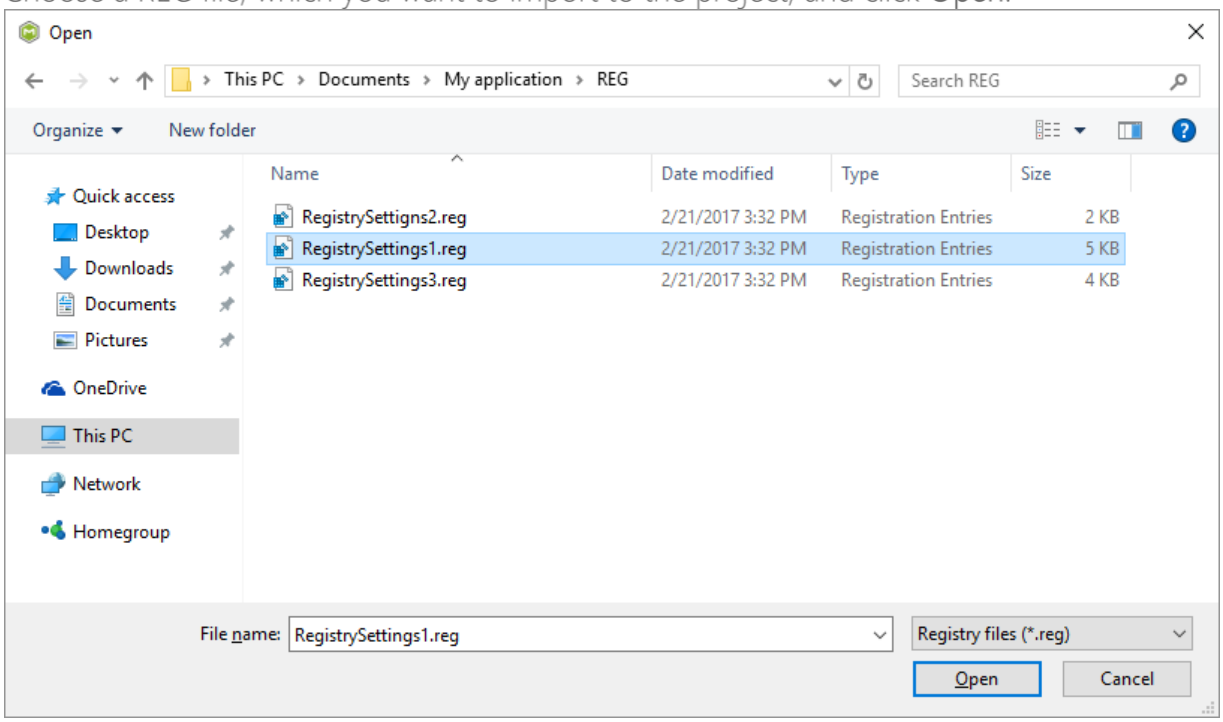
[10]. Choose files, which you want to import to the project and click **Open**.



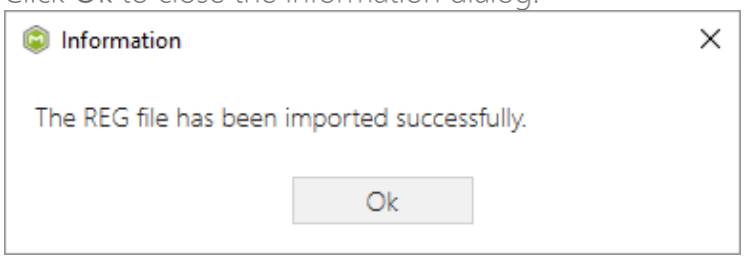

[11]. In order to import registry entries from the REG file, go to the **Registry** tab and click the **Import registry** button, located in the bottom part of the window. Note that the left '**Registry Included**' pane displays registry keys and values, which will be a part of your package, and the right '**Registry Excluded**' pane displays registry keys and values, which will not be saved to the package.



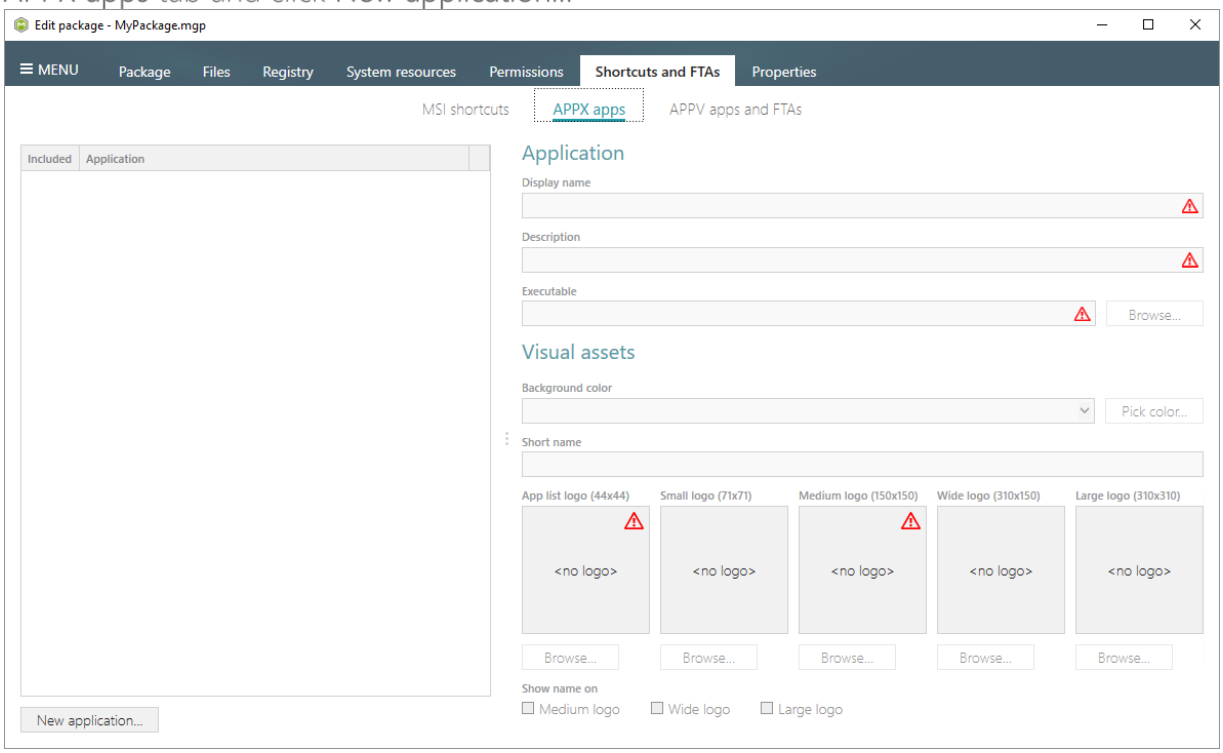
[12]. Choose a REG file, which you want to import to the project, and click Open.



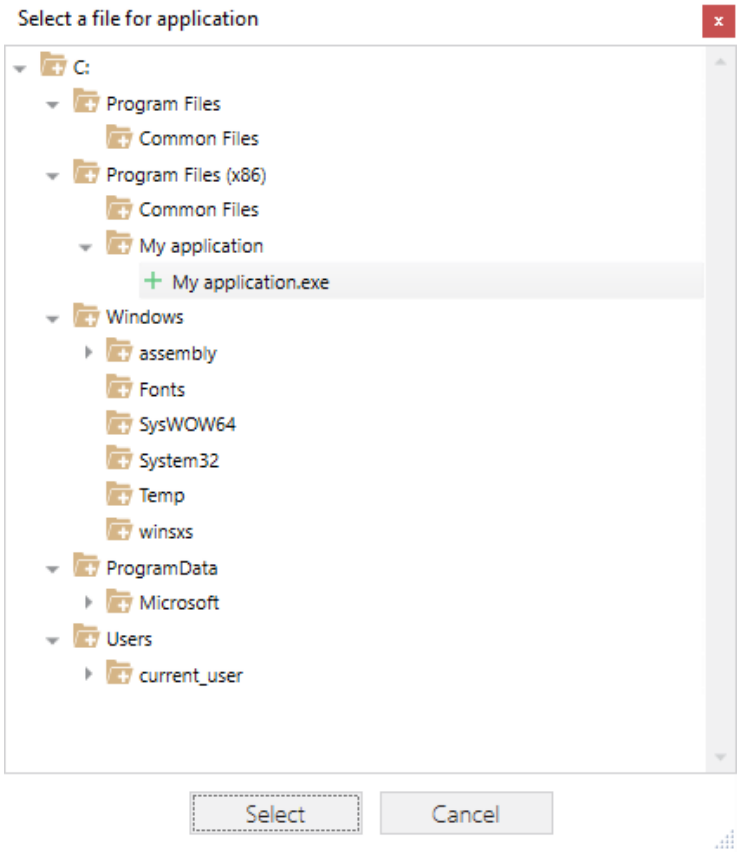

[13]. Click Ok to close the information dialog.



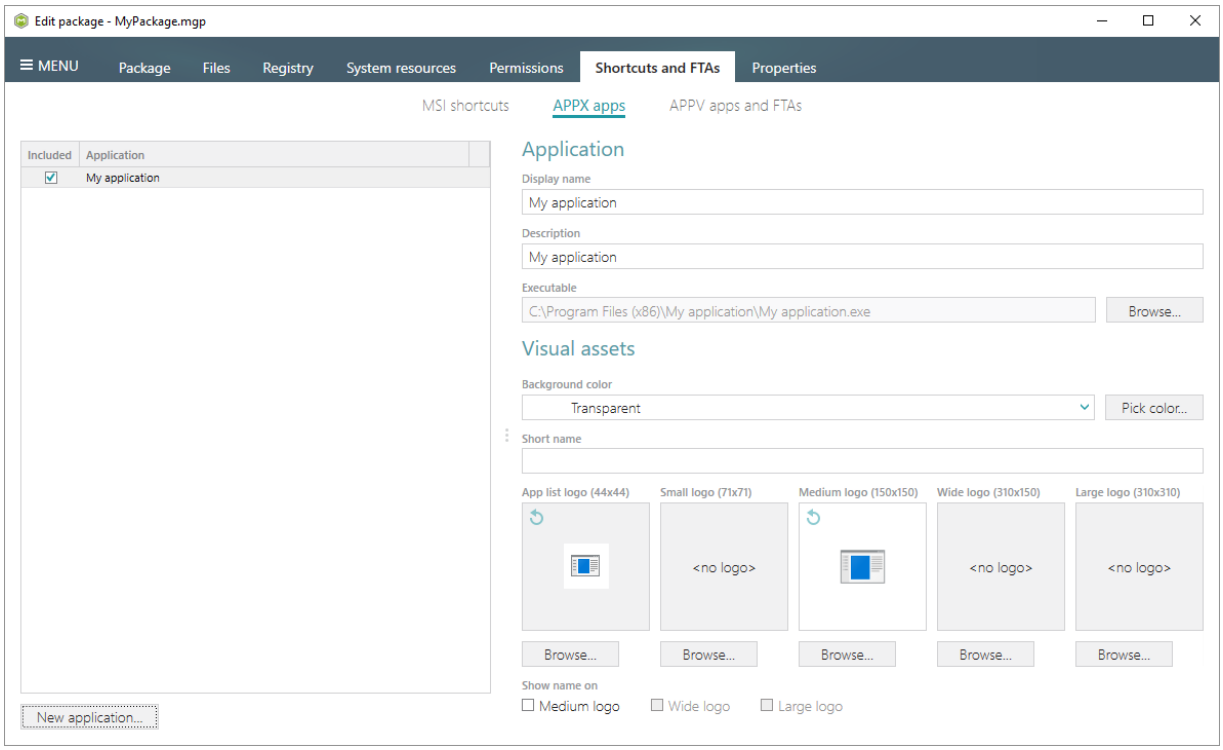

[14]. If you need to add an APPX app (shortcut) and its logos, go to the Shortcuts and FTAs -> APPX apps tab and click New application...



[15]. Select the main executable file (EXE) of your application from the tree and click **Select**.

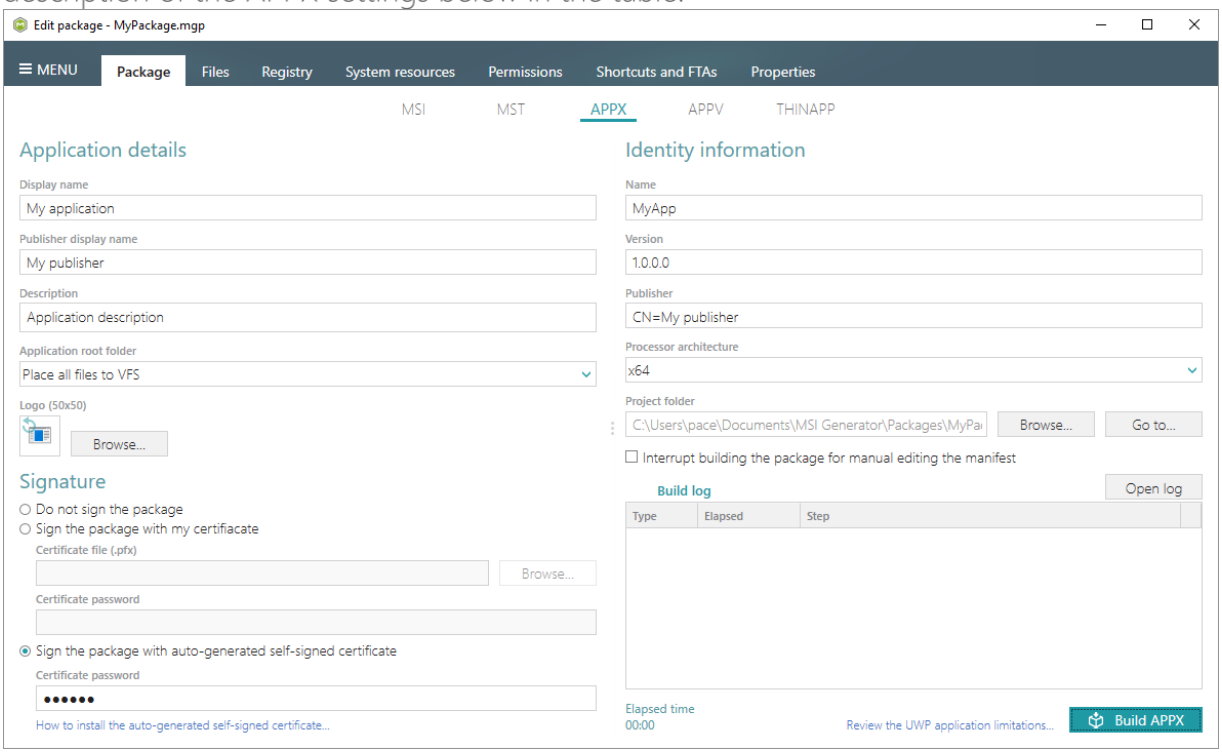



[16]. Using the details pane, you can update the APPX app display name, description, and visual assets such as background color and logos, which will be displayed in the Start menu.




[17]. In order to build APPX package from your project, navigate to the **Package** -> **APPX** tab, fill-in all the required fields, select the digital signing option and click **Build APPX**. Find

description of the APPX settings below in the table.




## Application details

| | |
|---|---|
| Display name | A friendly name of the package that can be displayed to users. |
| Publisher display name | A friendly name for the publisher of the package that can be displayed to users. |
| Description | A friendly description that can be displayed to users. |
| Application root folder | A location of the main application folder. Content of the selected folder will be placed to the package root, instead of VFS. |
| Logo (50x50) | A logo image of the package that can be displayed to users. |

## Signature

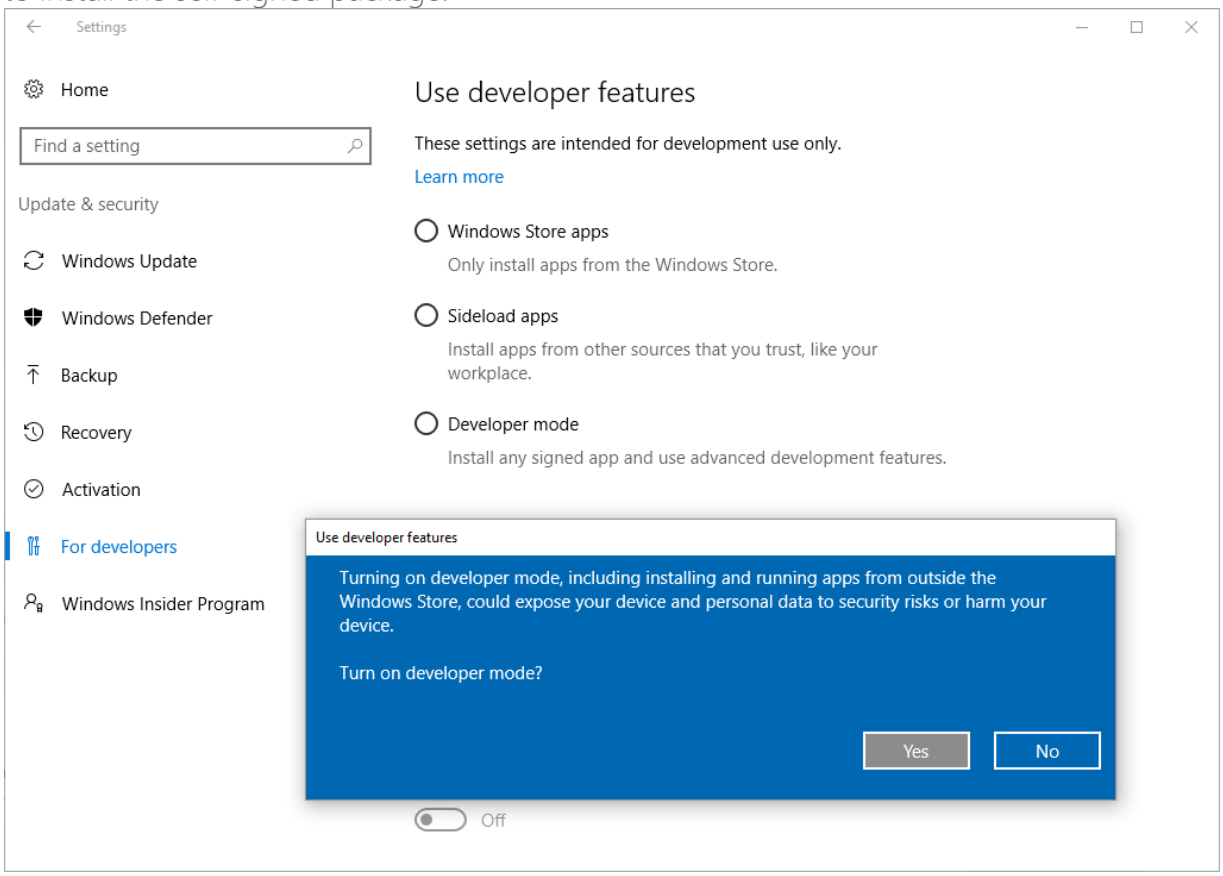
| | |
|---|---|
| Do not sign the package | Use this option if you are planning to submit the package to Windows Store or sign it manully. |
| Sign the package with my certificate | Use this option to sign the generated package with the valid certificate that you already have. |
| Sign the package with auto-generated self-signed certificate | Use this option to generate the self-signed certificate and sign the package with this certificate.<br>NOTE In order to deploy your package, install the generated certificate before (double-click the auto-generated.pfx file, and in the Certificate Import Wizard, install the certificate onto the Local Machine, and place the certificate into the Trusted People certificate store) and enable the Developer mode in Windows settings, as described in the next step. |

## Identity information

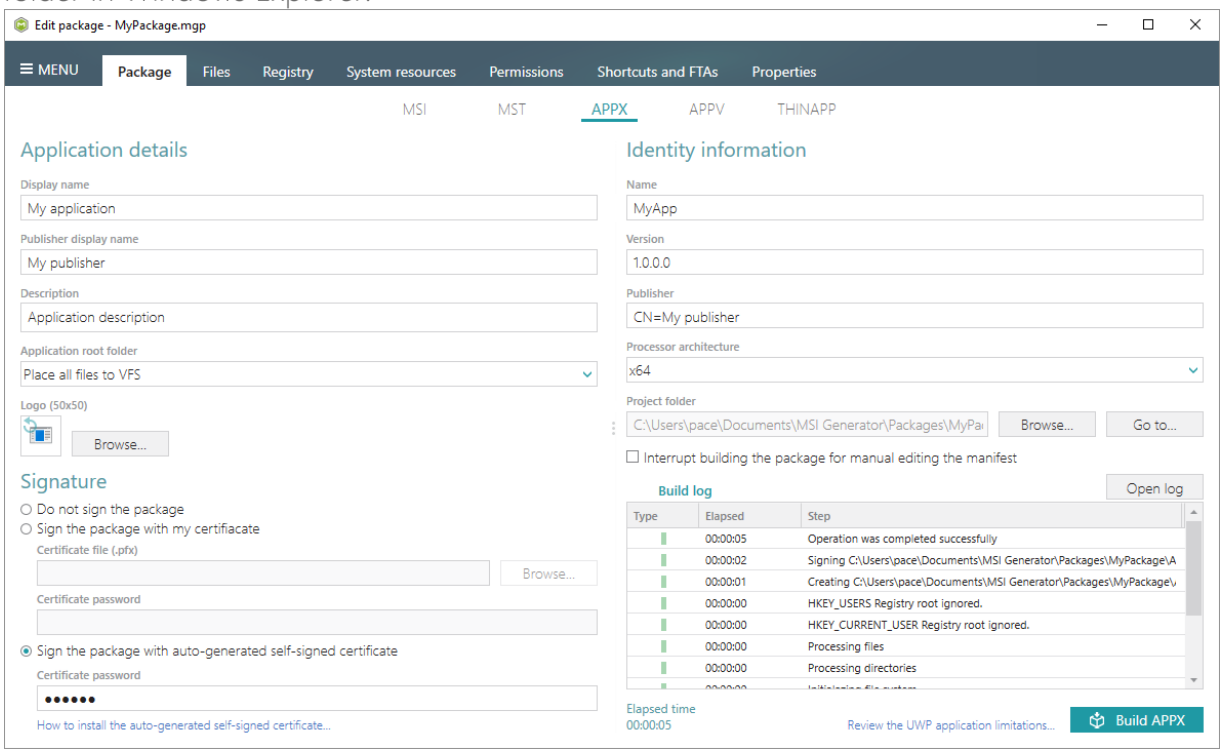
| | |
|---|---|
| Name | Describes the contents of the package. |
| Version | The version number of the package. |

| | |
|---|---|
| Publisher | Describes the publisher information. The Publisher attribute must match the publisher subject information of the certificate used to sign a package. |
| Processor architecture | Describes the architecture of the code contained in the package. |

[18]. If you have built the package with the **Sign the package with the auto-generated self-signed certificate** option selected, do not forget to install that generated certificate to the **Trusted People** certificate store and enable the **Developer mode** in the Windows settings to install the self-signed package.



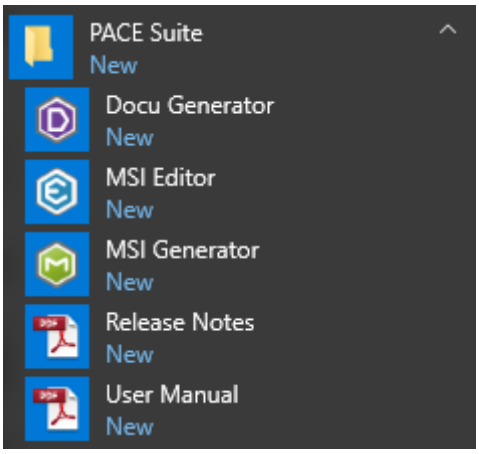
[19]. Click Go to..., located next to the Project folder field, to open the package containing folder in Windows Explorer.



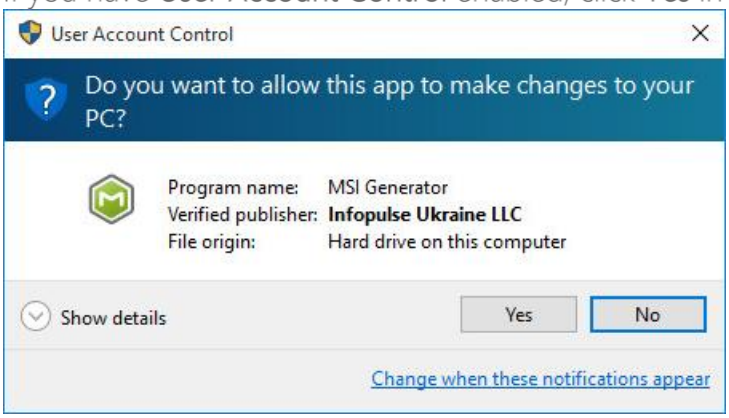

## 3.4.2 Convert (Repackage) EXE, MSI to APPX

Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into Universal Windows Platform app package (APPX) using MSI Generator.

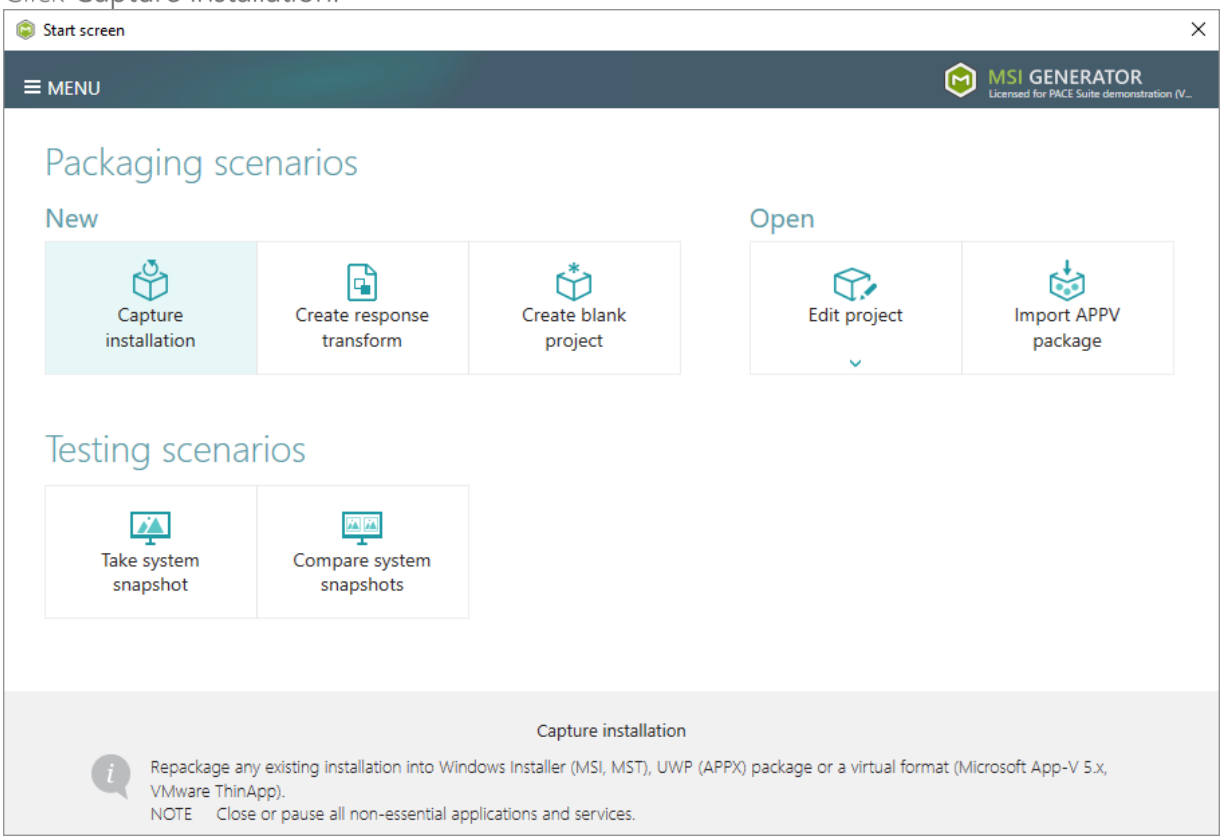
- Launch MSI Generator from the desktop or the start menu shortcut.



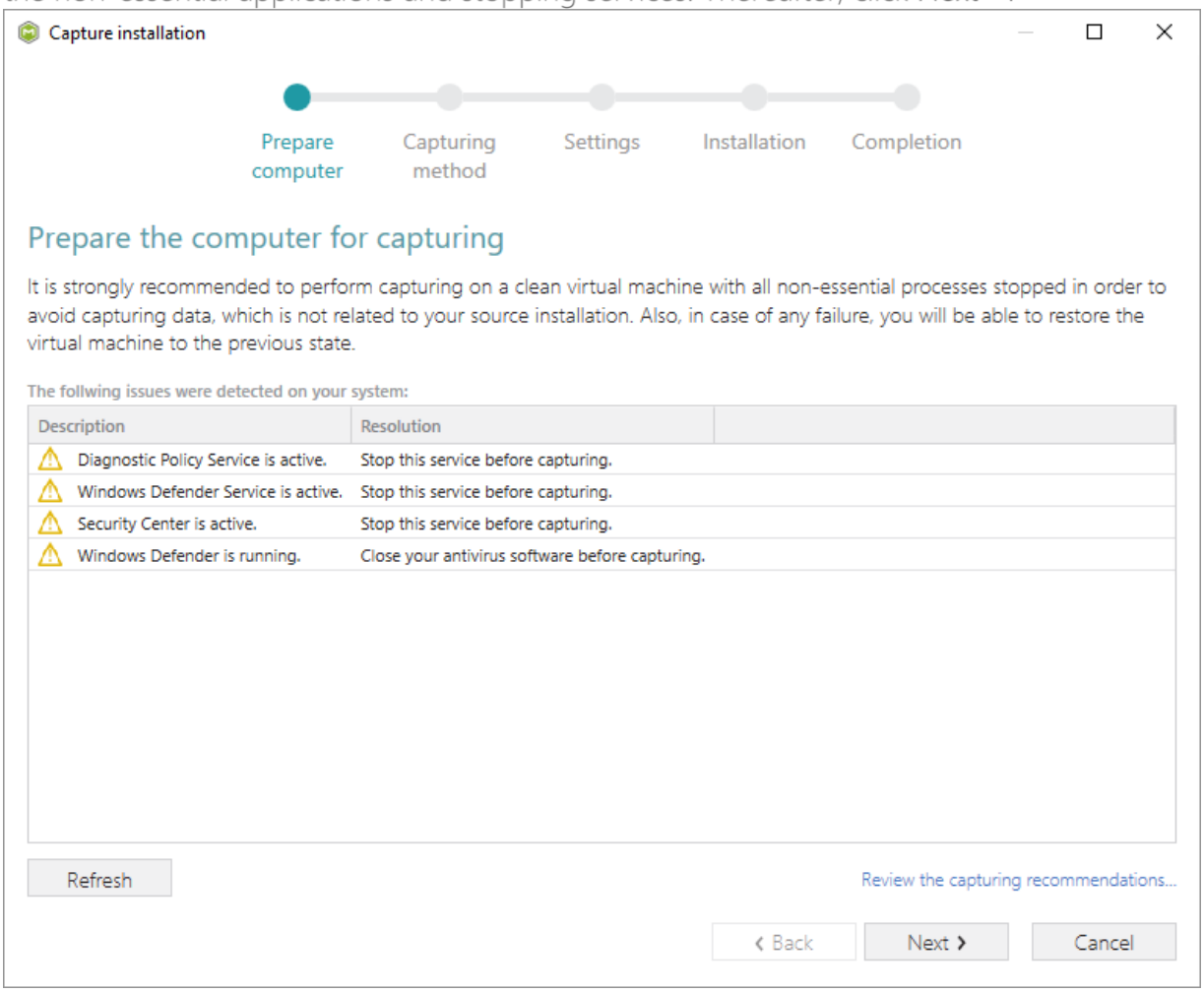
- If you have User Account Control enabled, click Yes in the opened window.



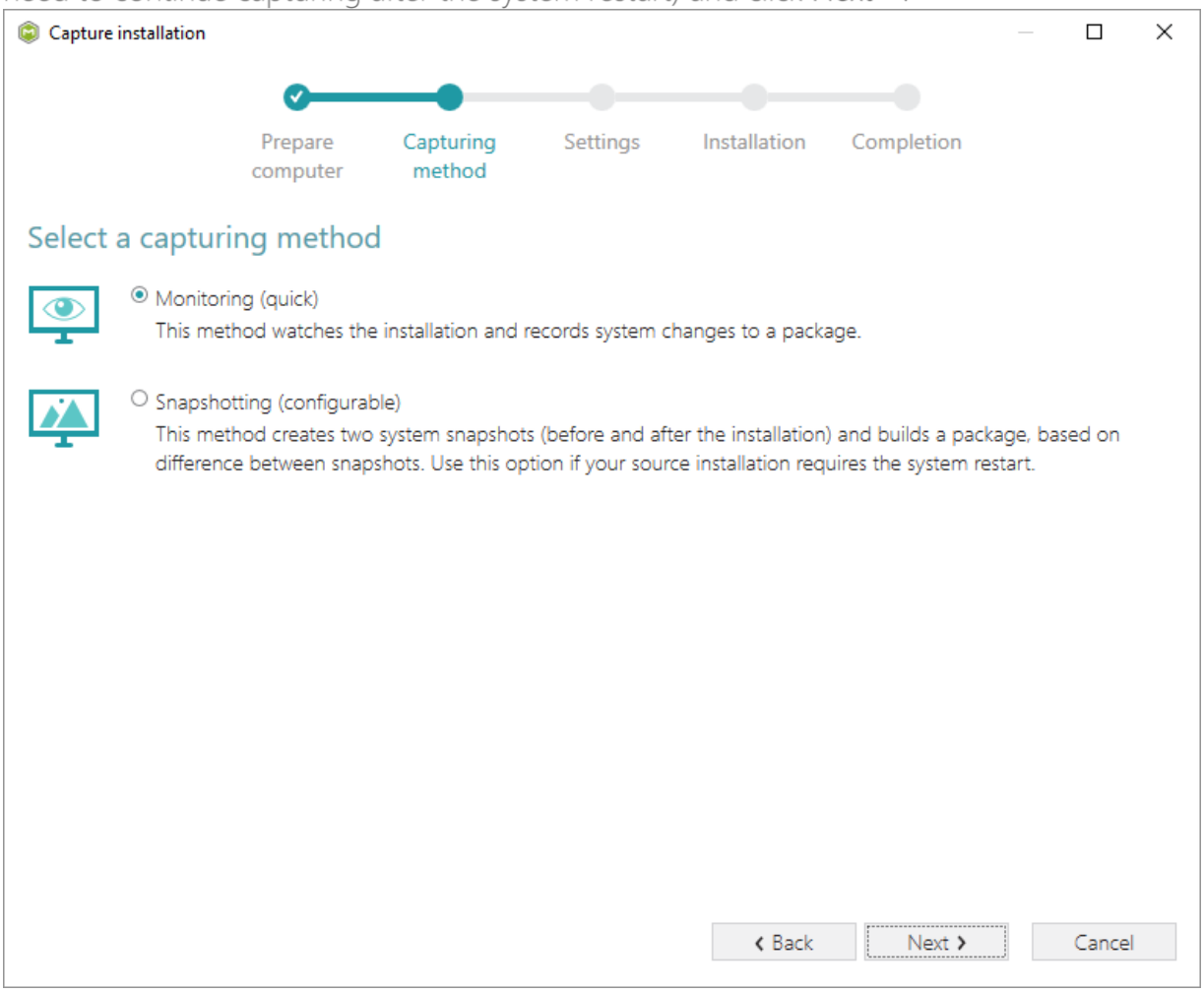

- Click Capture installation.



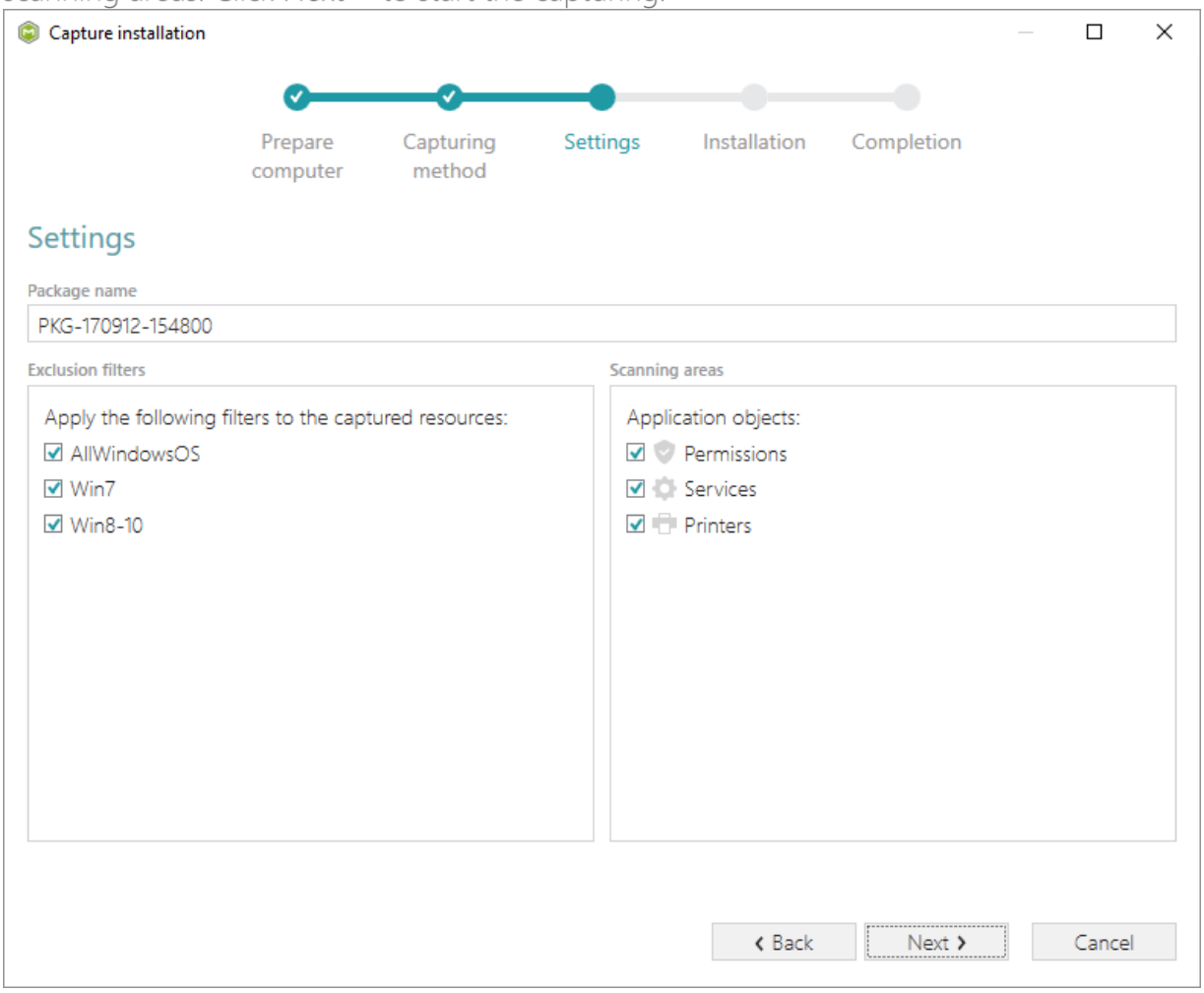
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



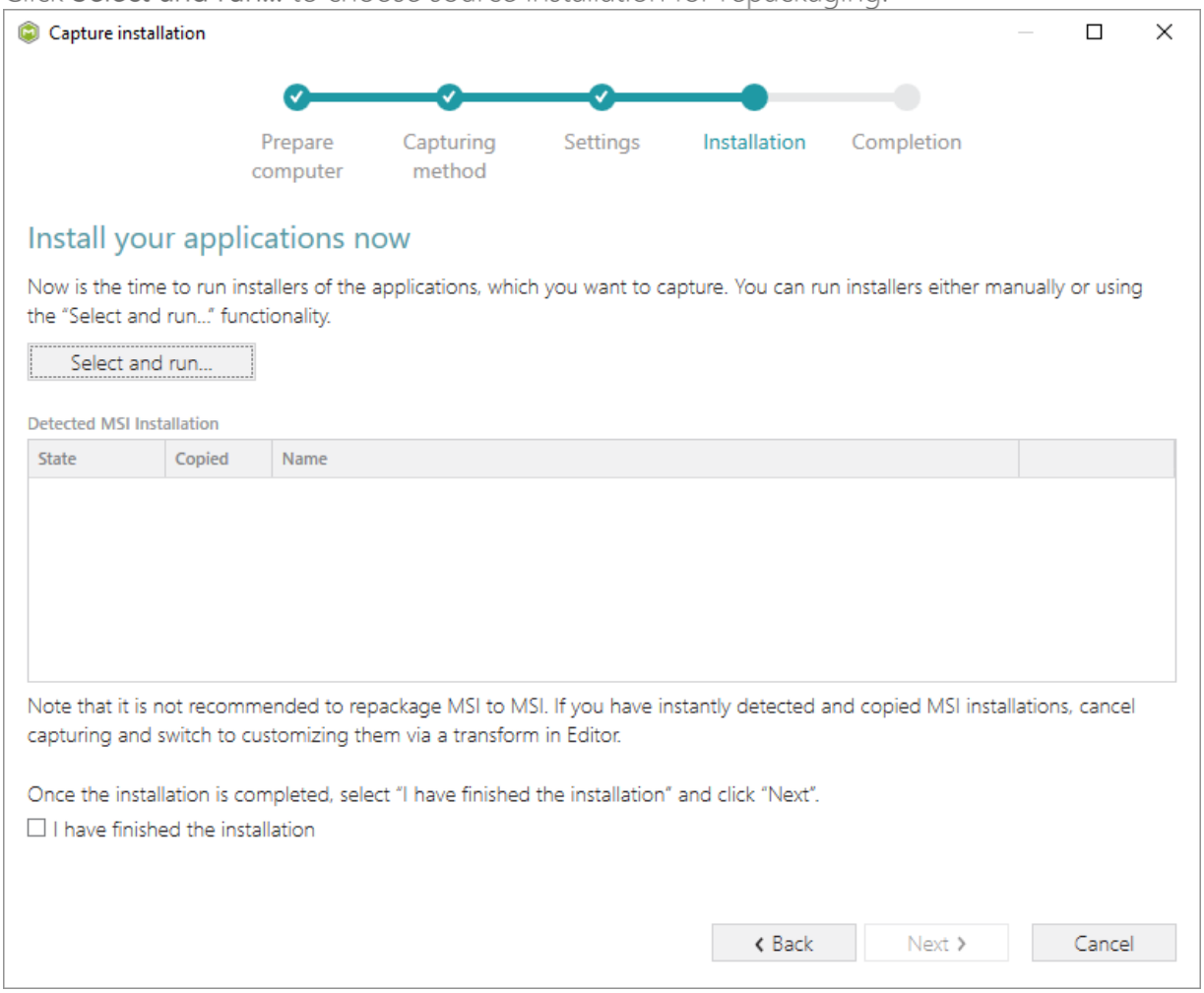
- Select the Monitoring method for the quicker capturing (or use the Snapshotting one if you need to continue capturing after the system restart) and click Next >.



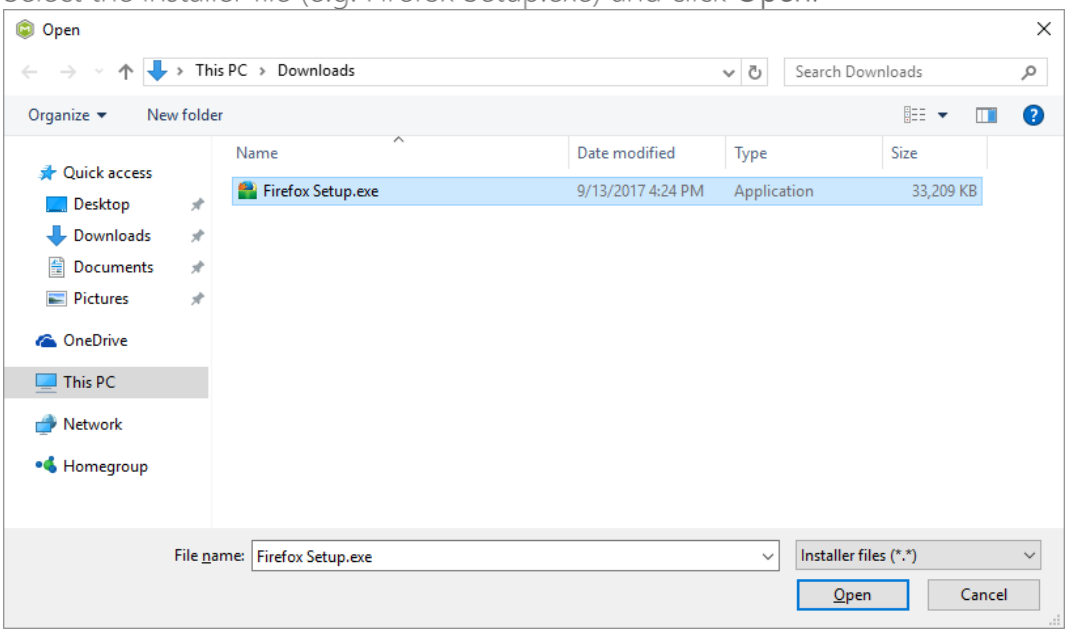
- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.



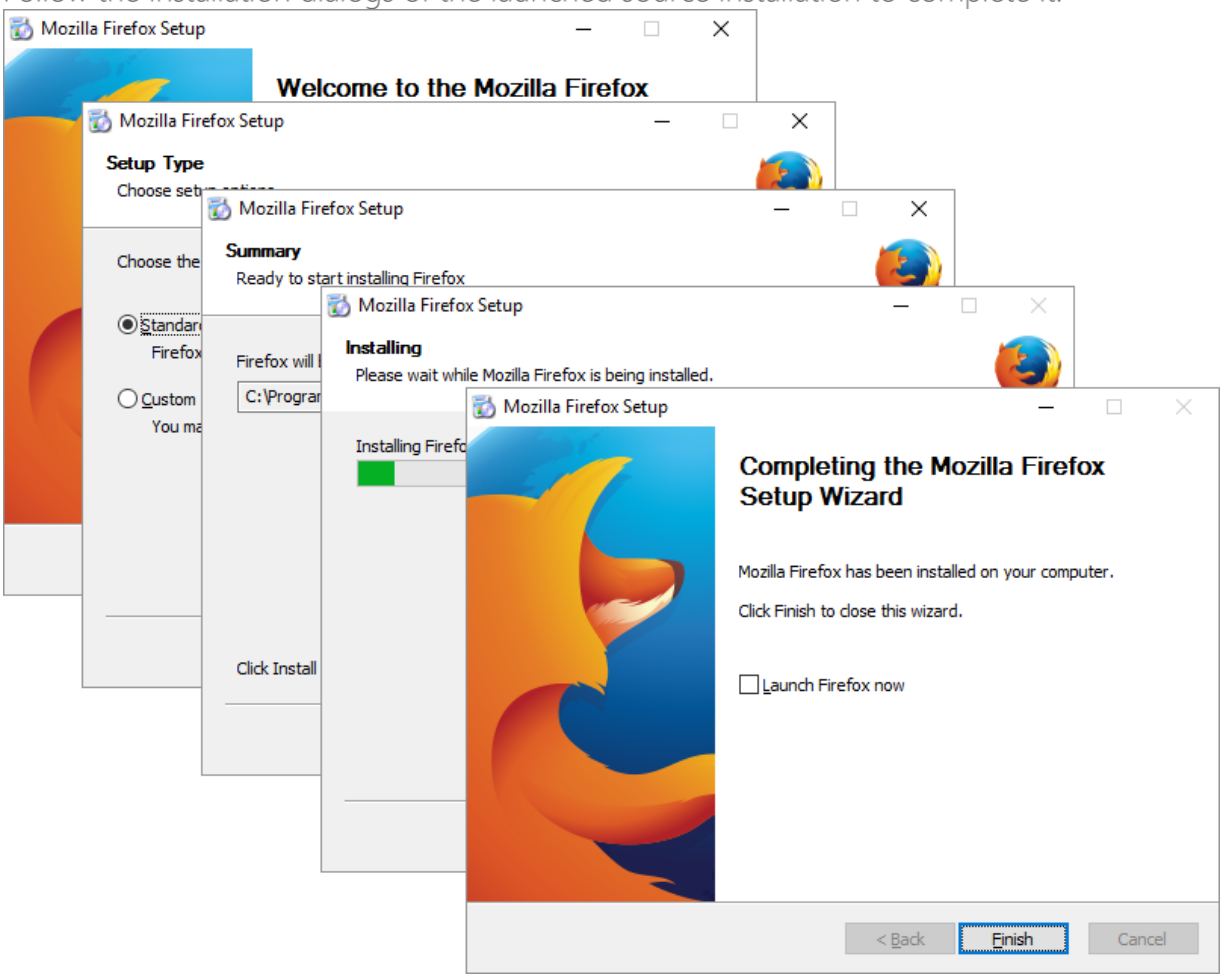
- Click Select and run... to choose source installation for repackaging.




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

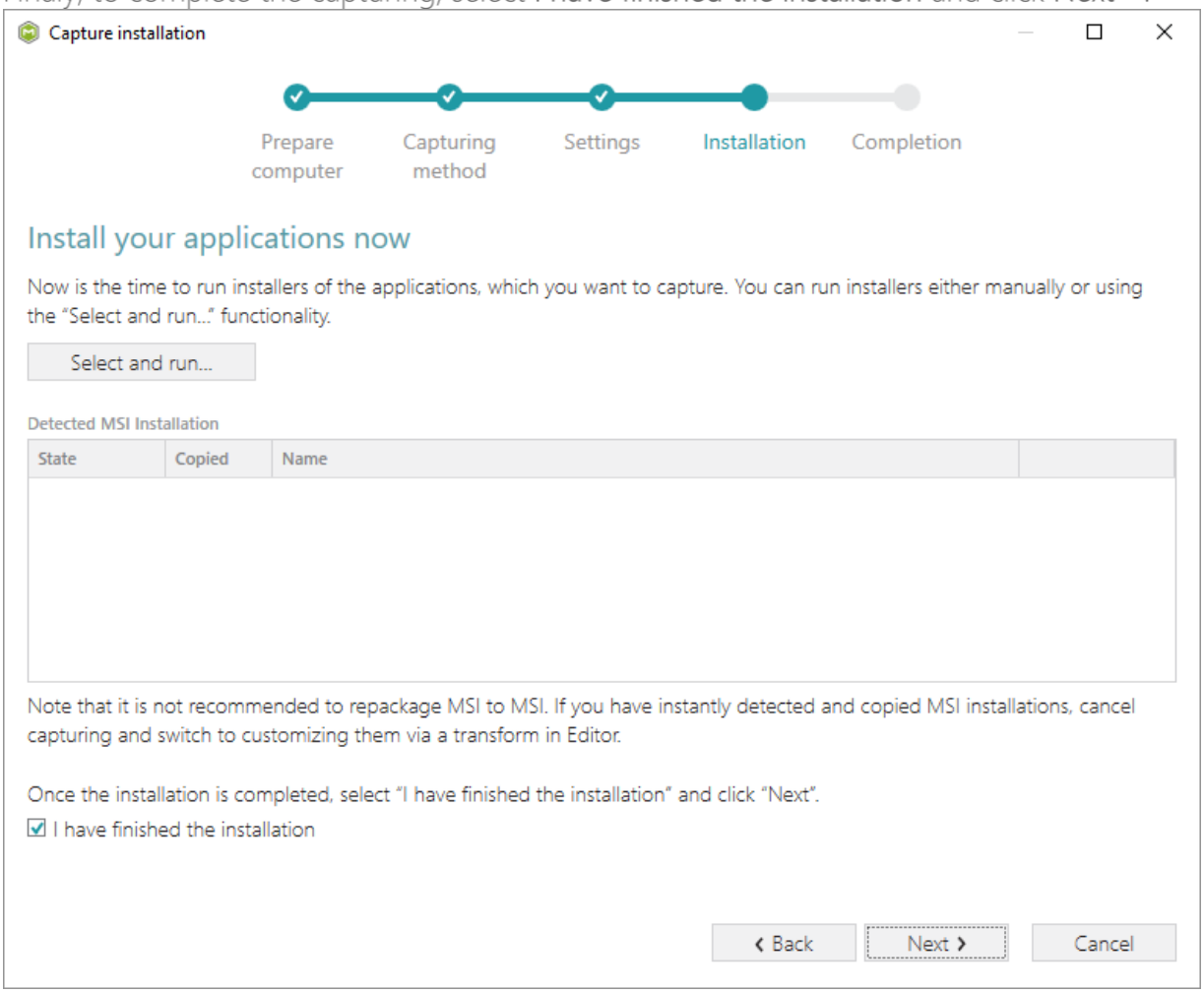


- Follow the installation dialogs of the launched source installation to complete it.

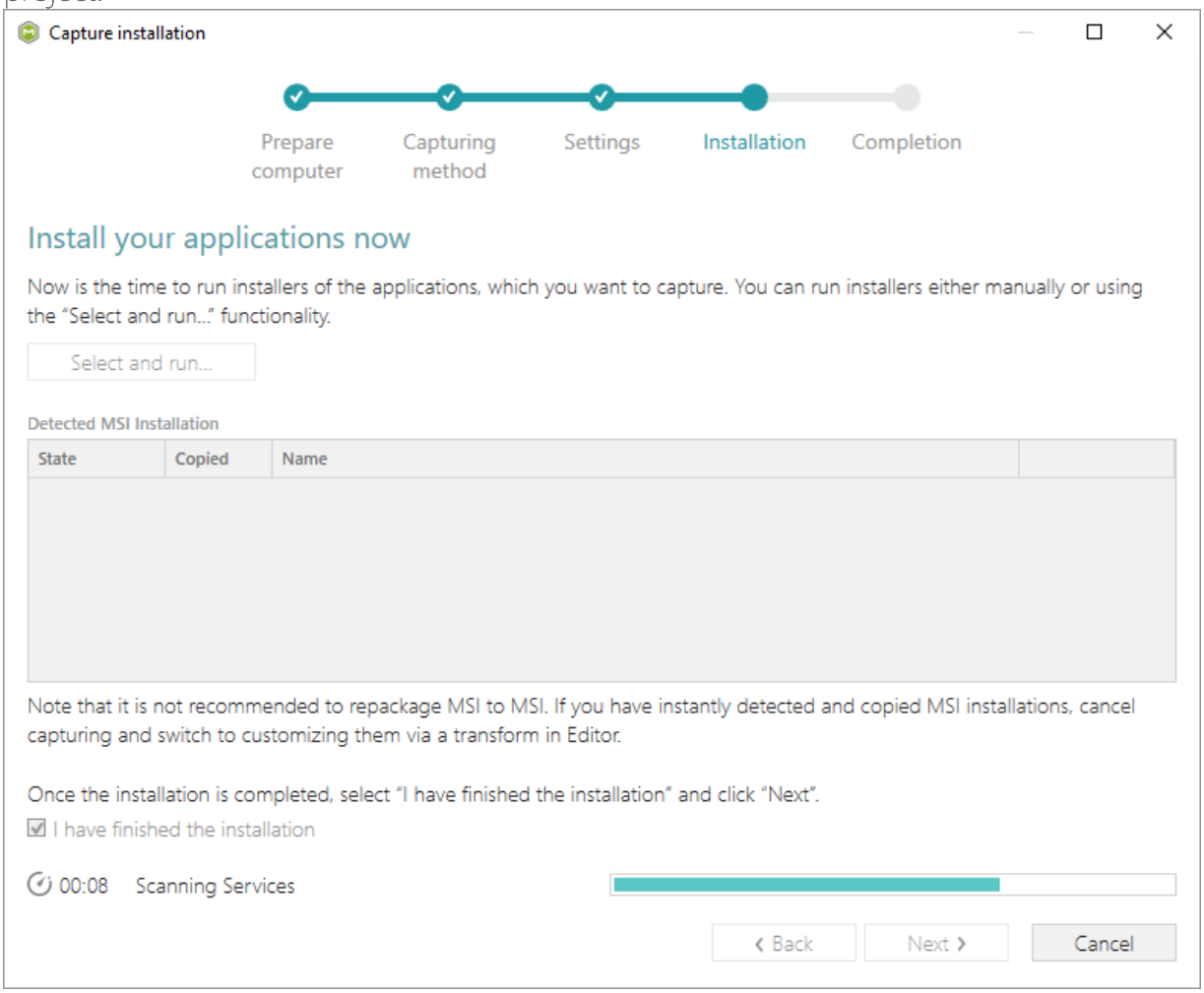



- Now you can make any changes to the file system and registry, which you want to capture and include to the package. For instance, you can create new or copy-paste existing files, import REG file to the system registry, changes permission settings, or launch the installed application in order to capture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

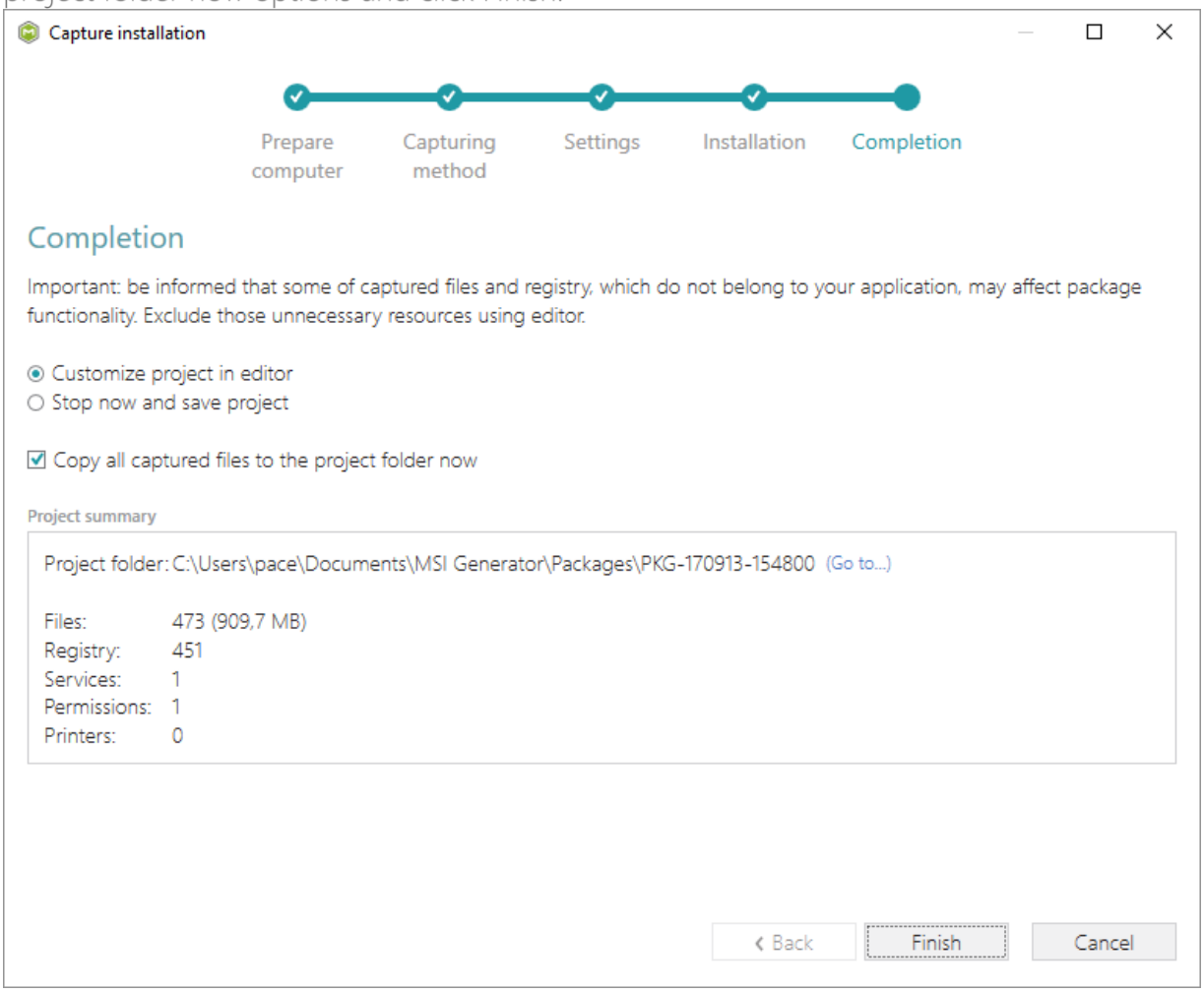
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



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

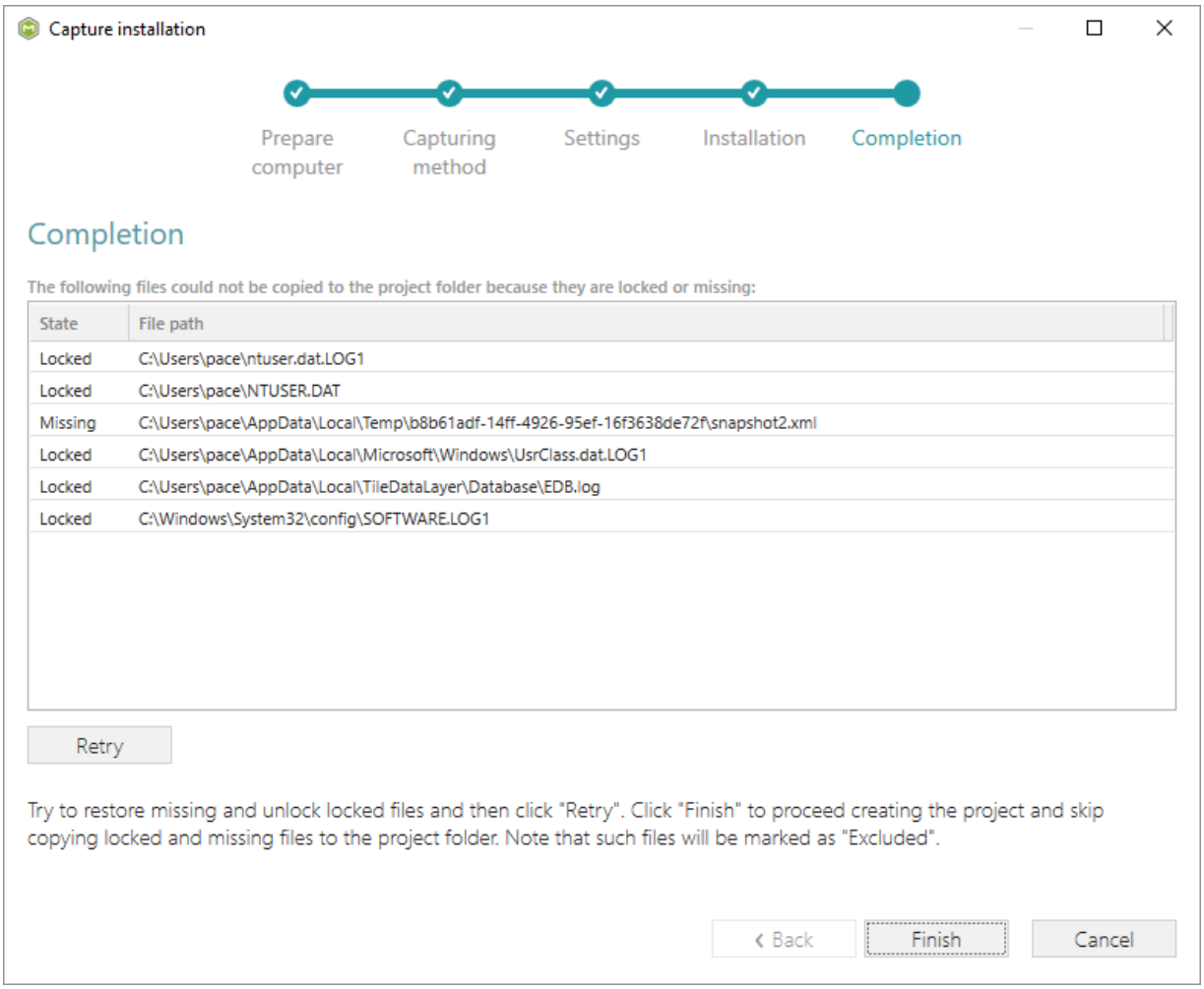


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



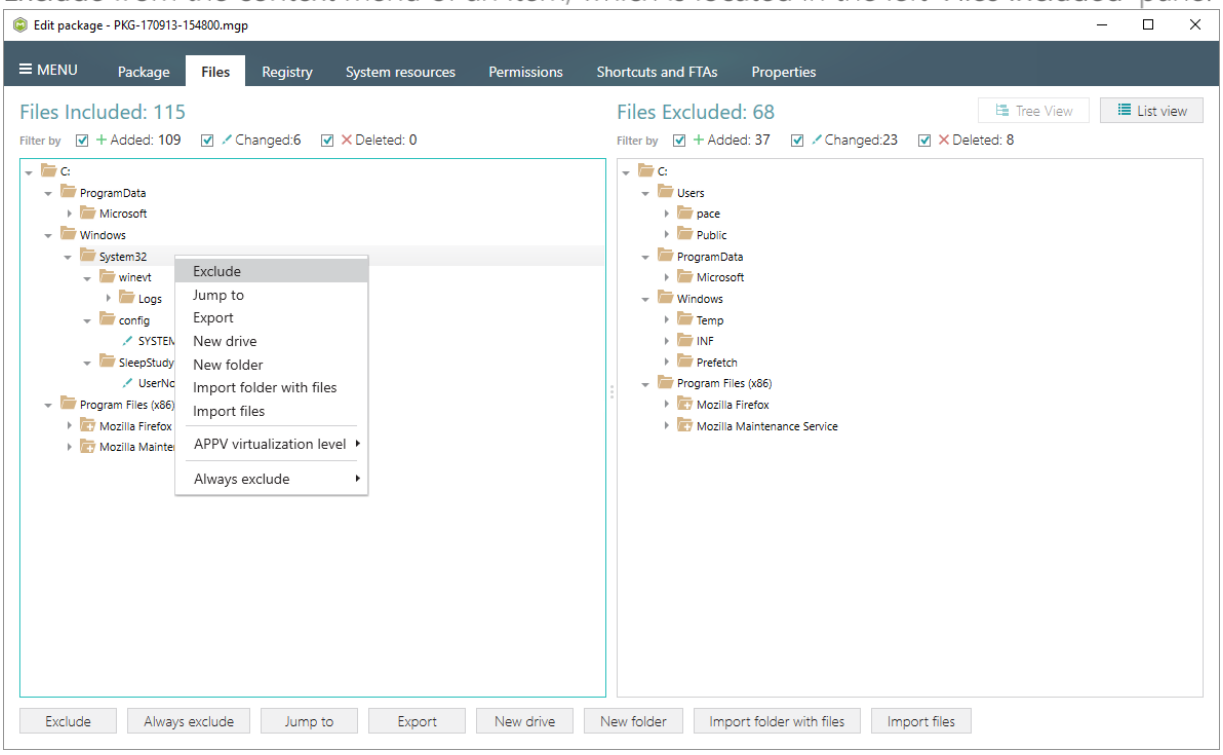

- The following dialog displays captured files, which could not be copied to the project folder because they do not exist anymore or locked by the system or by an application. Try to resolve these issues and then click Retry. Click Finish to skip copying the missing and locked

files and mark them as Excluded resources.

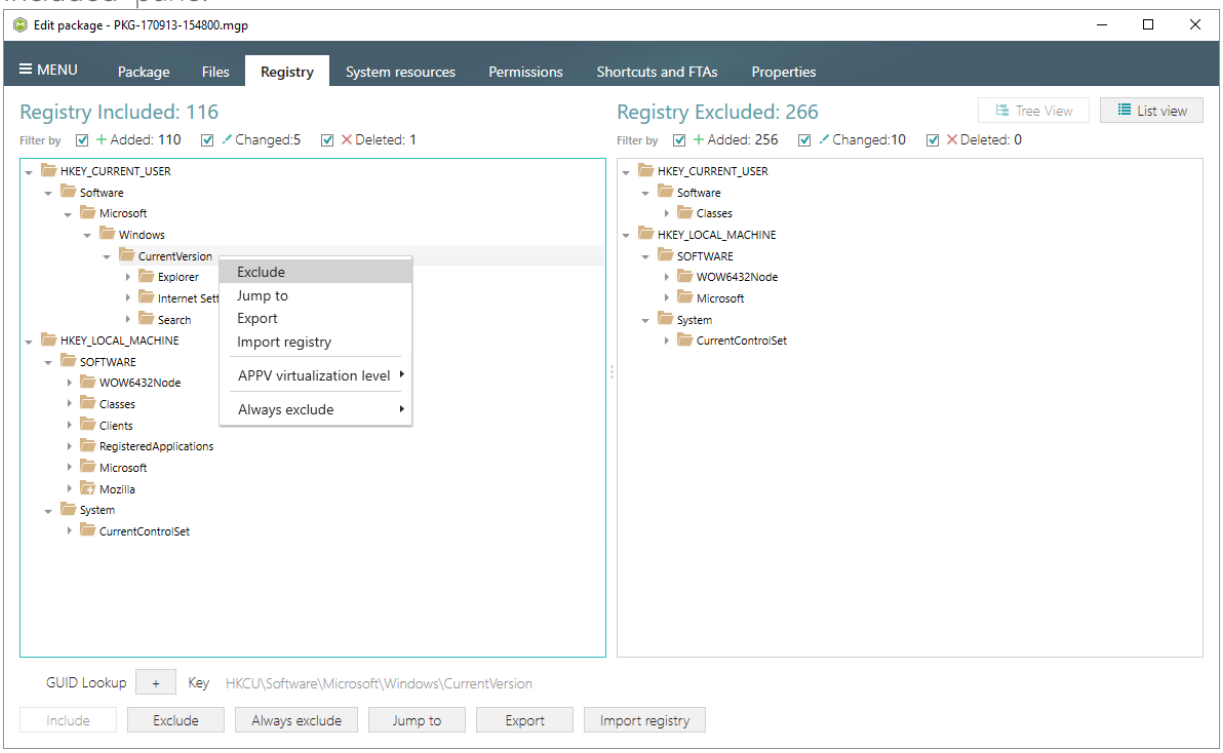



- Once the project is opened in the project editor, review the captured resources at the Files, Registry, System resources, Permissions, and Shortcuts and FTAs tabs and exclude unnecessary ones from the project. Unnecessary resources are files, registry entries, which are usually created or modified in the result of operating system work, and such resources could not be a part of your captured application. Unfortunately, there is no universal rule to discover which of captured files or registry entries should be excluded, so exclude only those ones, which almost 100% do not refer to your captured application (e.g. NOD32 antivirus files couldn't be a part of Firefox application).

- 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.

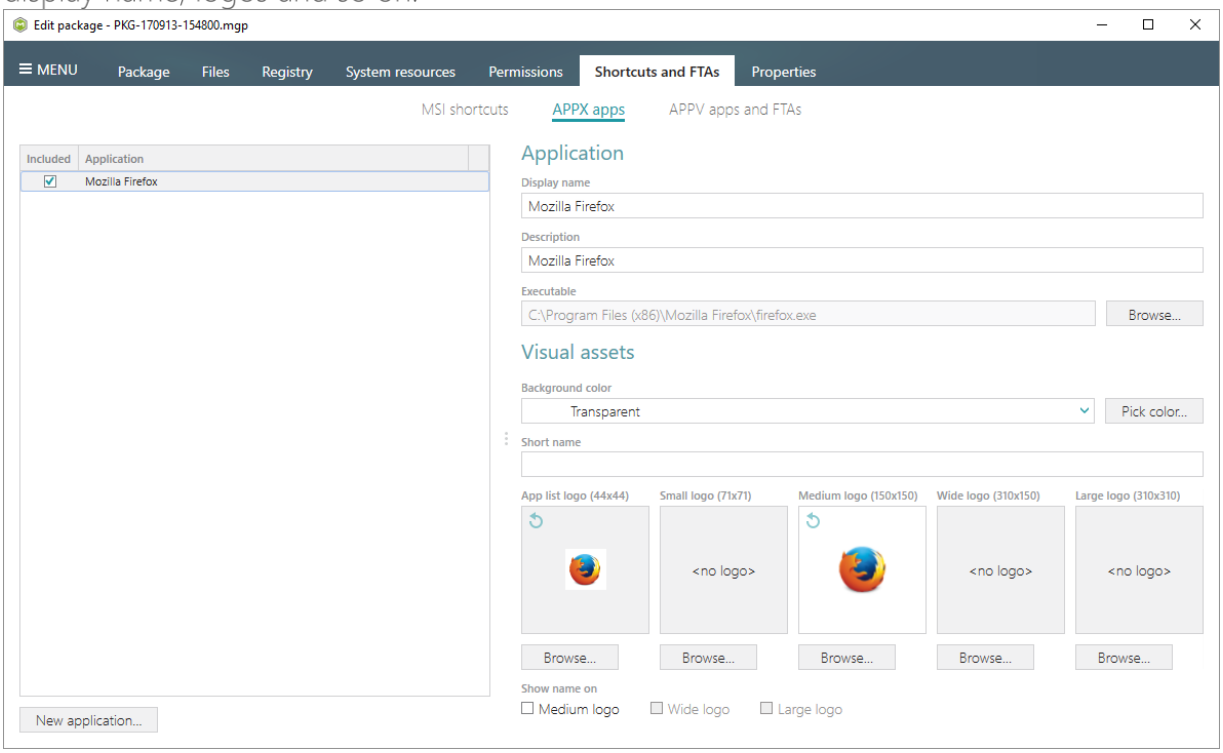



- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.

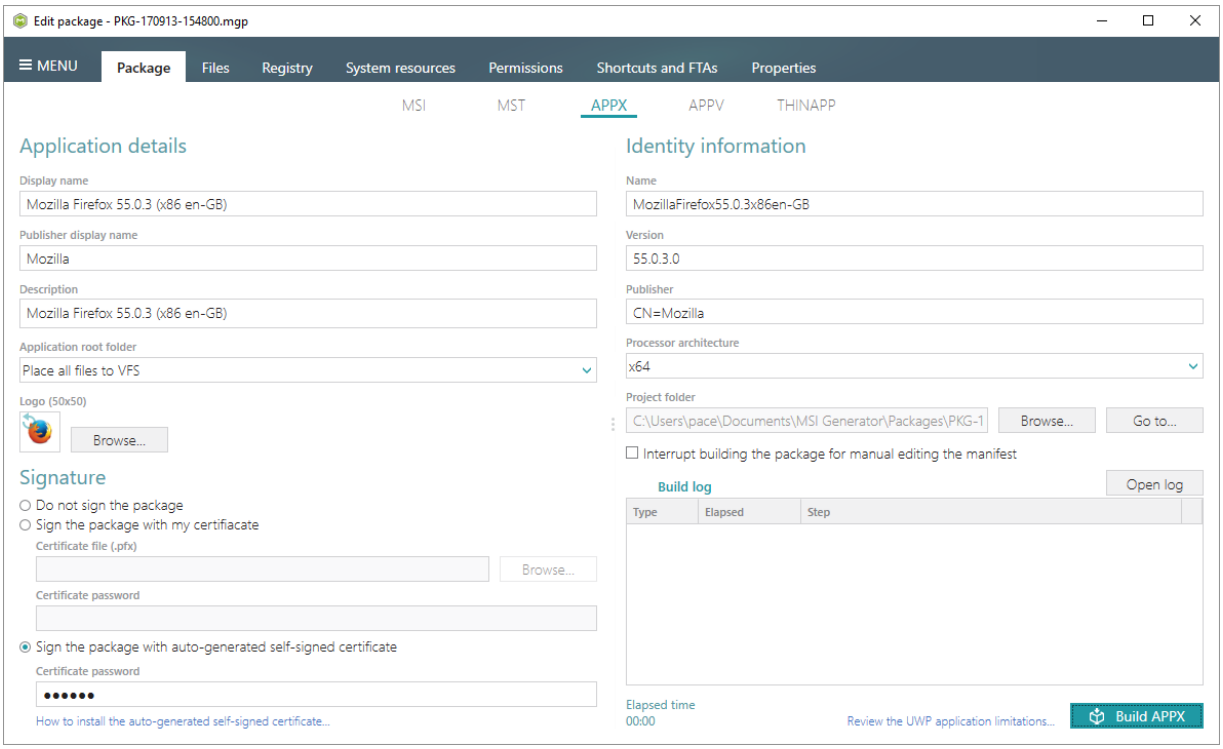



- In order to review and exlude unnecessary APPX apps (shortcuts), go to the **Shortcuts and FTAs -> APPX apps** tab and uncheck the checkbox, located before the app name in the list, for those apps, which you want to exclude. Using the details pane, you can update the app

display name, logos and so on.




- Finaly, to build APPX package from the project, navigate to the **Package** -> **APPX** tab, review and update (if necessary) **Application details**, **Indentity information**, choose the digital signing option and click **Build APPX**. Find description of the APPX settings below in the table.




| Application details | |
|---|---|
| **Display name** | A friendly name of the package that can be displayed to users. |

| | |
|---|---|
| Publisher display name | A friendly name for the publisher of the package that can be displayed to users. |
| Description | A friendly description that can be displayed to users. |
| Application root folder | A location of the main application folder. Content of the selected folder will be placed to the package root, instead of VFS. |
| Logo (50x50) | A logo image of the package that can be displayed to users. |

### Signature

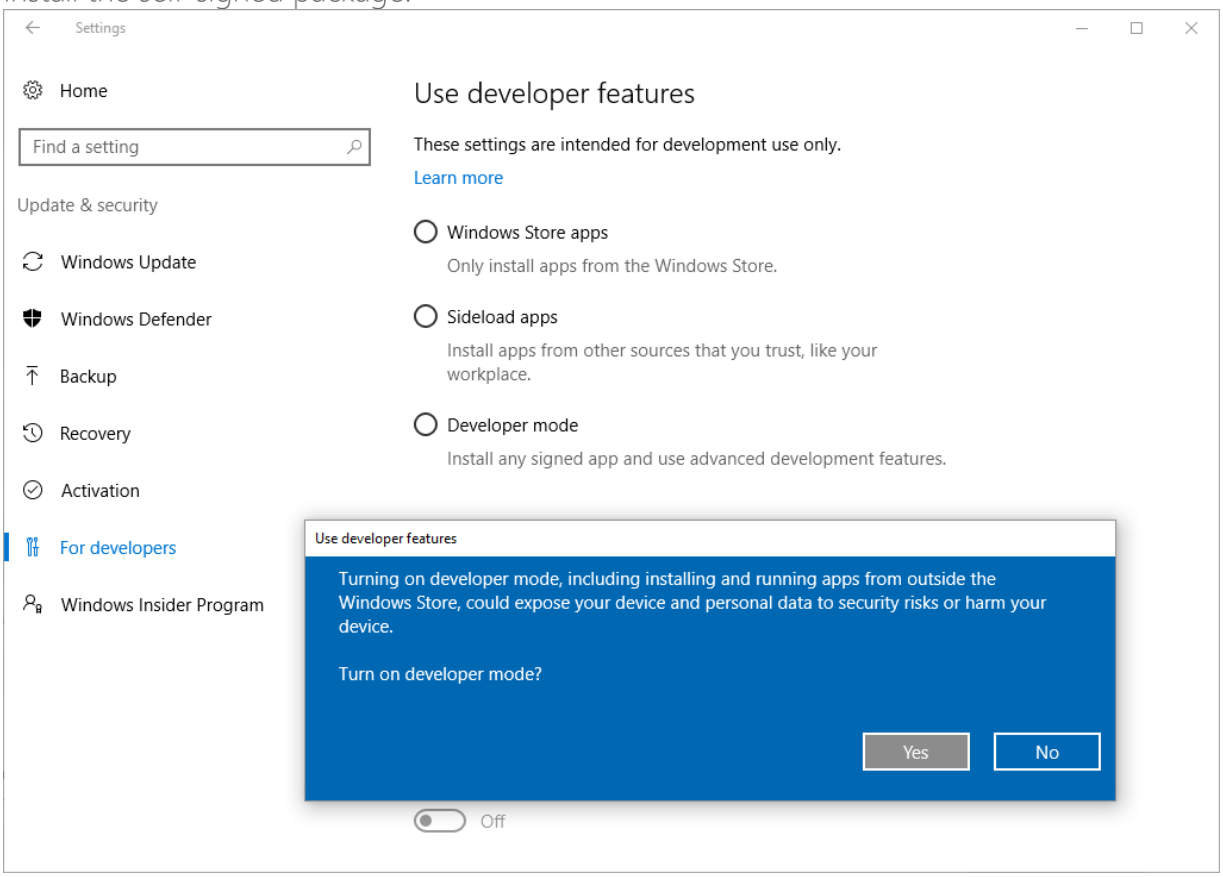
| | |
|---|---|
| Do not sign the package | Use this option if you are planning to submit the package to Windows Store or sign it manully. |
| Sign the package with my certificate | Use this option to sign the generated package with the valid certificate that you already have. |
| Sign the package with auto-generated self-signed certificate | Use this option to generate the self-signed certificate and sign the package with this certificate.<br>NOTE In order to deploy your package, install the generated certificate before (double-click the **auto-generated.pfx** file, and in the Certificate Import Wizard, install the certificate onto the **Local Machine**, and place the certificate into the **Trusted People** certificate store) and enable the **Developer mode** in Windows settings, as described in the next step. |

### Identity information

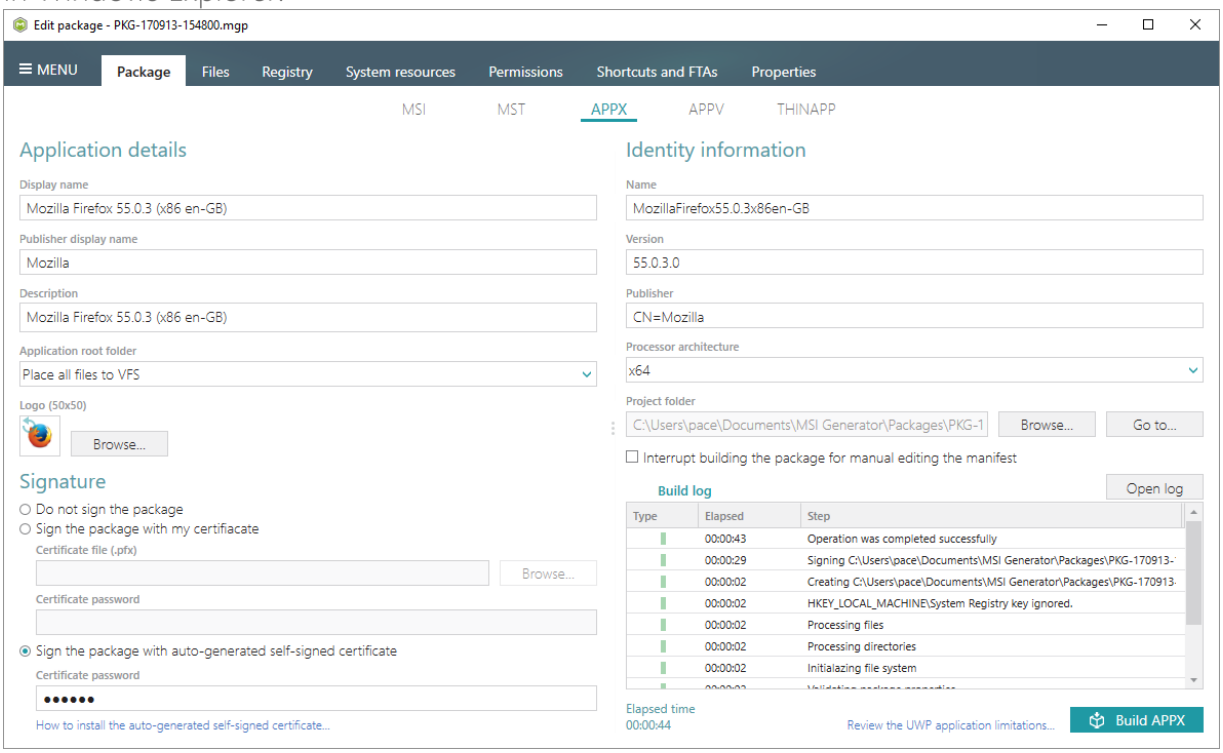
| | |
|---|---|
| Name | Describes the contents of the package. |
| Version | The version number of the package. |
| Publisher | Describes the publisher information. The Publisher attribute must match the publisher subject information of the certificate used to sign a package. |
| Processor architecture | Describes the architecture of the code contained in the package. |

- If you have built the package with the **Sign the package with the auto-generated self-signed certificate** option selected, do not forget to install that generated certificate to the **Trusted People** certificate store and enable the **Developer mode** in the Windows settings to

install the self-signed package.




- Click **Go to...**, located next to the **Project folder** field, to open the package containing folder in Windows Explorer.



## 3.5 Create APPV

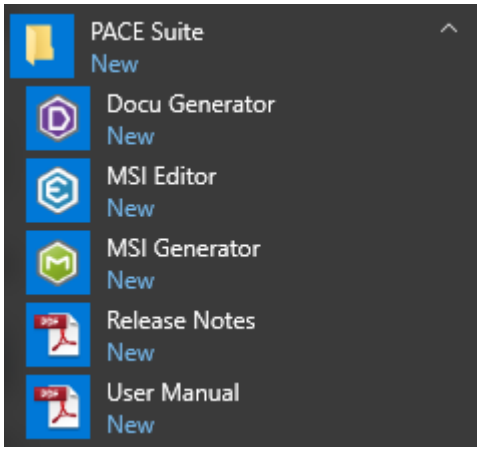
Choose a scenario that better suits your needs:

- 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.
- Convert (Repackage) EXE, MSI to APPV 5.x, 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.

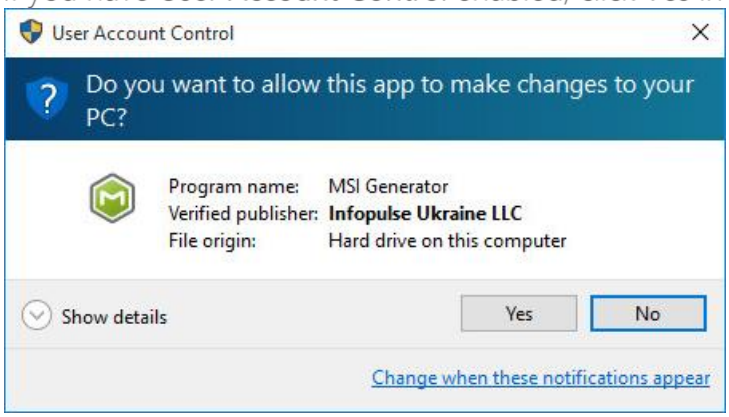
### 3.5.1 Create APPV from Scratch

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.

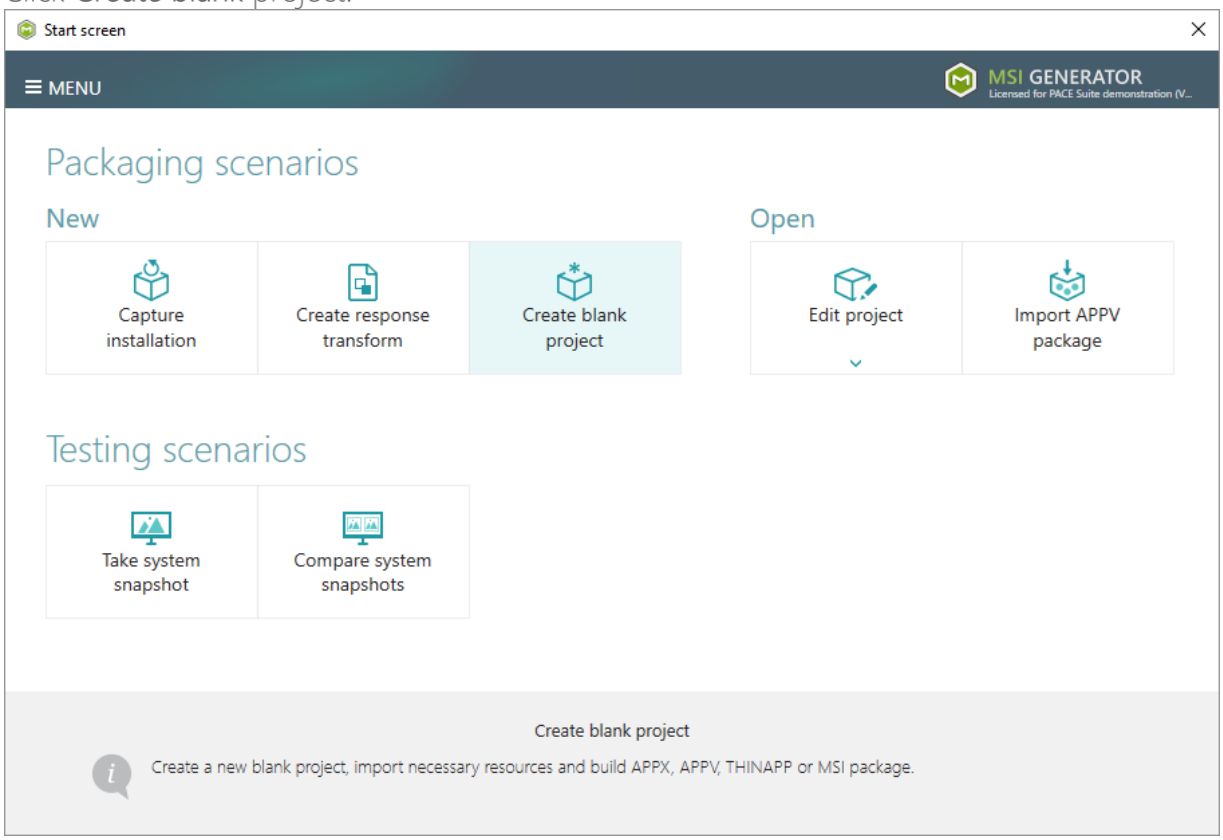
[1]. Launch MSI Generator from the desktop or the start menu shortcut.



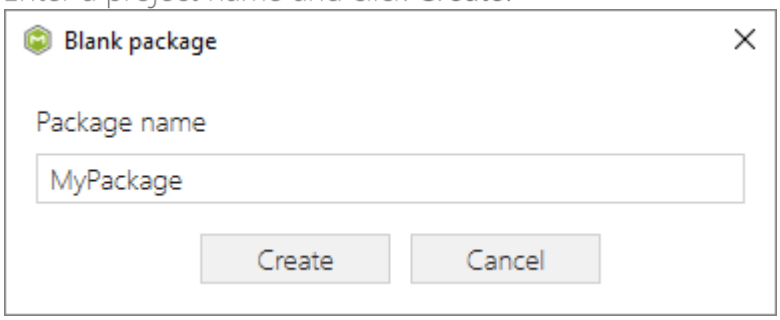

[2]. If you have User Account Control enabled, click Yes in the opened window.



[3]. Click Create blank project.

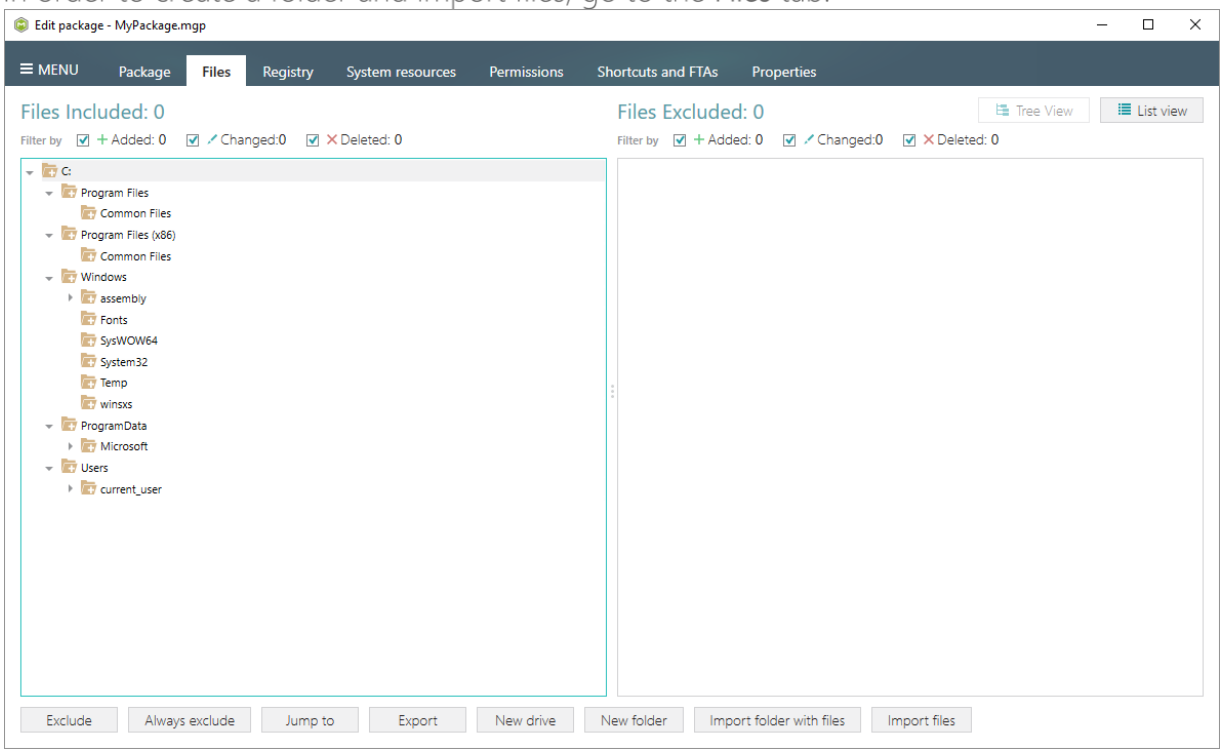



[4]. Enter a project name and click Create.

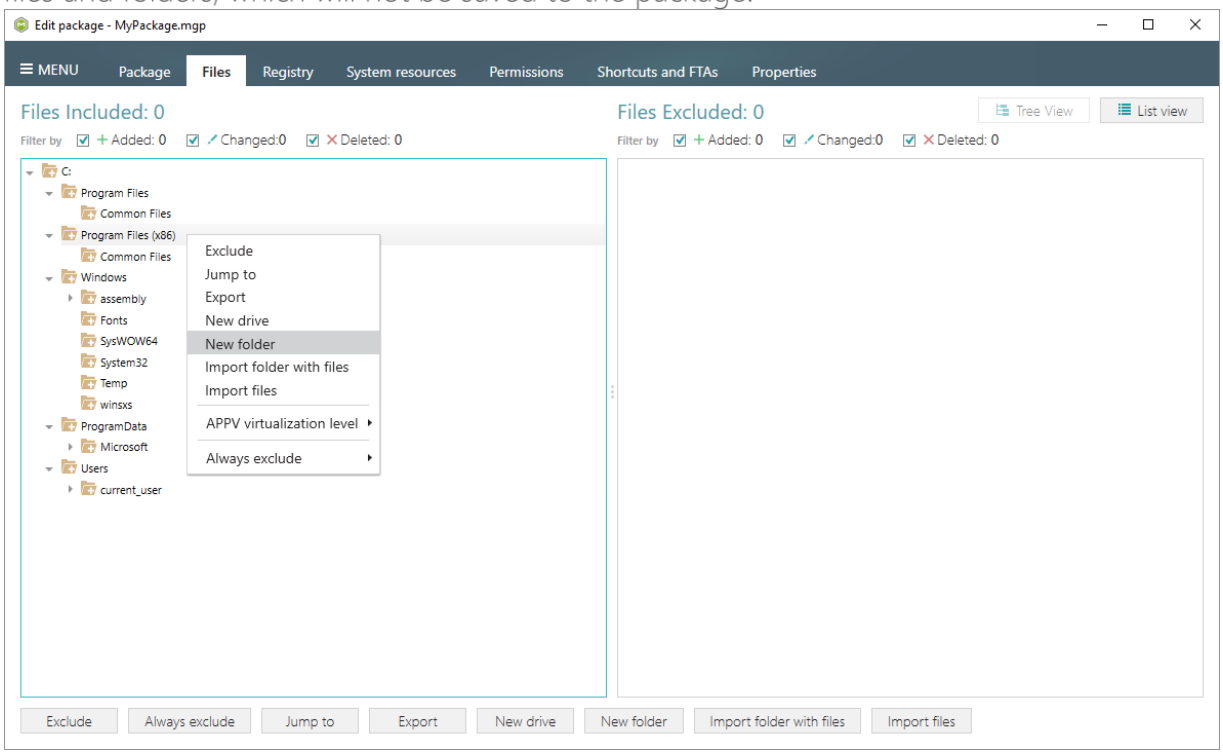



[5]. Once the newly created project is opened automatically, you can start adding the necessary resources using MSI Generator. The steps from [6] to [11] describe how to create a new folder, import files to this folder and change the default APPV virtualization level of a folder; the steps from [12] to [15] describe how to import registry entries from the REG file and change the default APPV virtualization level of a registry key; the steps from [16] to [21] describe how create a new APPV application, its shortcuts and associate file extensions with the application; and two last steps [22] and [23] describe how to build APPV package from the prepared project.

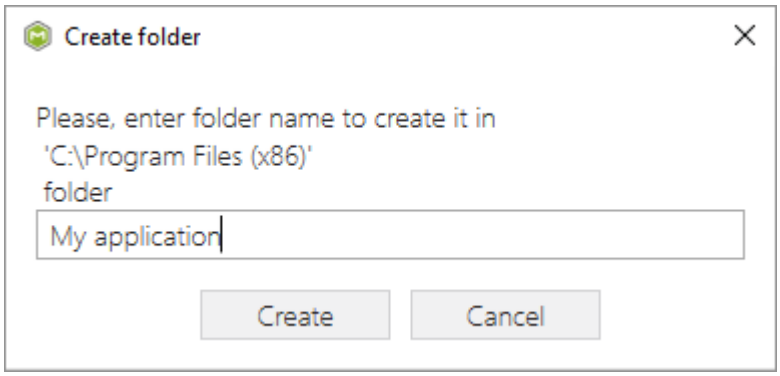
[6]. In order to create a folder and import files, go to the **Files** tab.



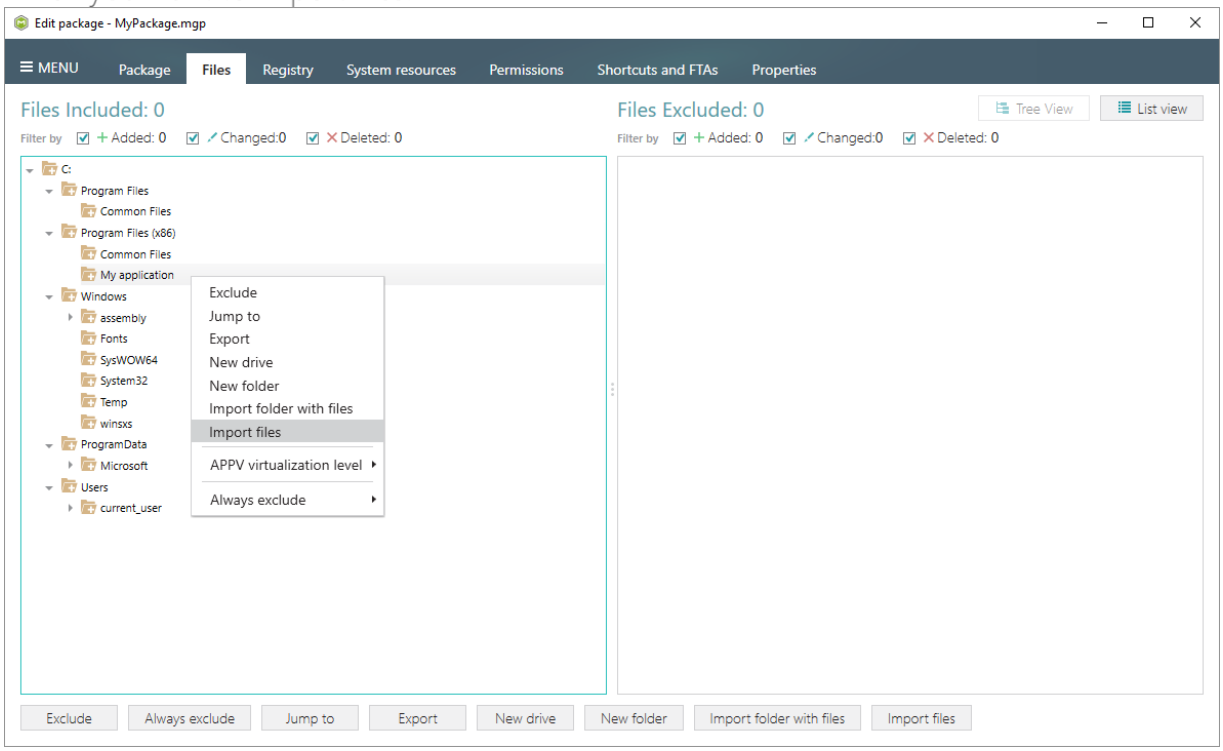

[7]. In the left pane, select **New folder** from the context menu of a folder, where you want to create a new empty folder. Note that the left '**Files Included**' pane displays files and folders, which will be a part of your package, and the right '**Files Excluded**' pane displays files and folders, which will not be saved to the package.



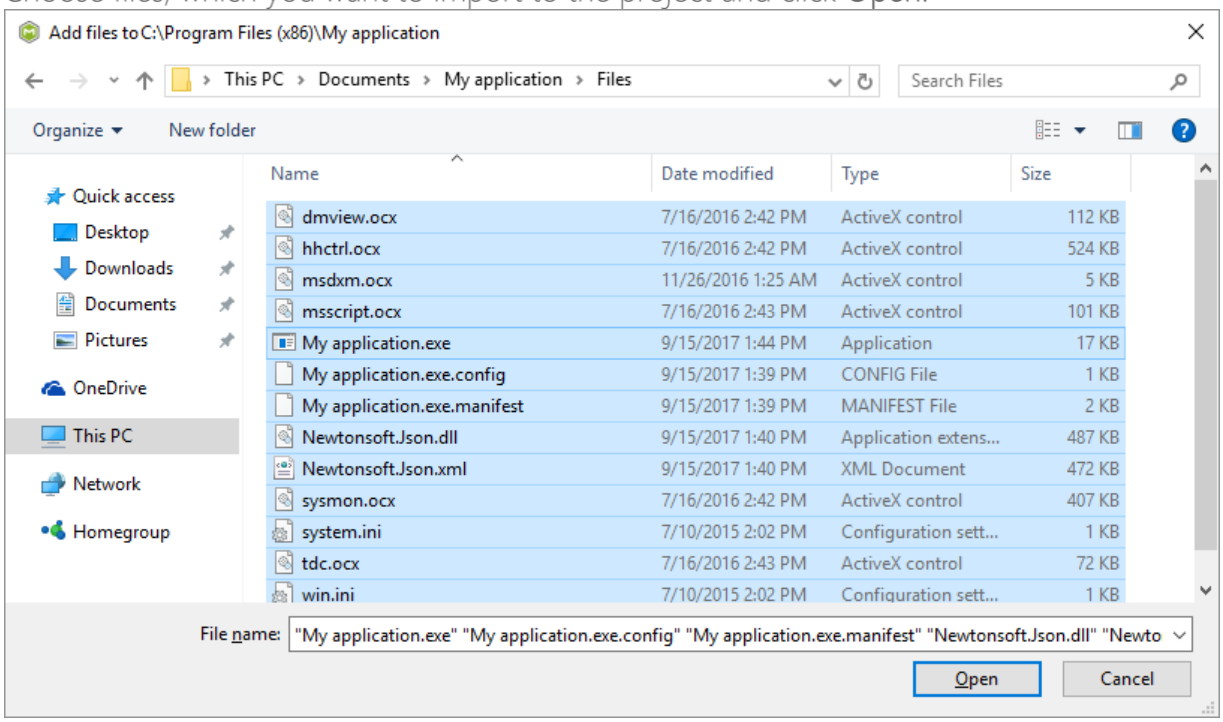
[8]. Enter a folder name and click **Create**.



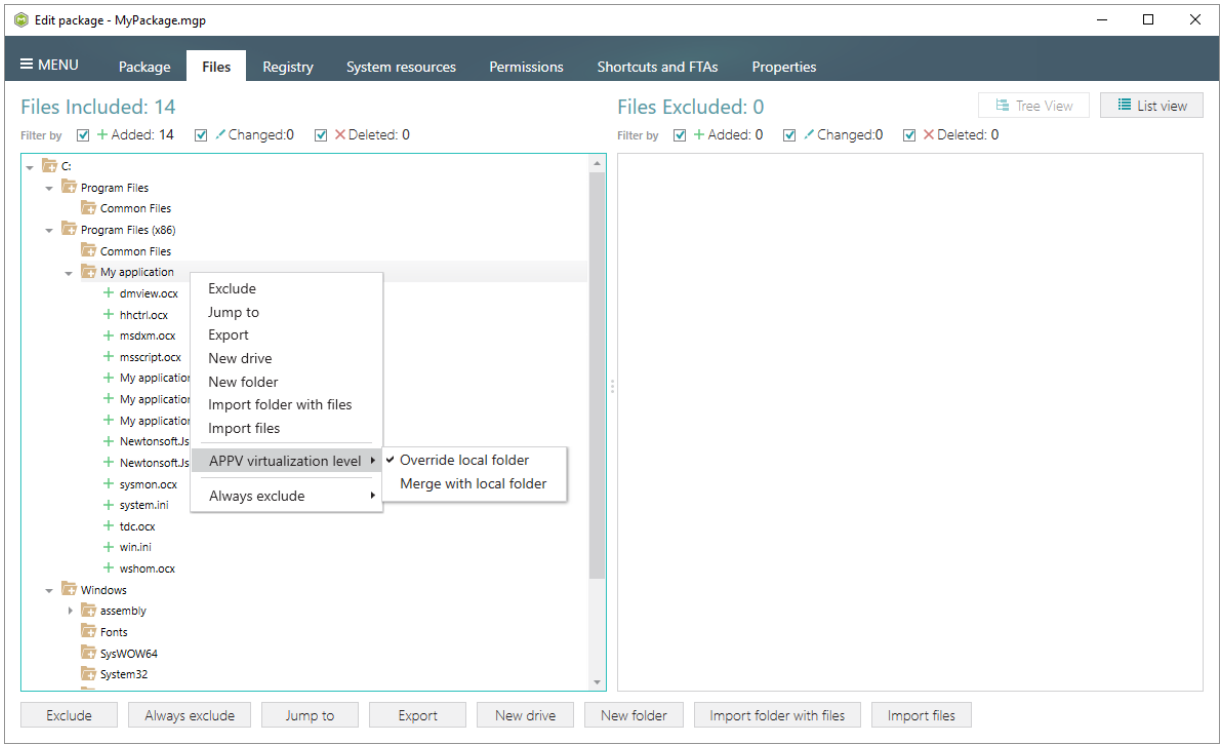

[9]. For adding files to your project, select **Import files** from the context menu of a folder, to which you want to import files.



[10]. Choose files, which you want to import to the project and click Open.

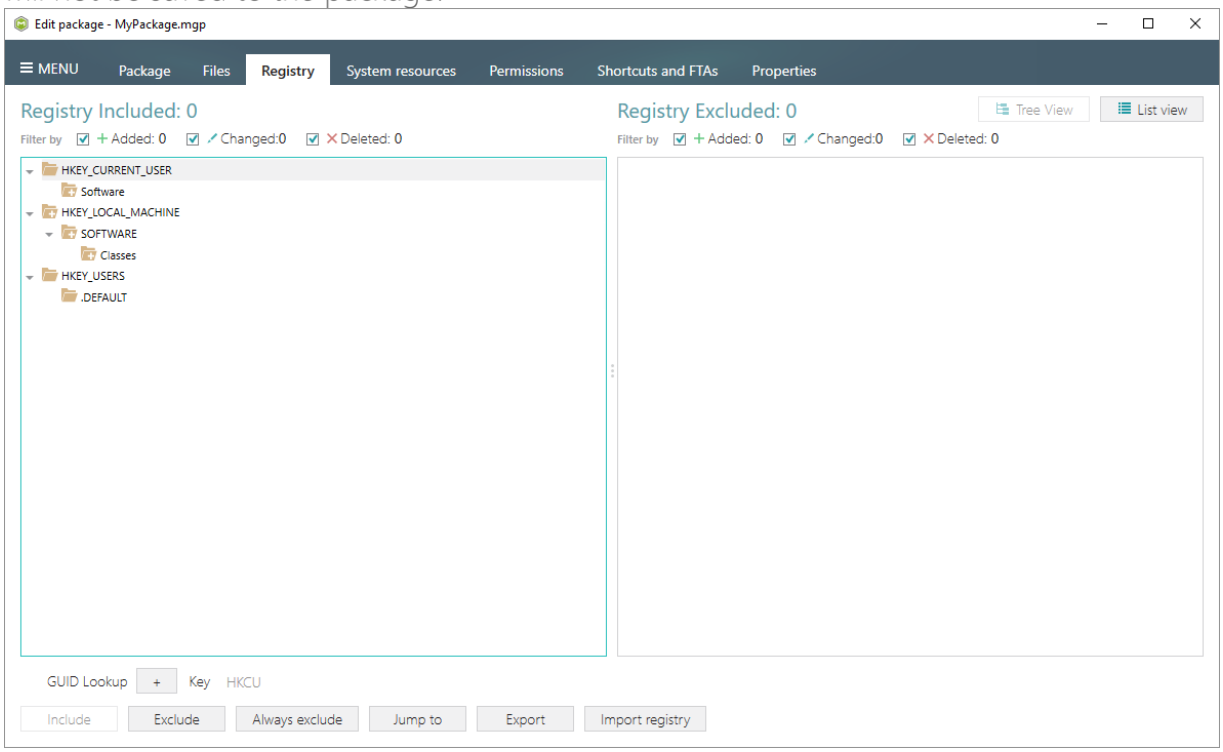



[11]. In order to change the default virtualization (isolation) level of a folder, open the folder context menu, and select either Override local folder or Merge with local folder option from the APPV virtualization level submenu. Note that if the APPV virtualization level submenu is disabled with the Override local folder option selected, it means that this option is inherited from the parent folder. Child folder could not have the Merge with local folder virtualization level, if parent one has the Override local folder virtualization level.

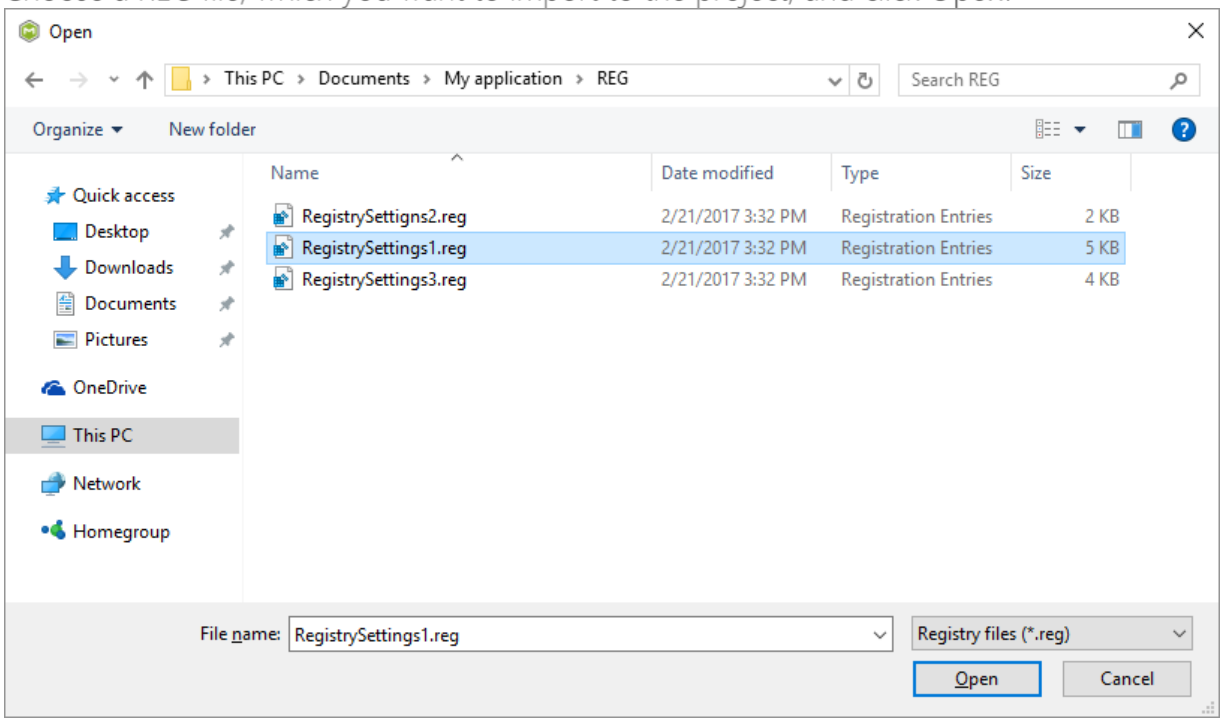



[12]. In order to import registry entries from the REG file, go to the Registry tab and click the Import registry button, located in the bottom part of the window. Note that the left

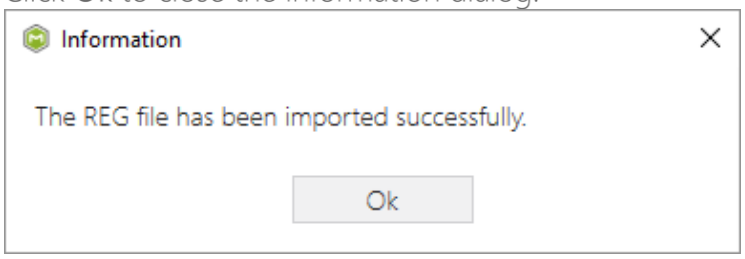
'Registry Included' pane displays registry keys and values, which will be a part of your package, and the right 'Registry Excluded' pane displays registry keys and values, which will not be saved to the package.



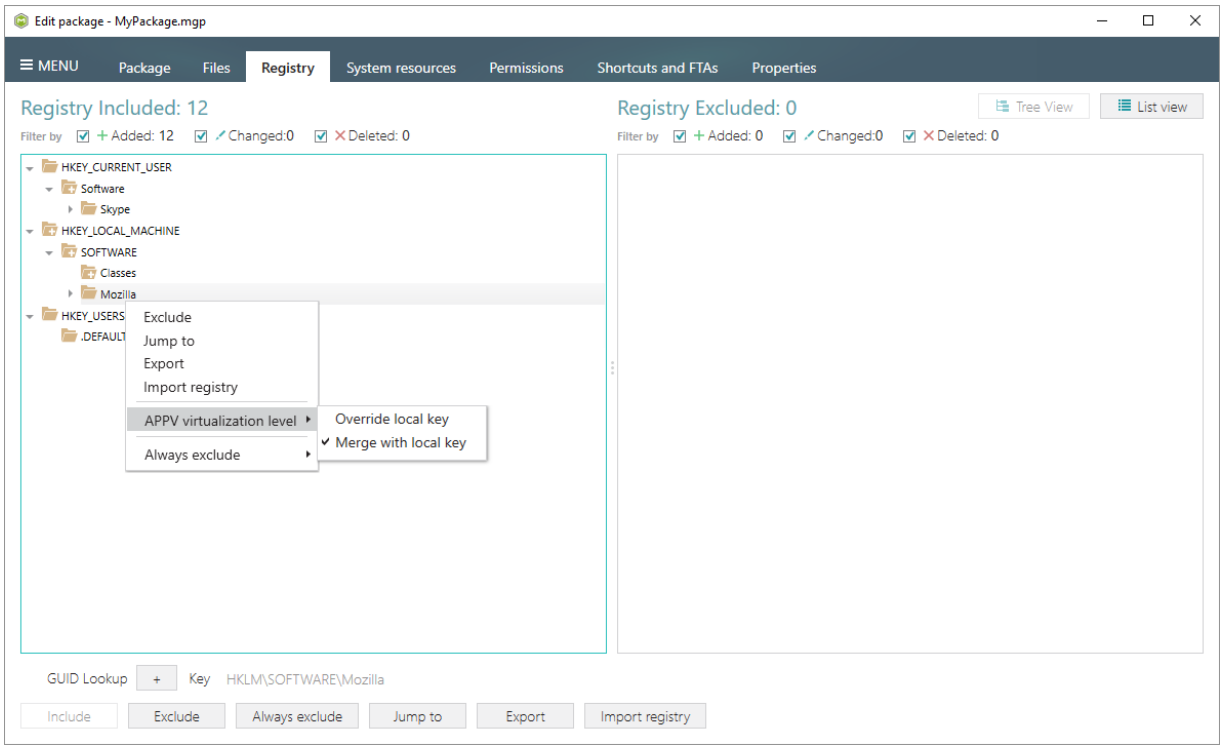

[13]. Choose a REG file, which you want to import to the project, and click Open.



[14]. Click Ok to close the information dialog.

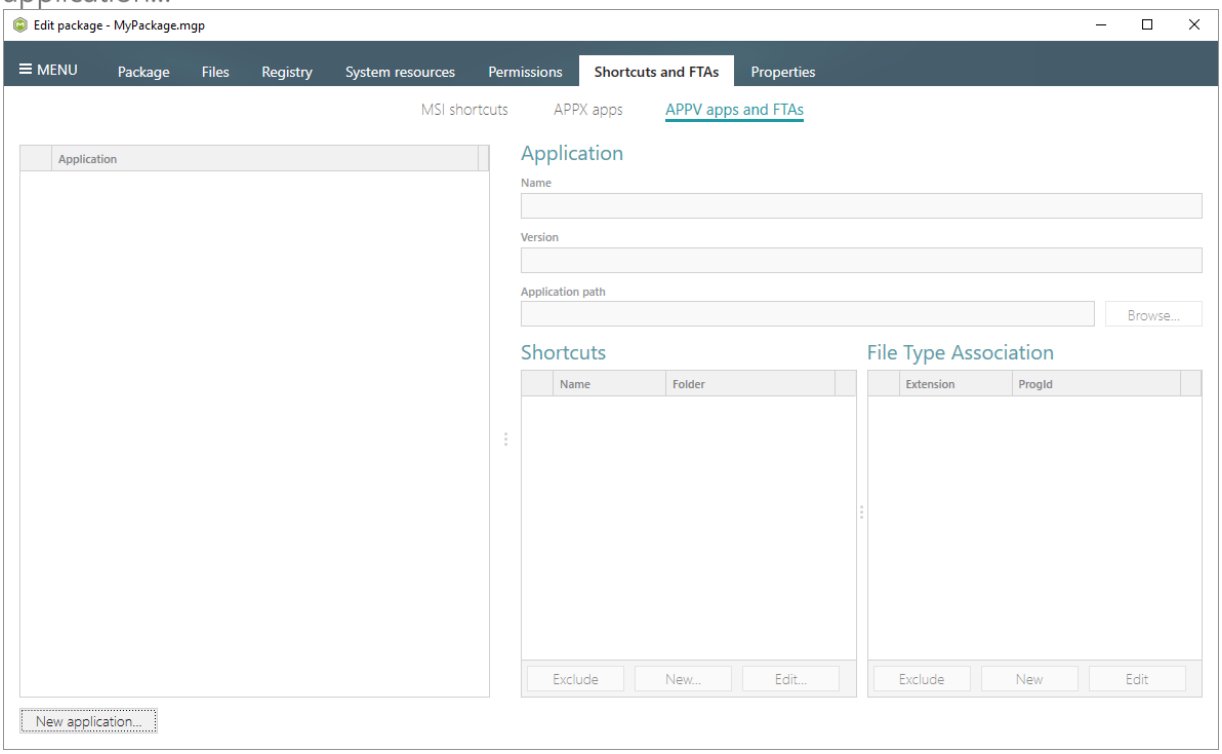



[15]. In order to change the default virtualization (isolation) level of a registry key, open the registry key context menu, and select either Override local key or Merge with local key option from the APPV virtualization level submenu. Note that if the APPV virtualization level submenu is disabled with the Override local key option selected, it means that this option is inherited from the parent registry key. A child registry key could not have the Merge with local key virtualization level, if the parent one has the Override local key virtualization level.

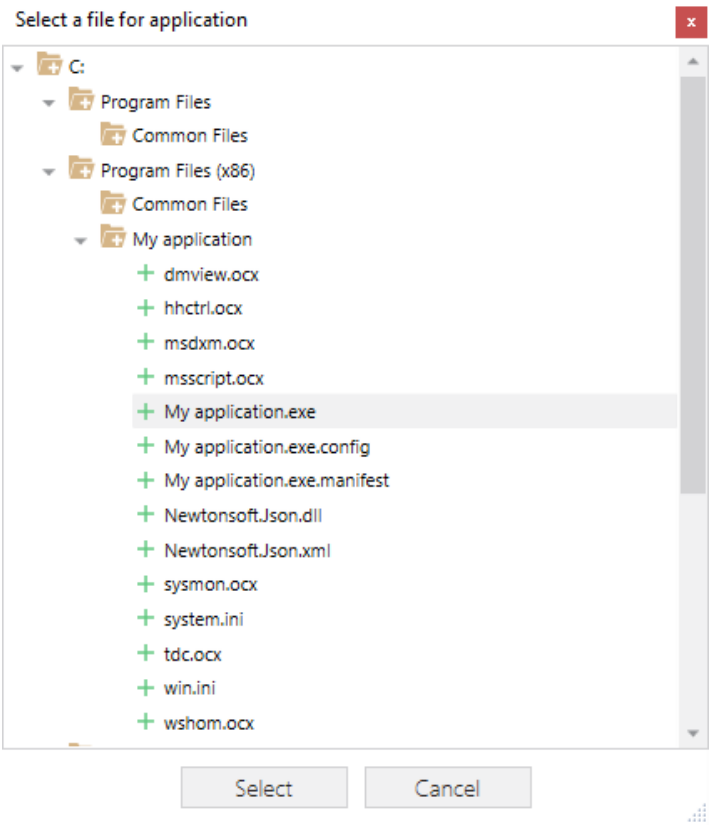



[16]. If you need to create an APPV application, its shortcuts and associate file extensions with this application, go to the Shortcuts and FTAs -> APPV apps and FTAs tab and click New

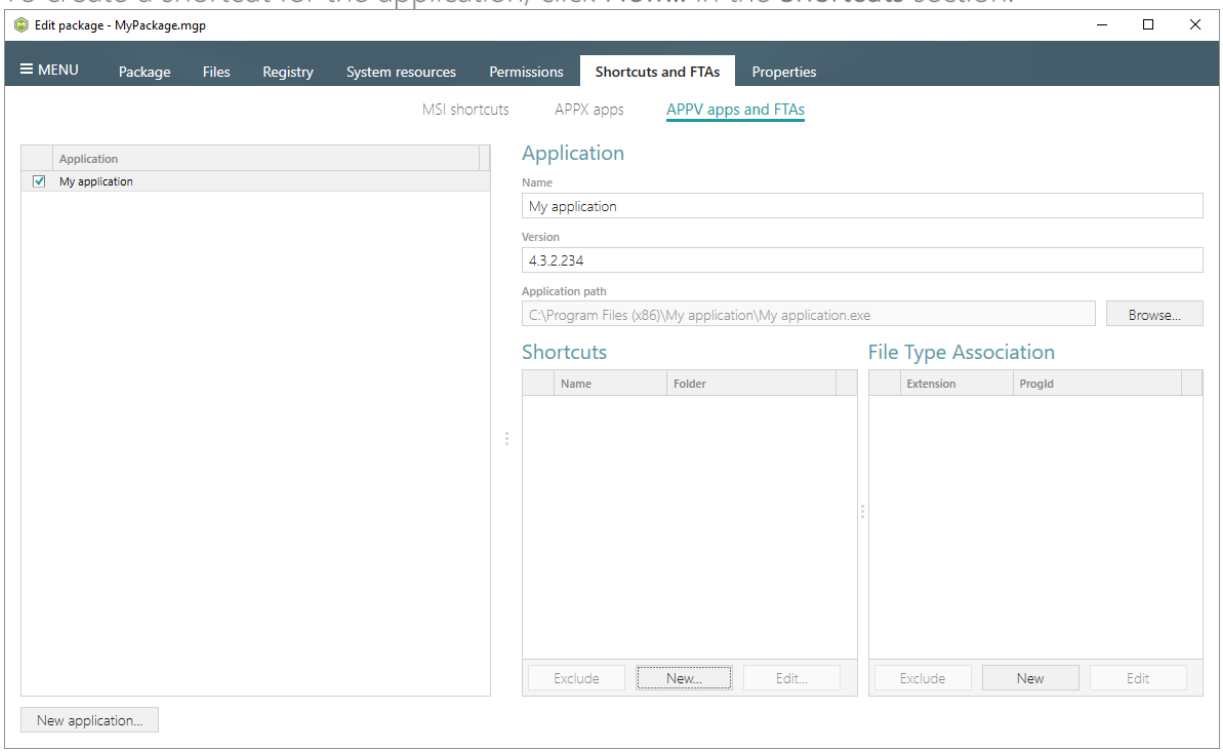
application...



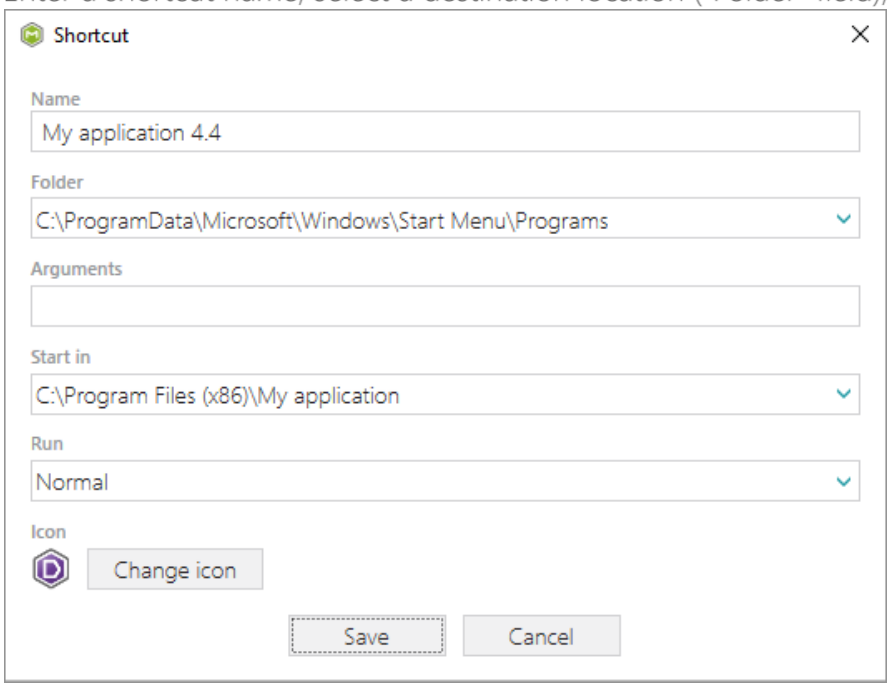

[17]. Select the main executable file (EXE) of your application from the tree and click **Select**.



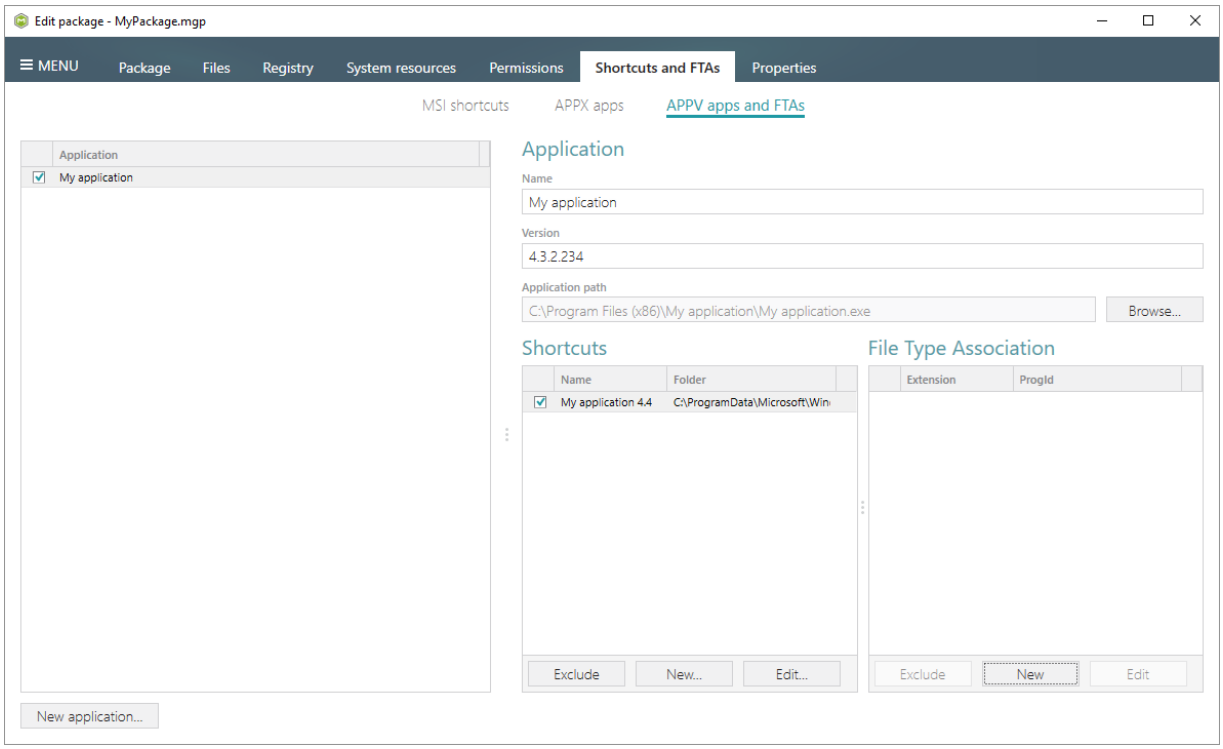
[18]. To create a shortcut for the application, click **New...** in the **Shortcuts** section.



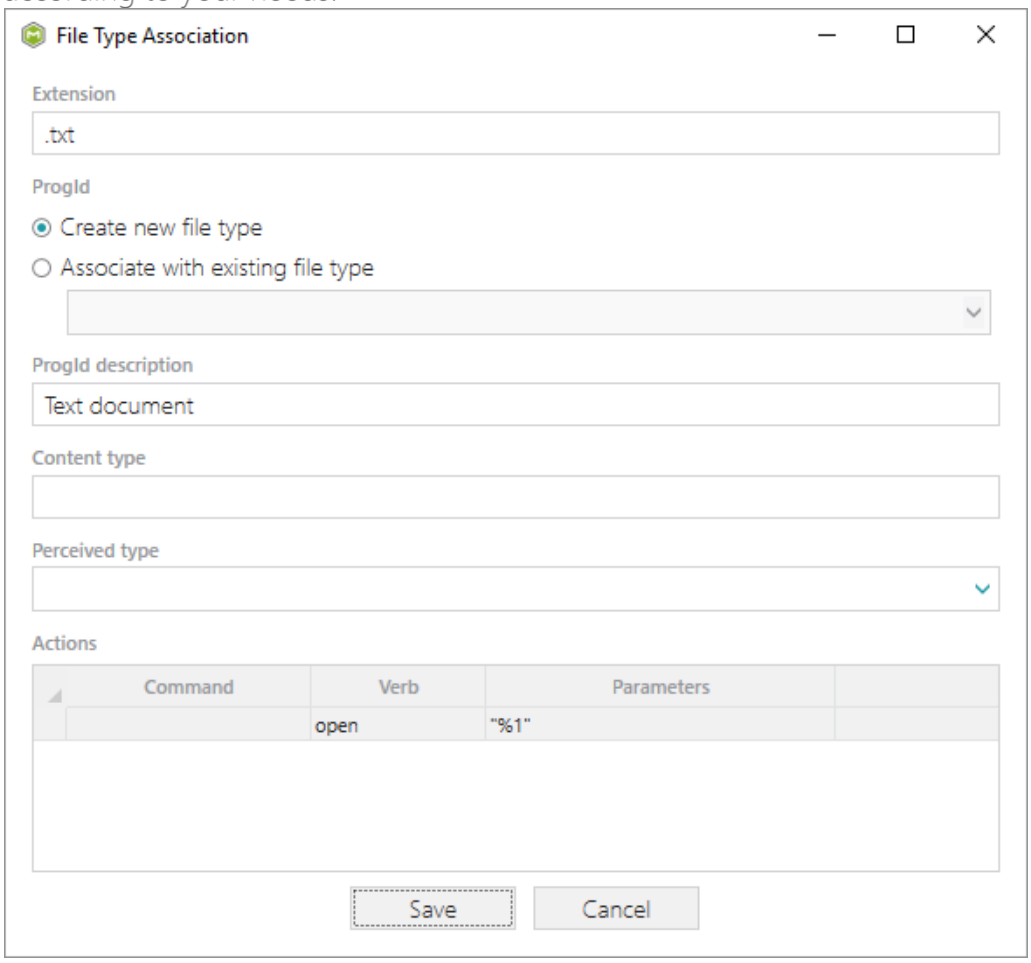

[19]. Enter a shortcut name, select a destination location ("Folder" field), icon and click **Save**.



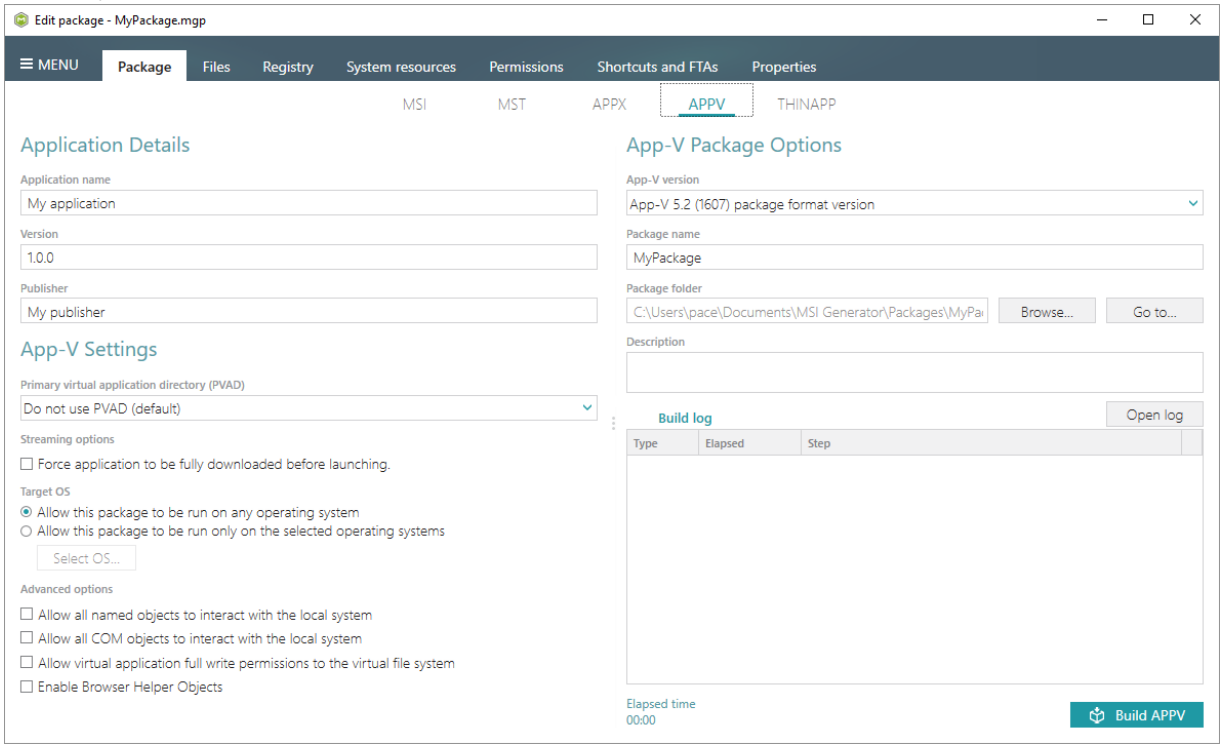
[20]. To associate some file extension with your application, click **New...** in the **File Type Association** section.



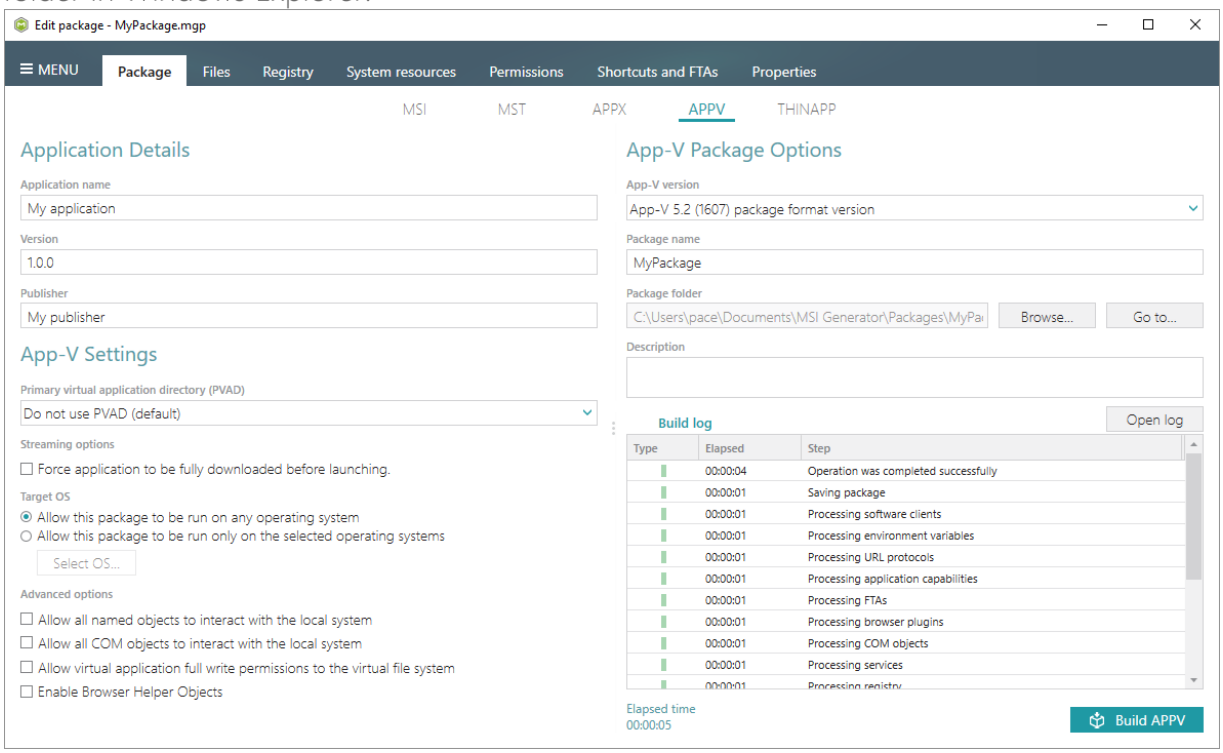

[21]. Enter a file extension and click **Save**. Optinally you can modify other FTA settings according to your needs.



[22]. In order to build APPV package from your project, navigate to the **Package** -> **APPV** tab, update **Application Details**, select the necessary **App-V Settings**, App-V package format version, and then click **Build APPV**.



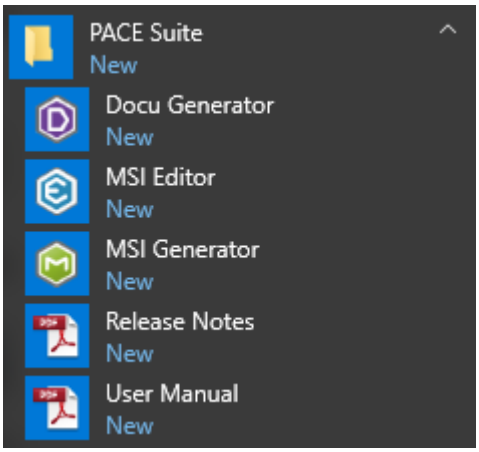

[23]. Click **Go to...**, located next to the **Project folder** field, to open the package containing folder in Windows Explorer.



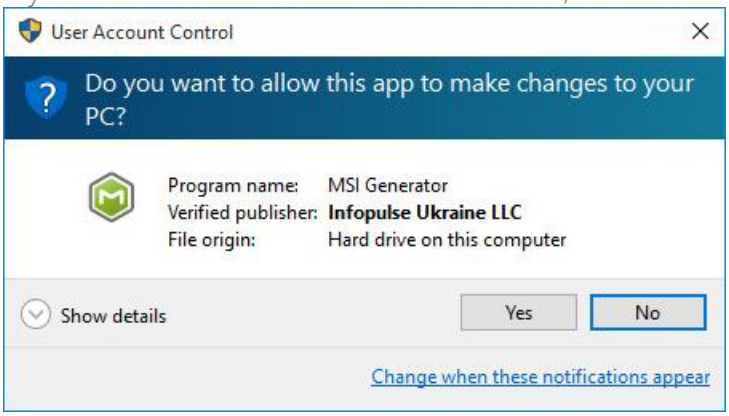
### 3.5.2 Convert (Repackage) EXE, MSI to APPV 5.x

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

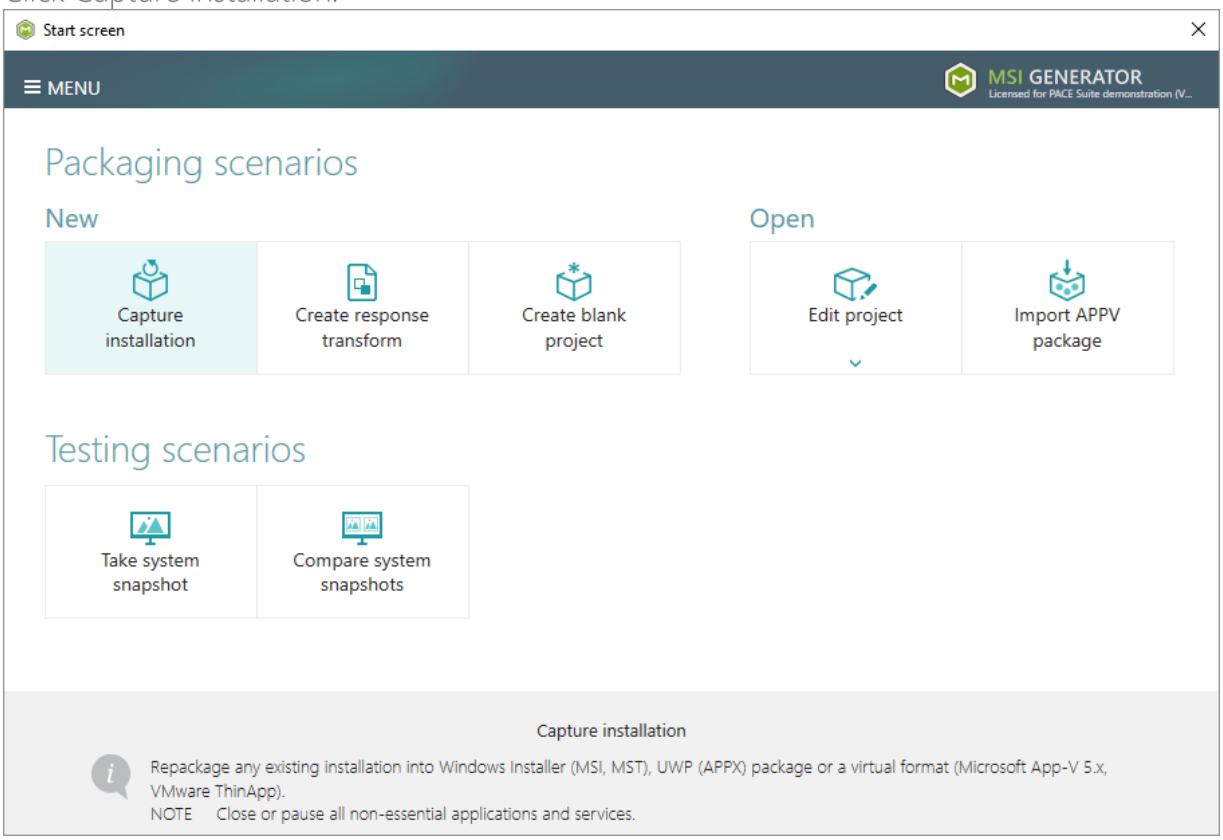
- Launch MSI Generator from the desktop or the start menu shortcut.



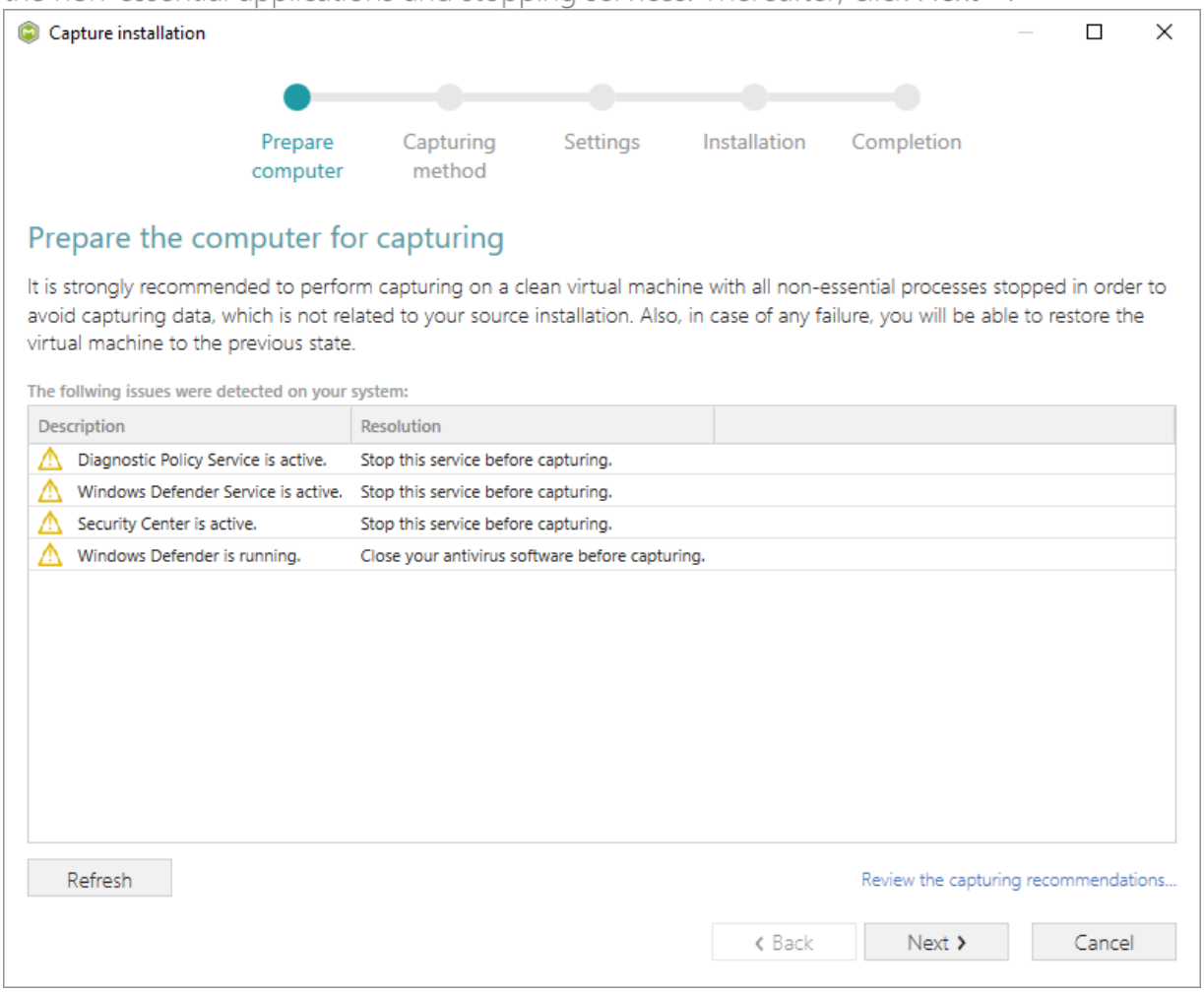

- If you have User Account Control enabled, click Yes in the opened window.



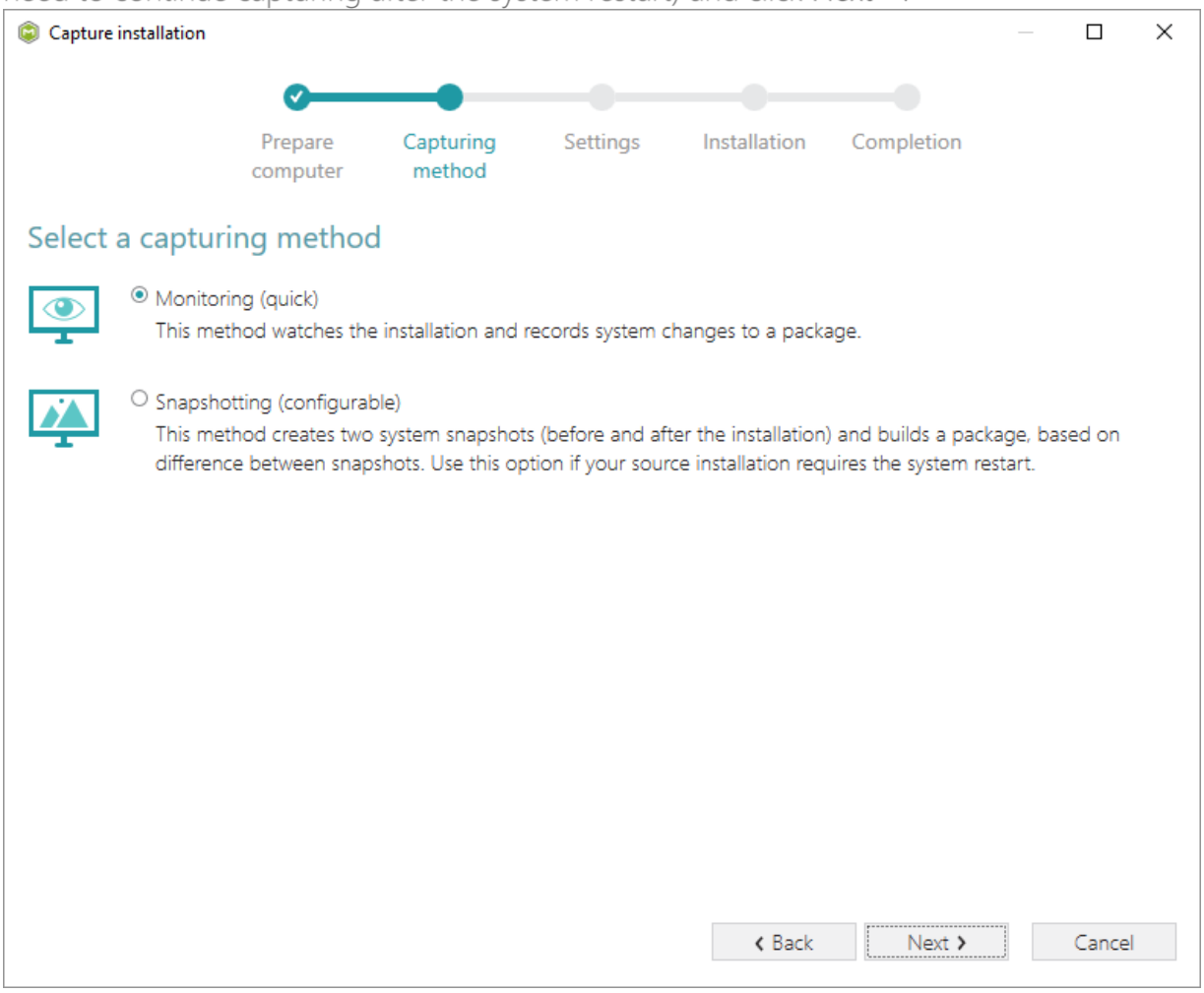
- Click Capture installation.



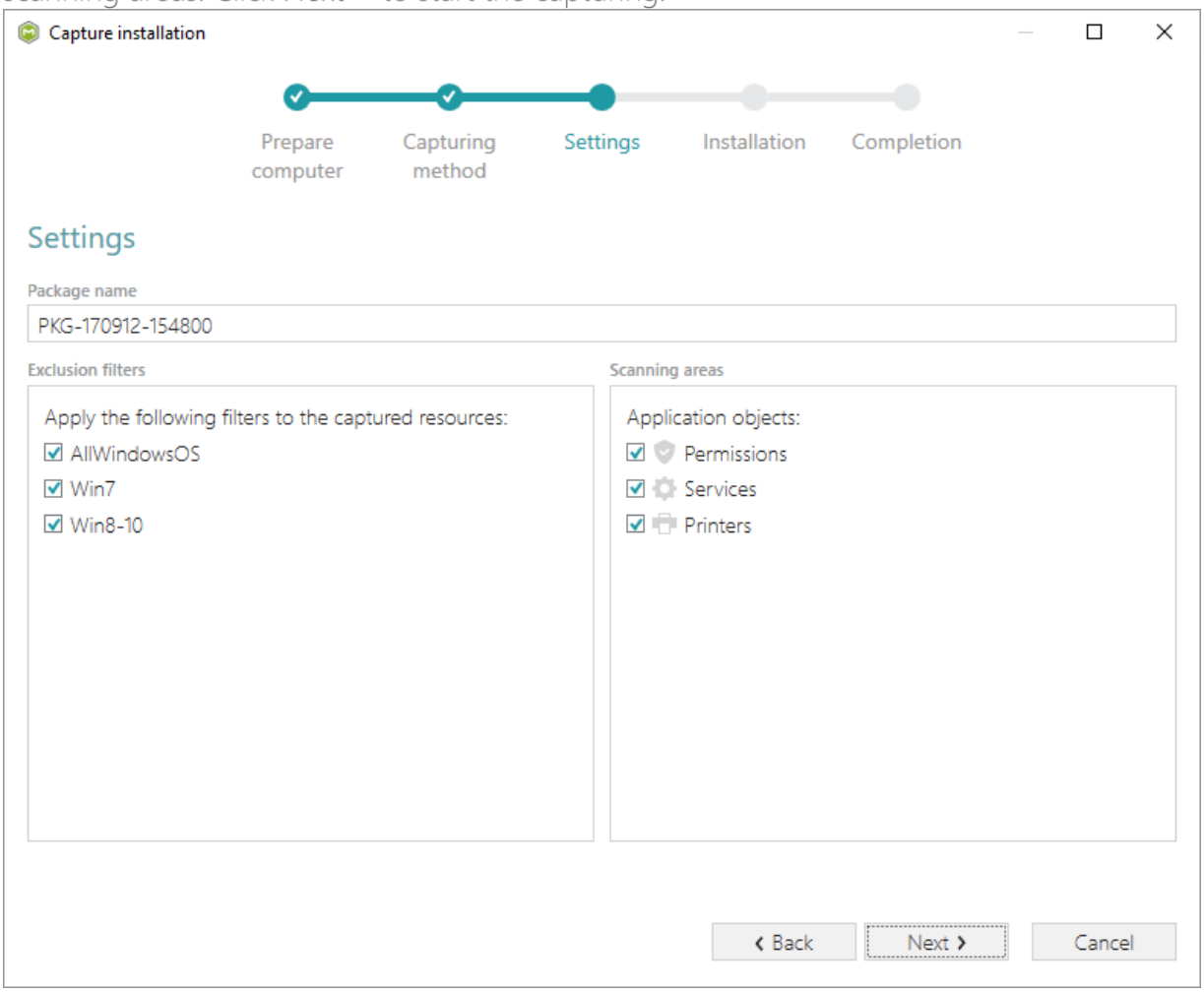
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



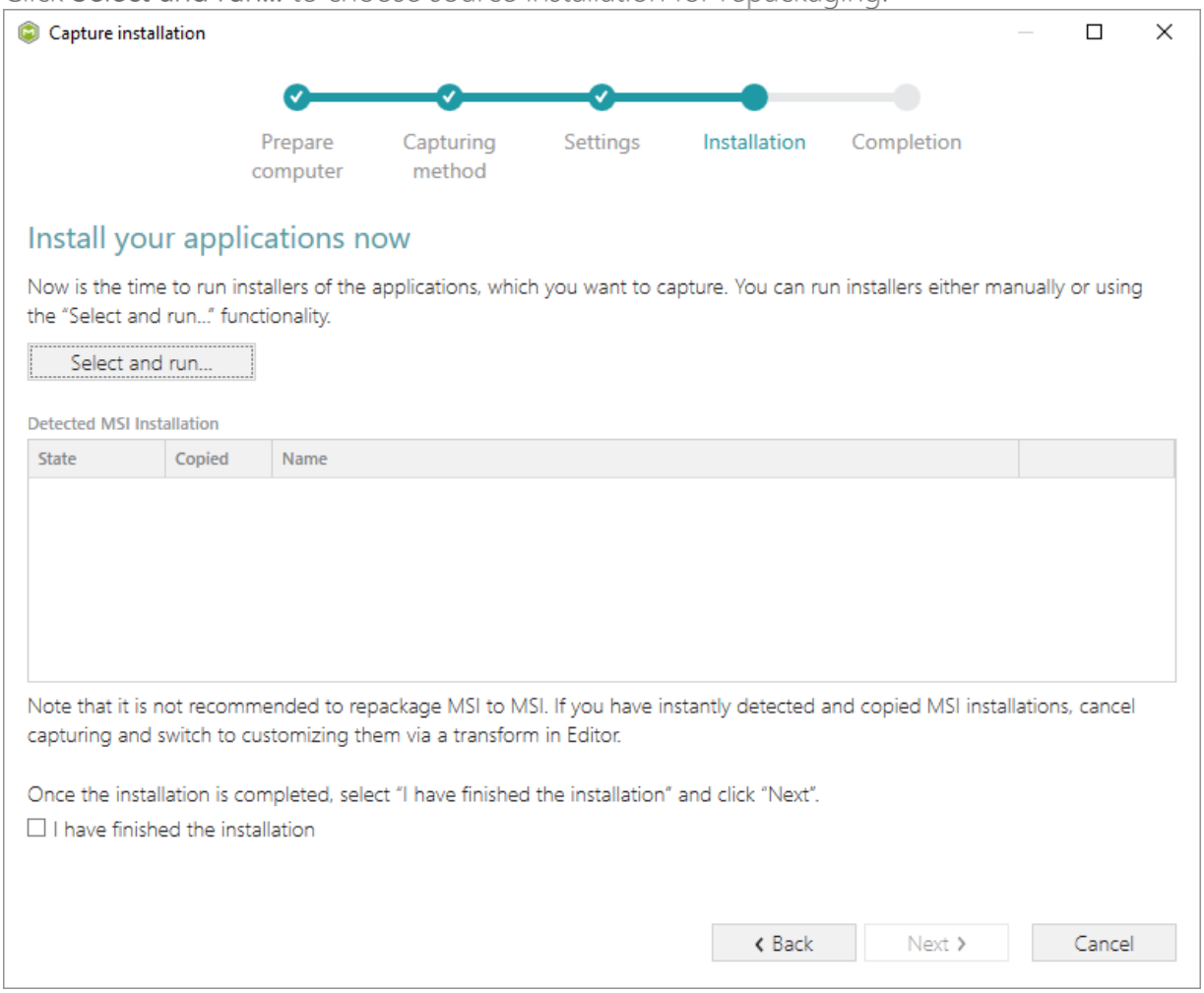
- Select the Monitoring method for the quicker capturing (or use the Snapshotting one if you need to continue capturing after the system restart) and click Next >.



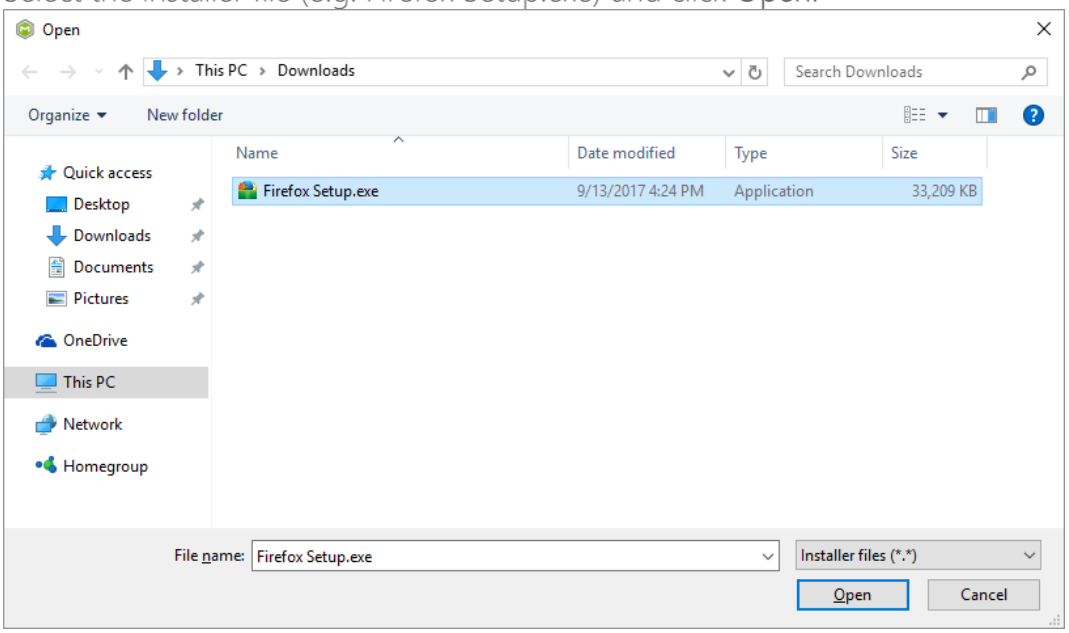
- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.



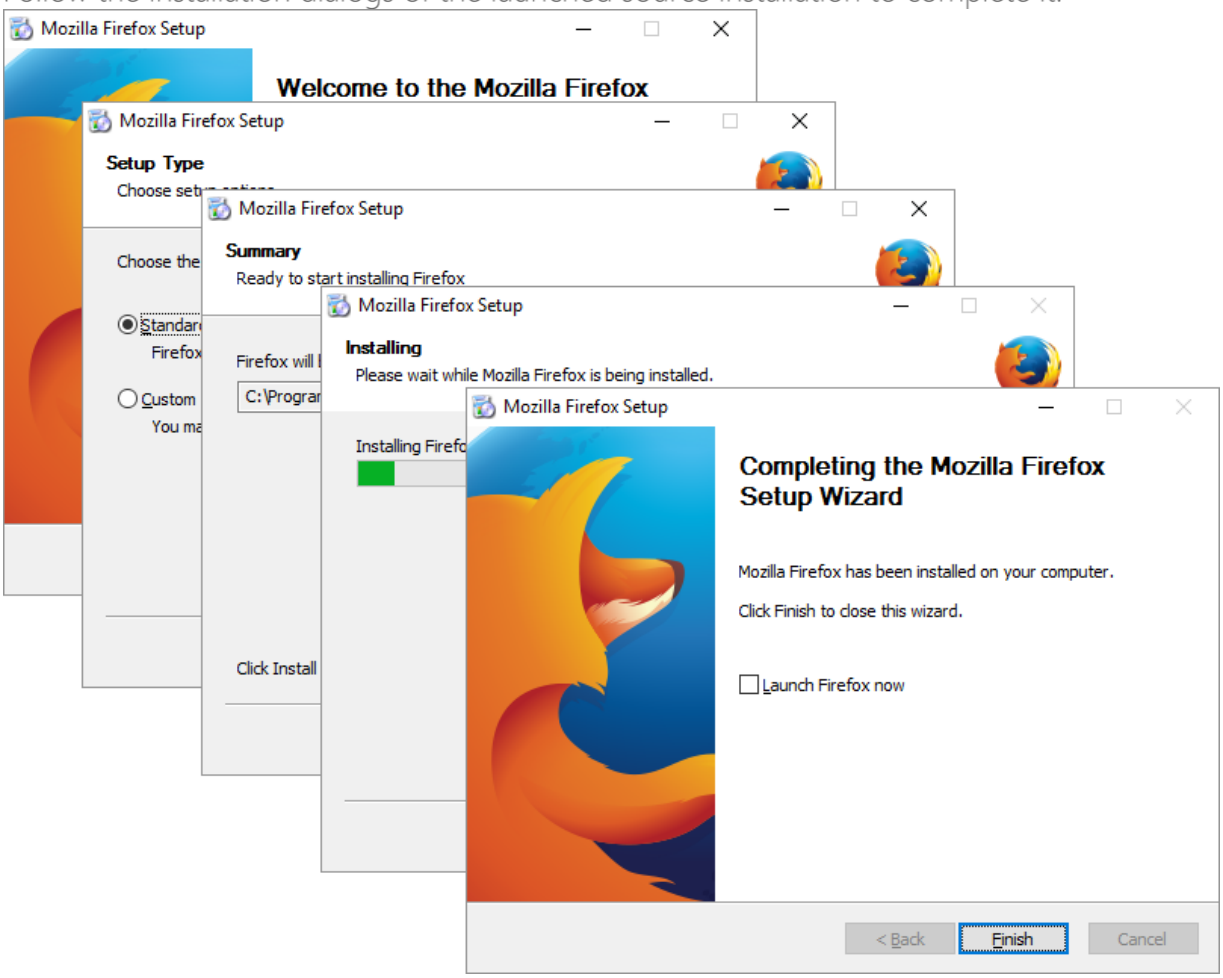
- Click Select and run... to choose source installation for repackaging.




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

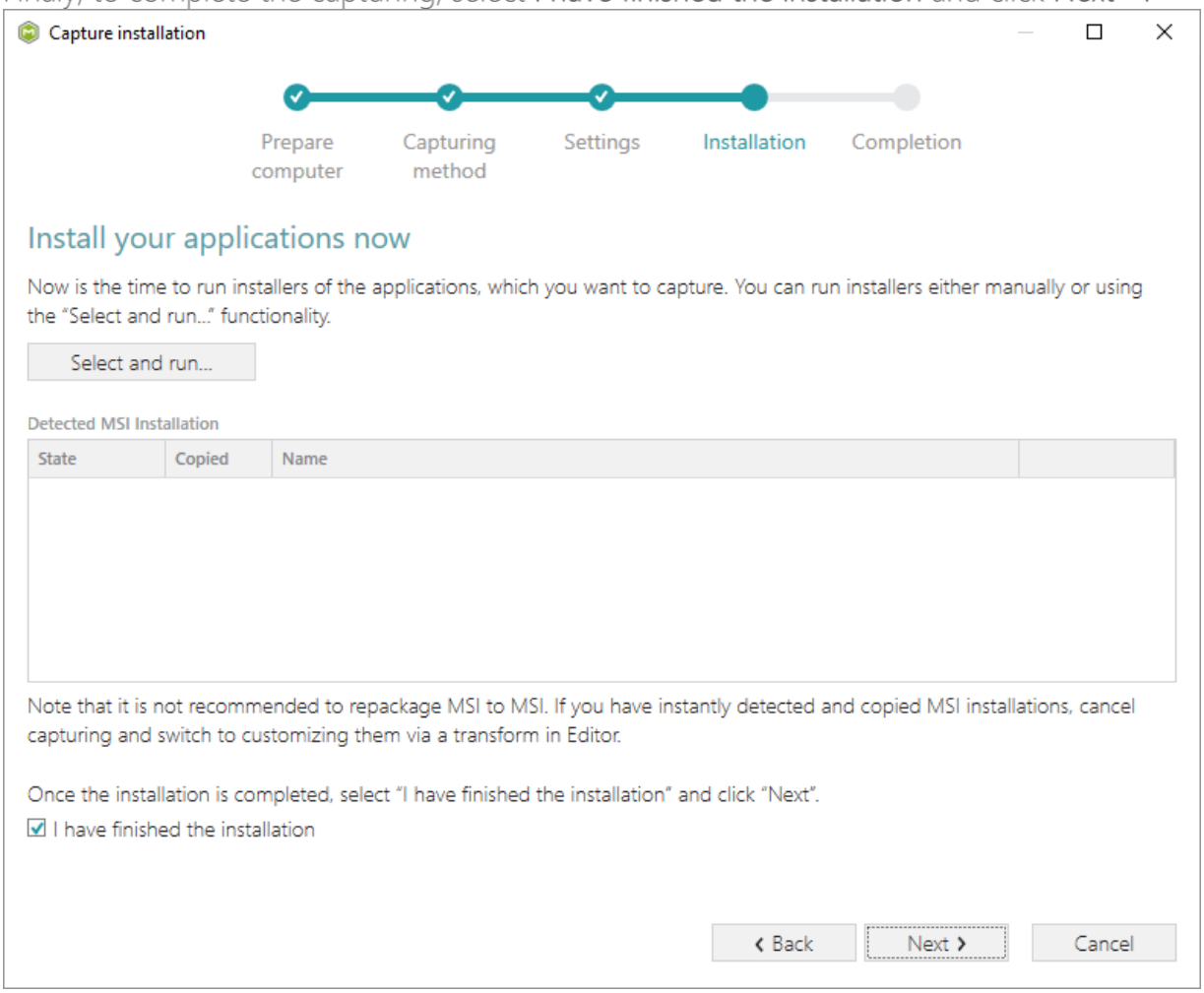


- Follow the installation dialogs of the launched source installation to complete it.

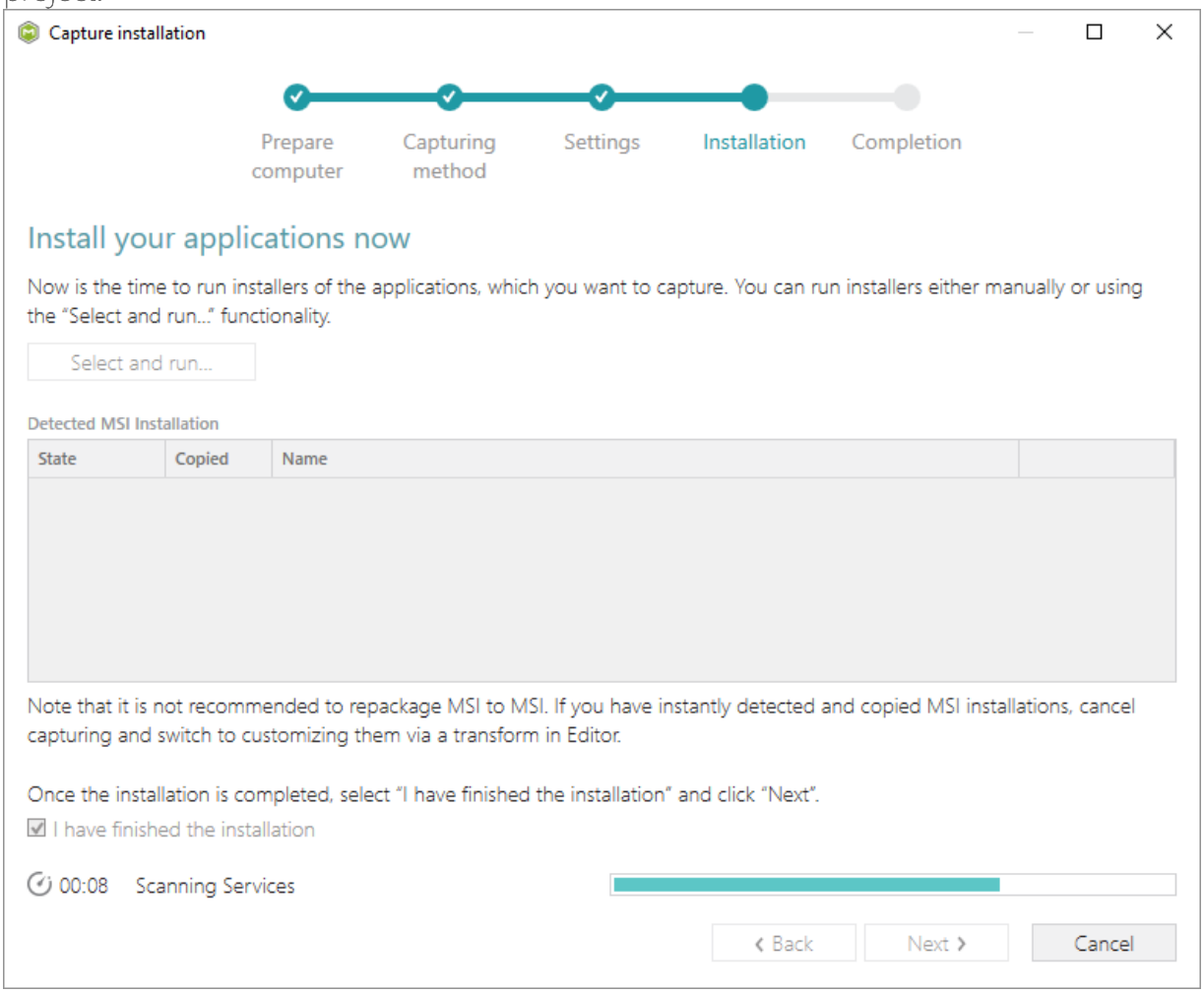



- Now you can make any changes to the file system and registry, which you want to capture and include to the package. For instance, you can create new or copy-paste existing files, import REG file to the system registry, changes permission settings, or launch the installed application in order to capture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

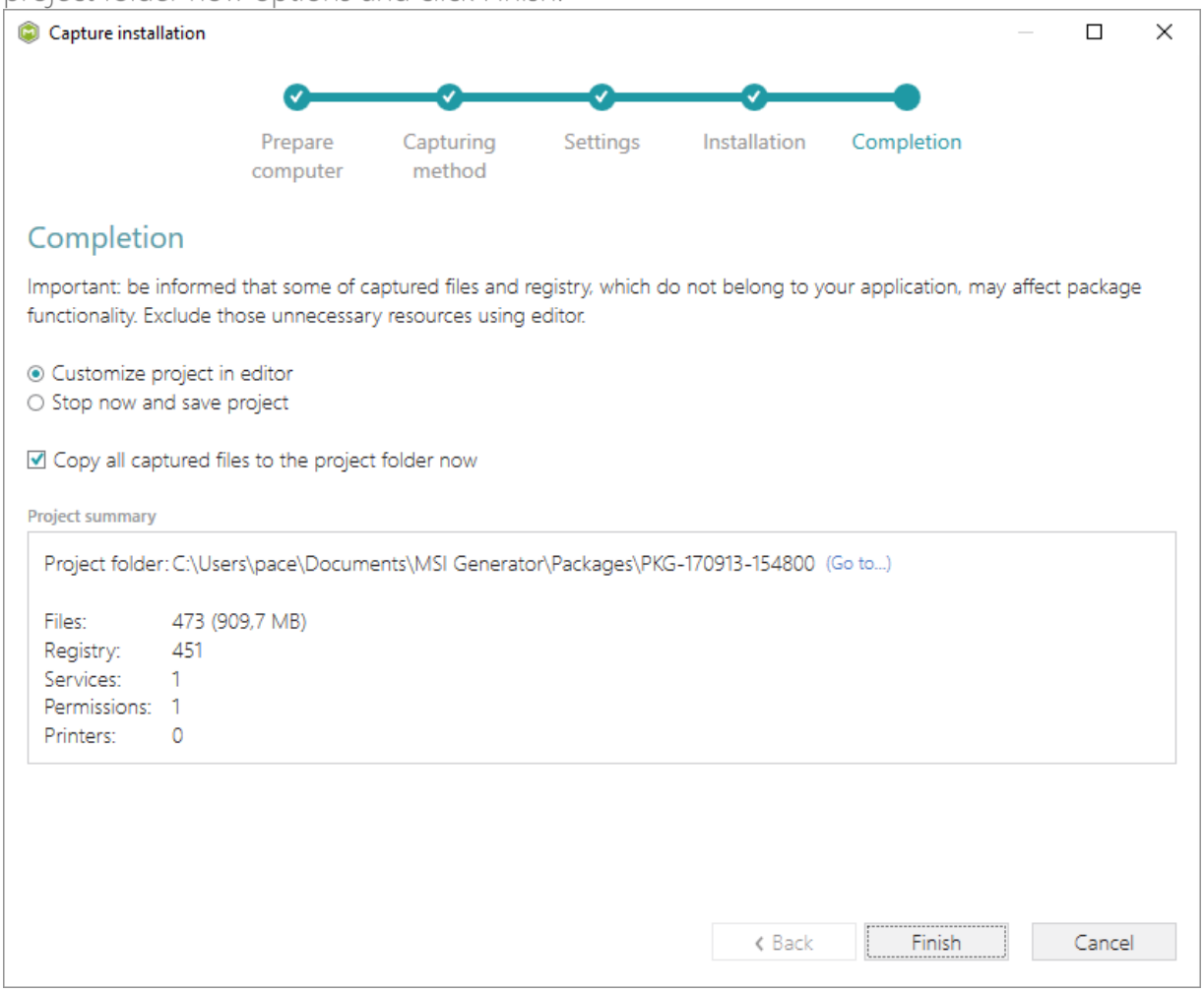
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



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

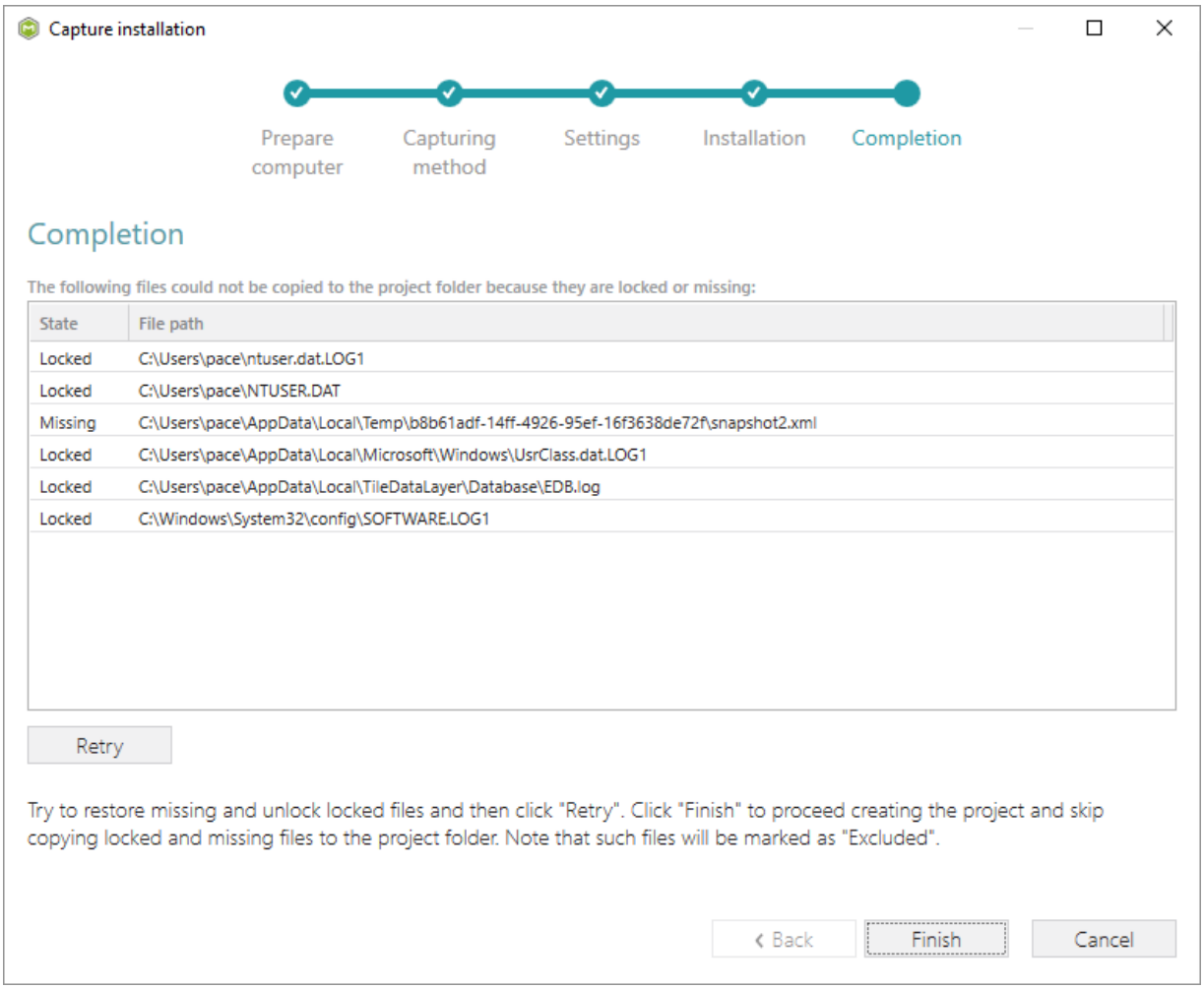


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



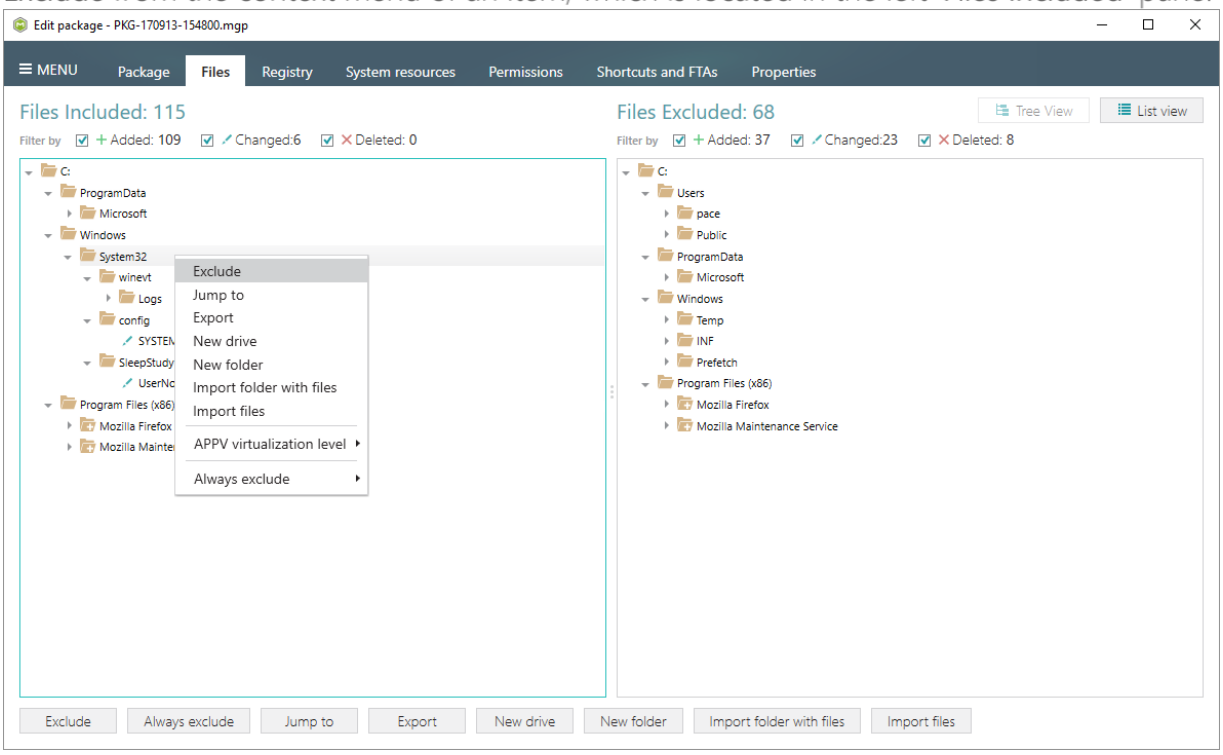

- The following dialog displays captured files, which could not be copied to the project folder because they do not exist anymore or locked by the system or by an application. Try to resolve these issues and then click Retry. Click Finish to skip copying the missing and locked

files and mark them as Excluded resources.

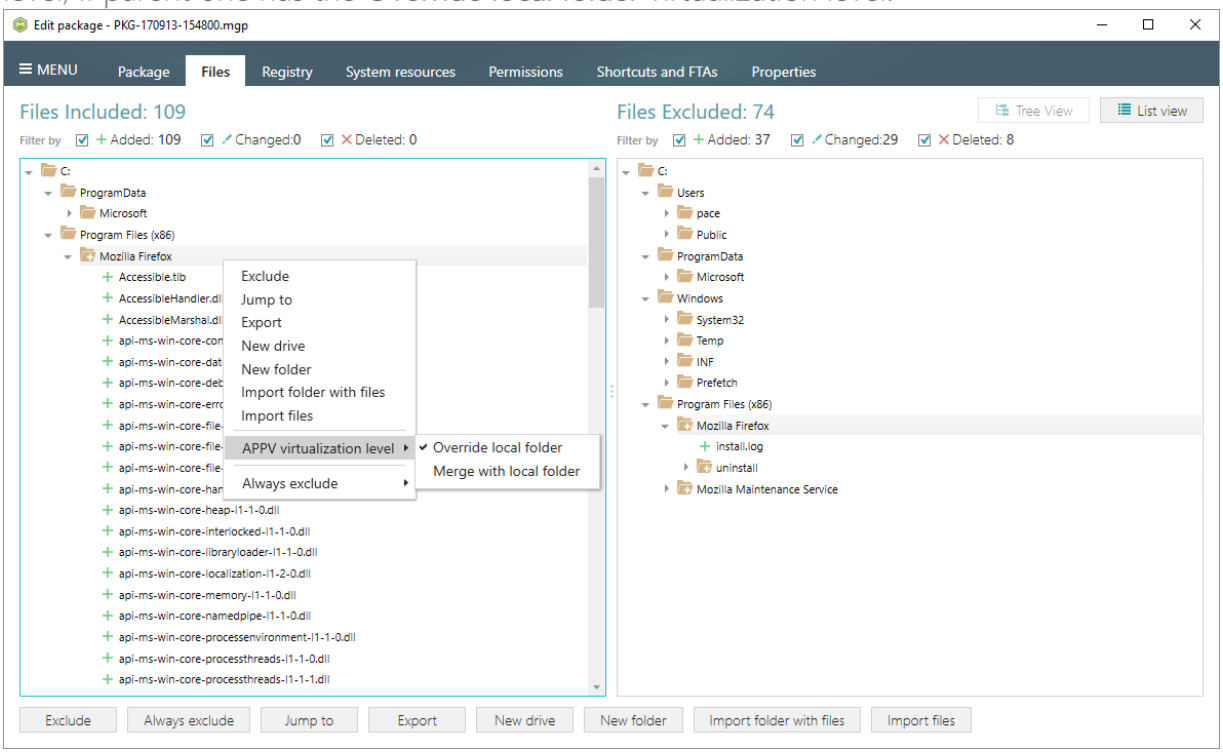



- Once the project is opened in the project editor, review the captured resources at the Files, Registry, System resources, Permissions, and Shortcuts and FTAs tabs and exclude unnecessary ones from the project. Unnecessary resources are files, registry entries, which are usually created or modified in the result of operating system work, and such resources could not be a part of your captured application. Unfortunately, there is no universal rule to discover which of captured files or registry entries should be excluded, so exclude only those ones, which almost 100% do not refer to your captured application (e.g. NOD32 antivirus files couldn't be a part of Firefox application).

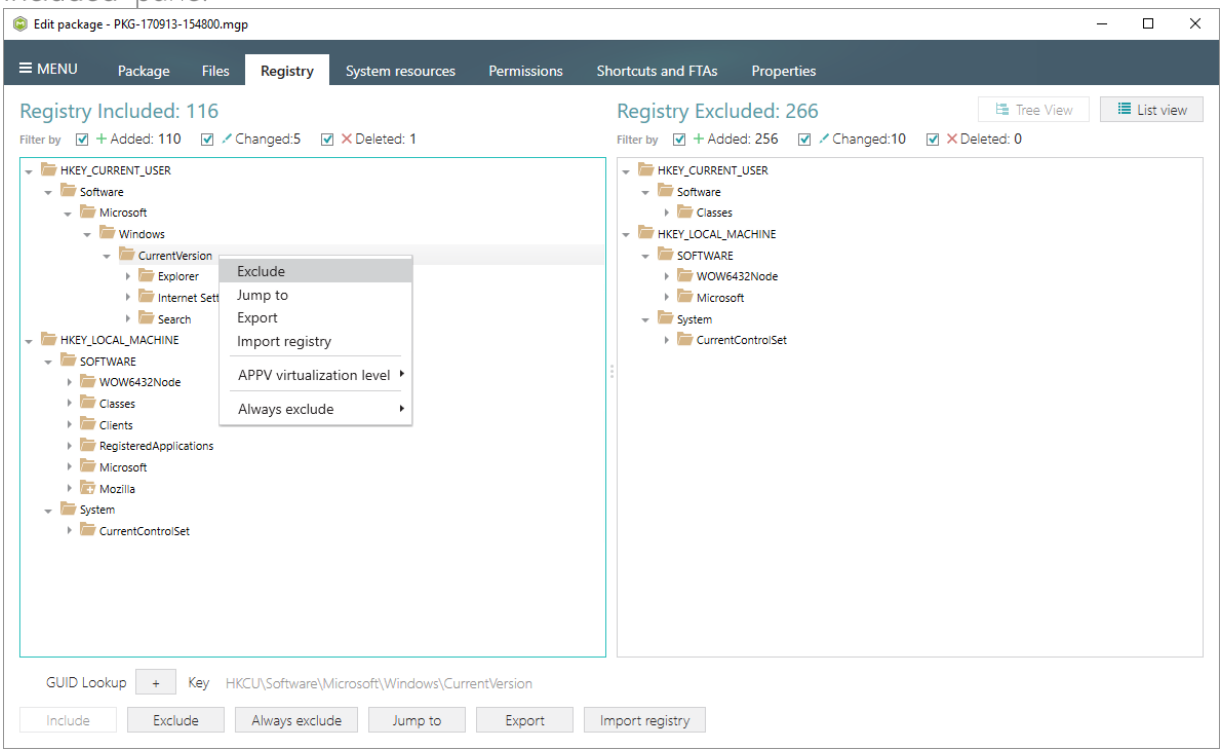
- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.



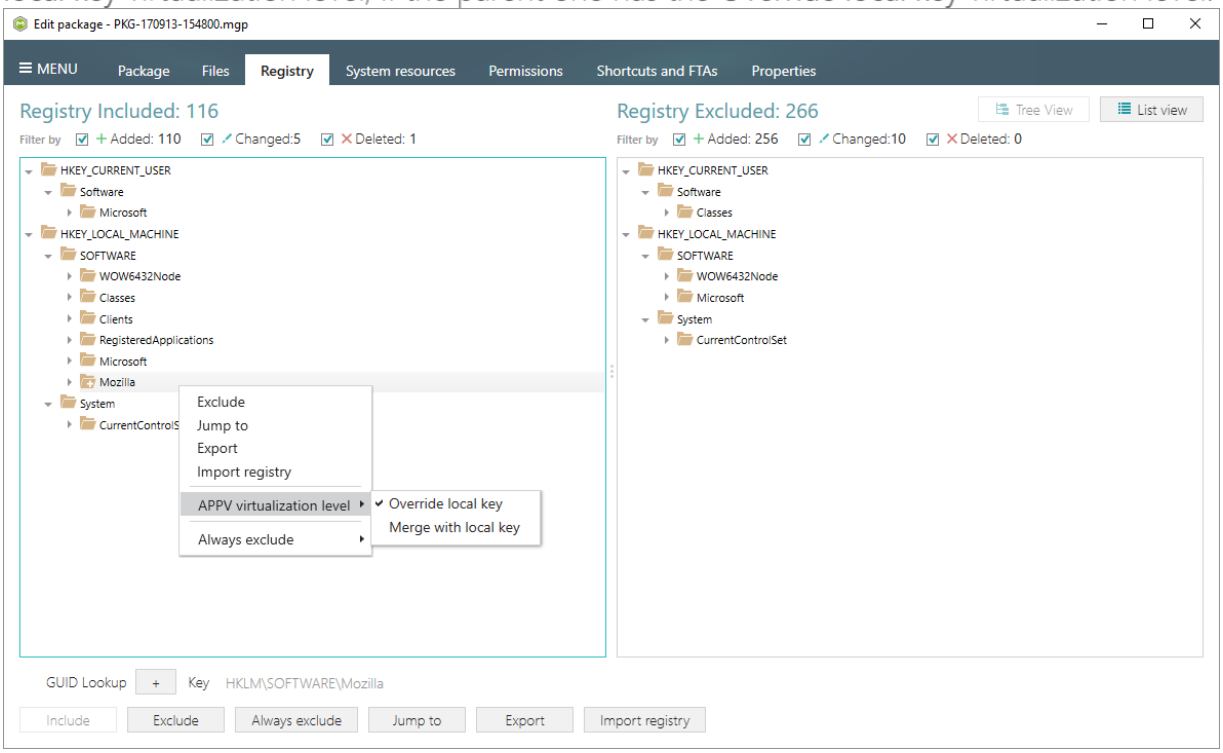

- In order to change the default virtualization (isolation) level of a folder, open the folder context menu, and select either **Override local folder** or **Merge with local folder** option from the **APPV virtualization level** submenu. Note that if the **APPV virtualization level** submenu is disabled with the **Override local folder** option selected, it means that this option is inherited from the parent folder. Child folder could not have the **Merge with local folder** virtualization level, if parent one has the **Override local folder** virtualization level.



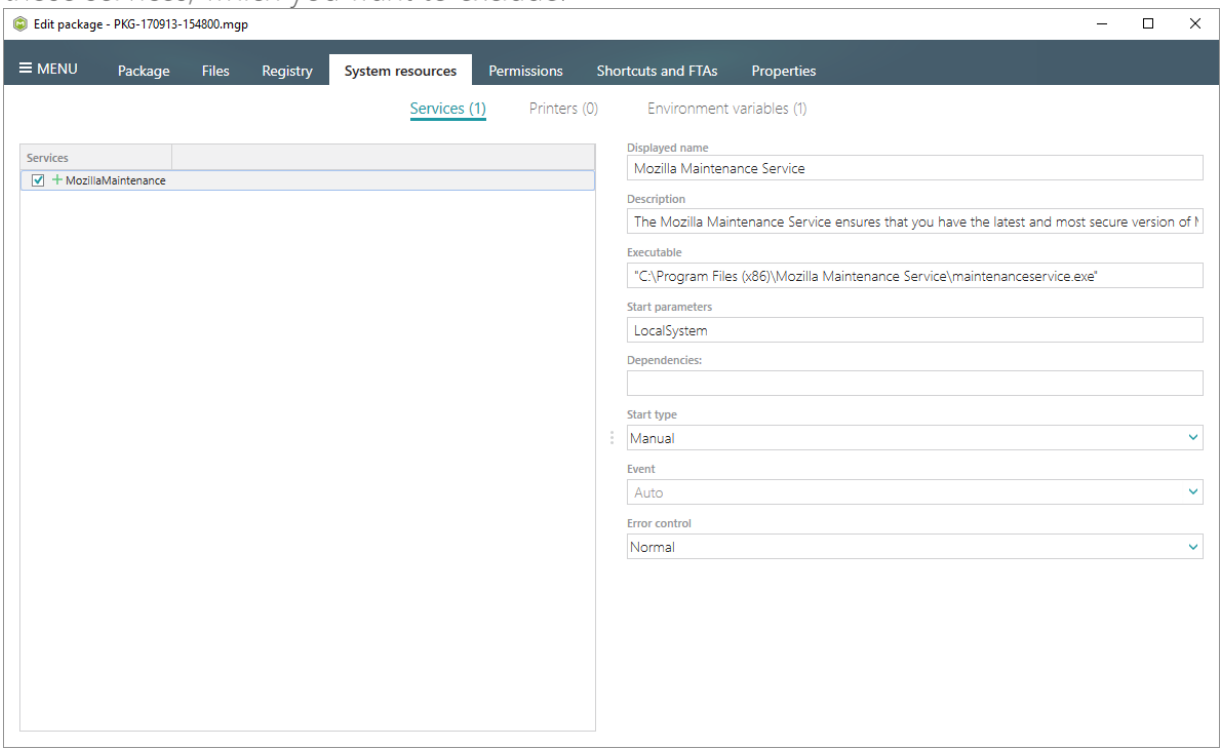
- In order to review and exclude unnecessary registry keys or values, go to the **Registry** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane.



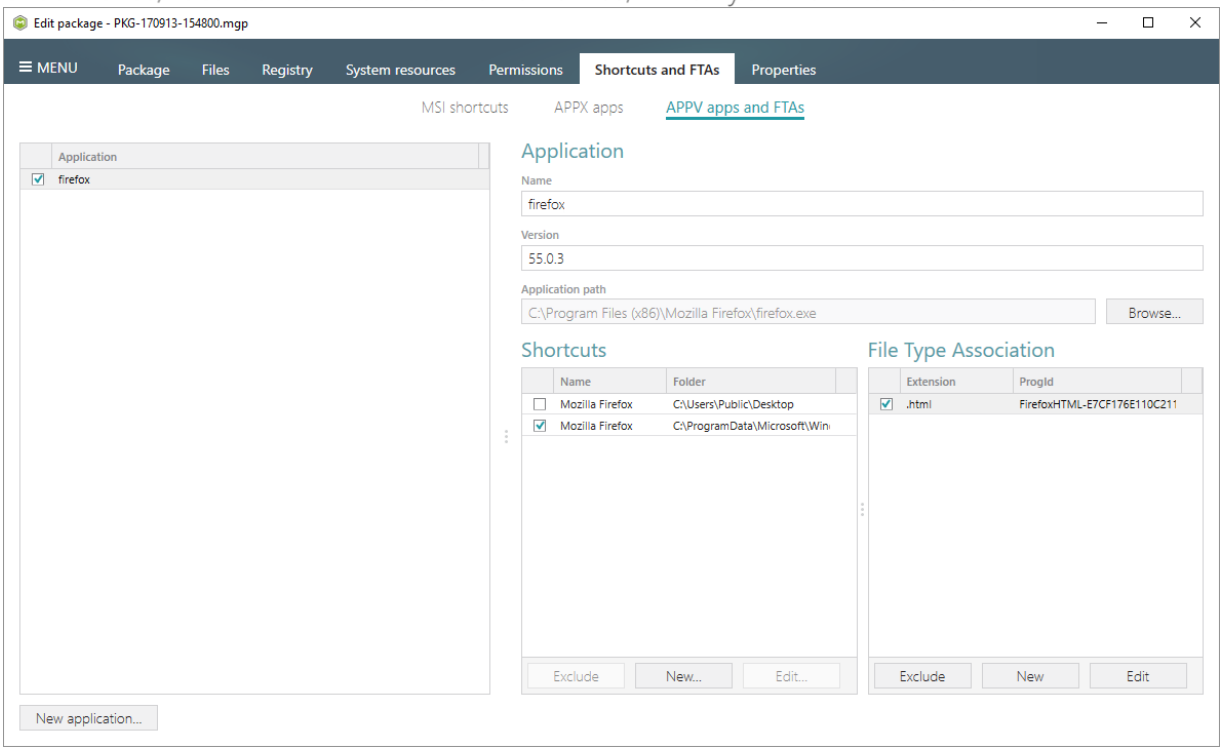

- In order to change the default virtualization (isolation) level of a registry key, open the registry key context menu, and select either **Override local key** or **Merge with local key** option from the **APPV virtualization level** submenu. Note that if the **APPV virtualization level** submenu is disabled with the **Override local key** option selected, it means that this option is inherited from the parent registry key. A child registry key could not have the **Merge with local key** virtualization level, if the parent one has the **Override local key** virtualization level.



- 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.

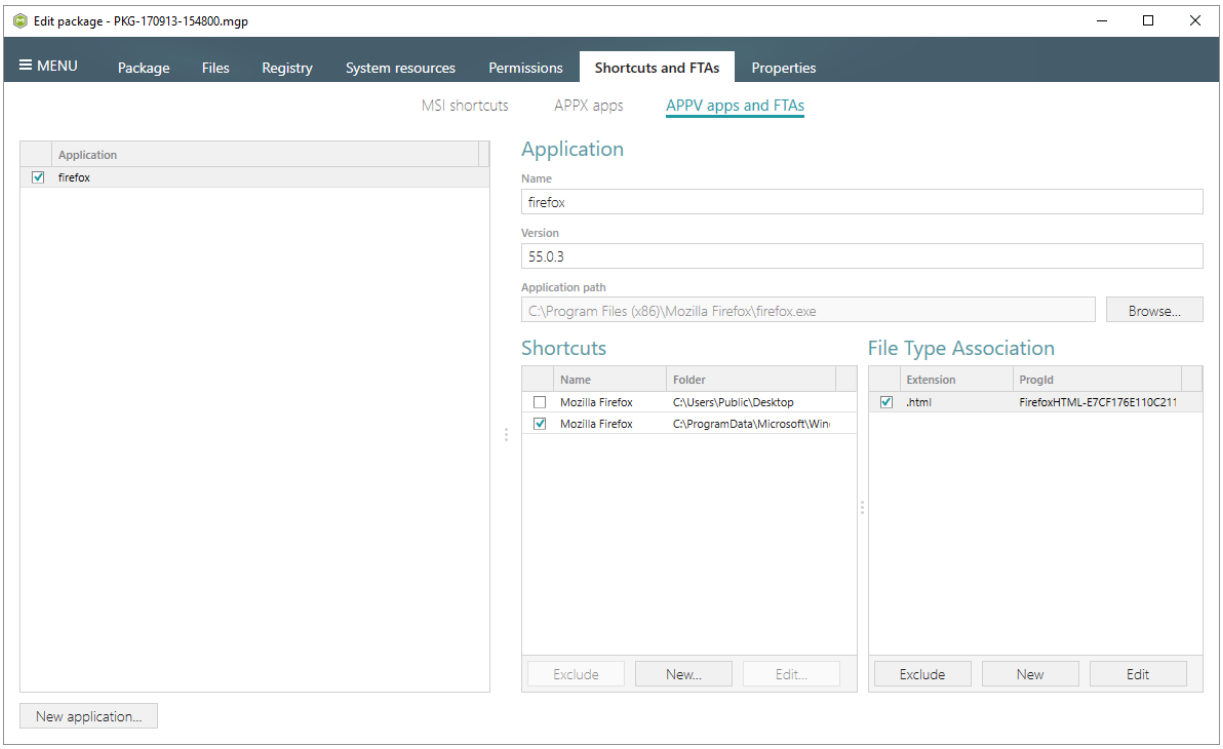



- In order to review and exclude unnecessary APPV applications, shortcuts and their file type associations, go to the **Shortcuts and FTAs -> APPV apps and FTAs** tab, and uncheck checkboxes, located before those items names, which you want to exclude.

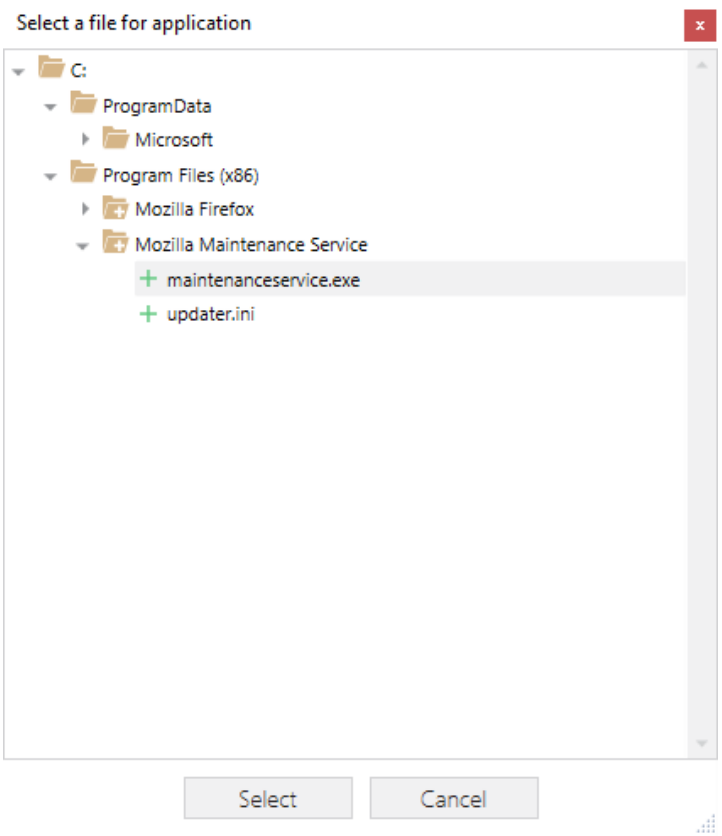



- If you need to add a new APPV application, its shortcuts and associate file extensions with this application, click **New application...** on the **Shortcuts and FTAs -> APPV apps and FTAs**

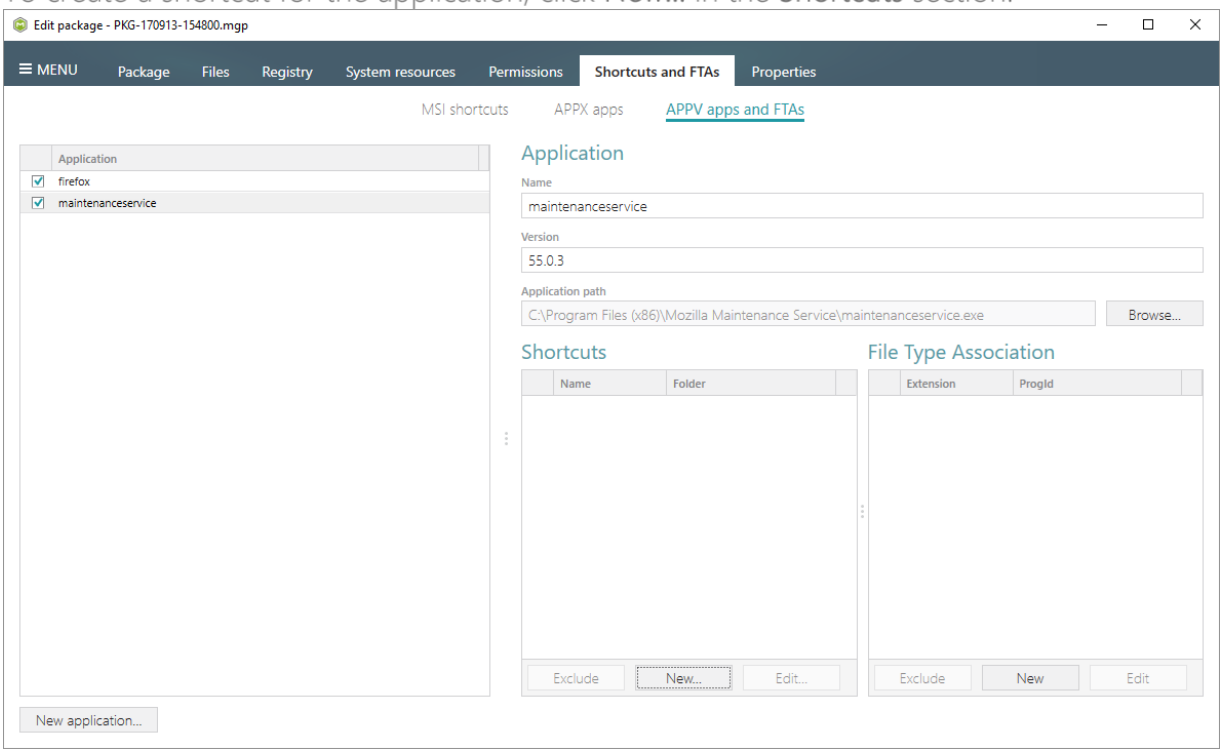
tab.



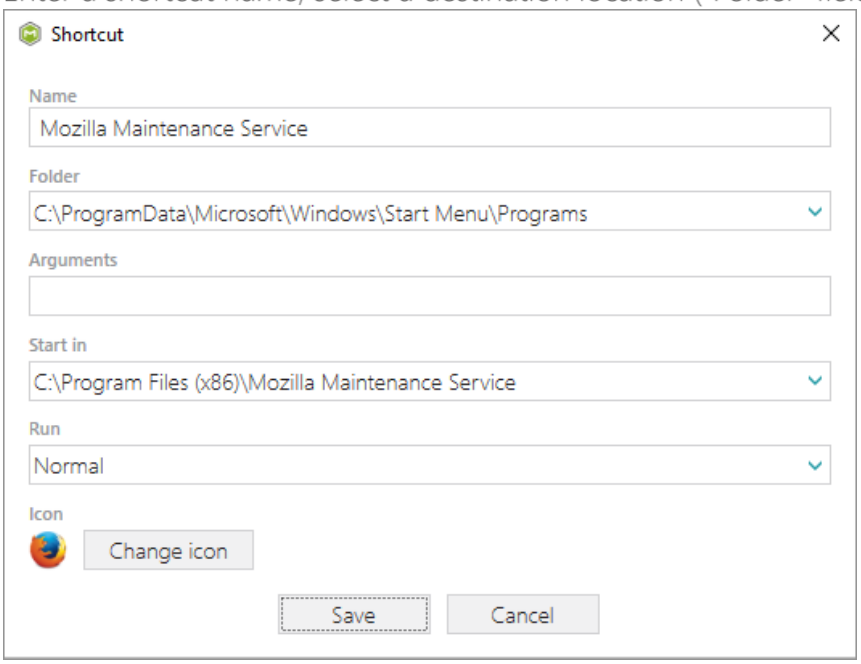

- Select the executable file (EXE) from the tree and click **Select**.



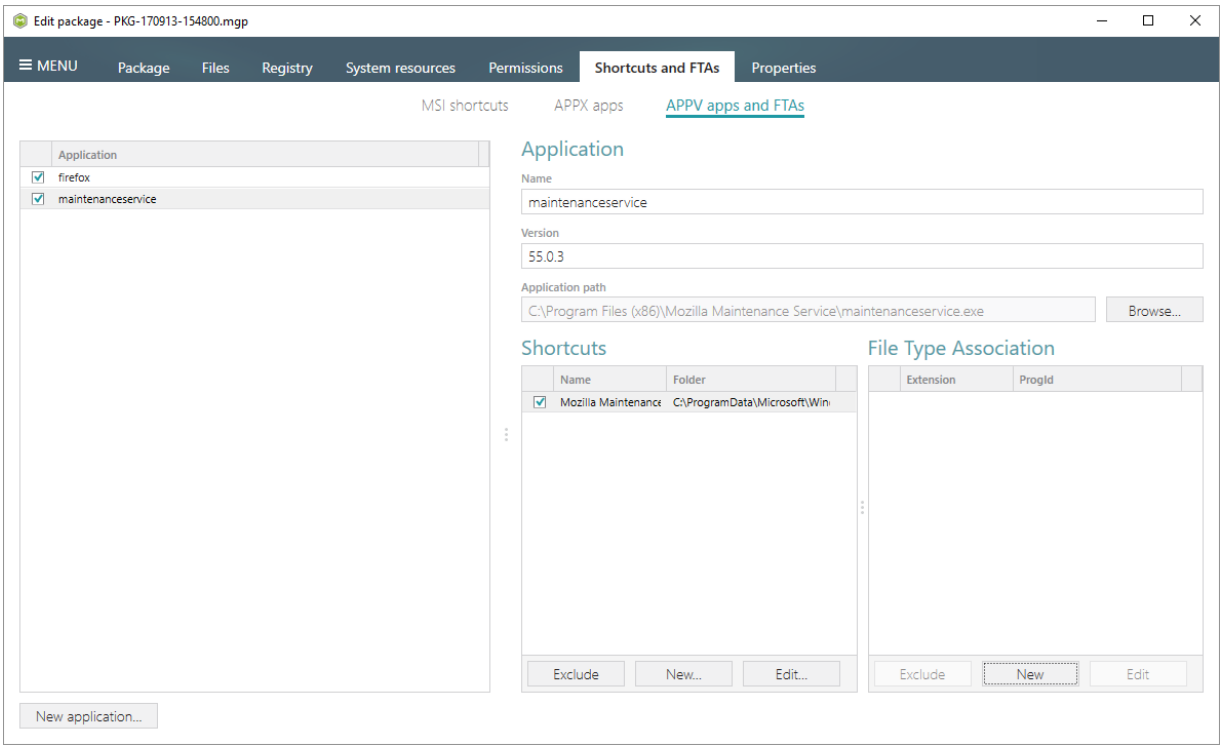
- To create a shortcut for the application, click **New...** in the **Shortcuts** section.



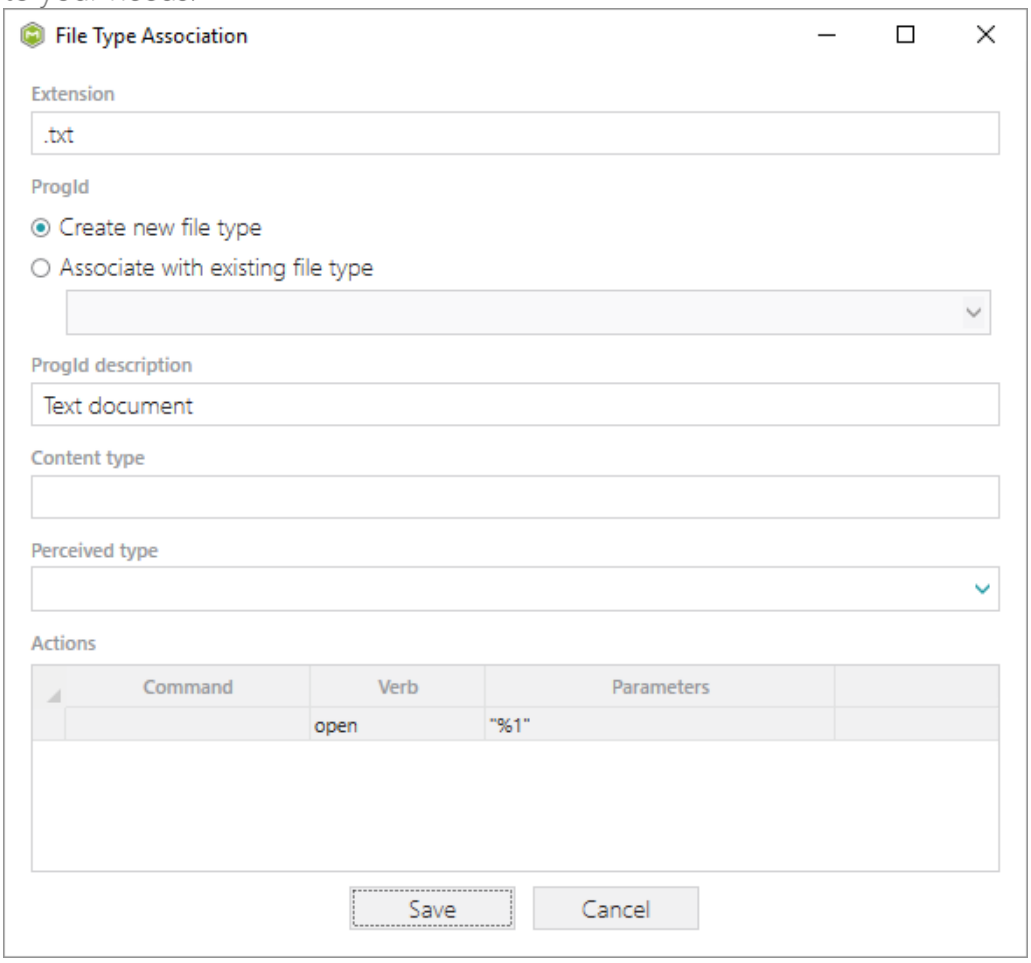

- Enter a shortcut name, select a destination location ("Folder" field), icon and click **Save**.



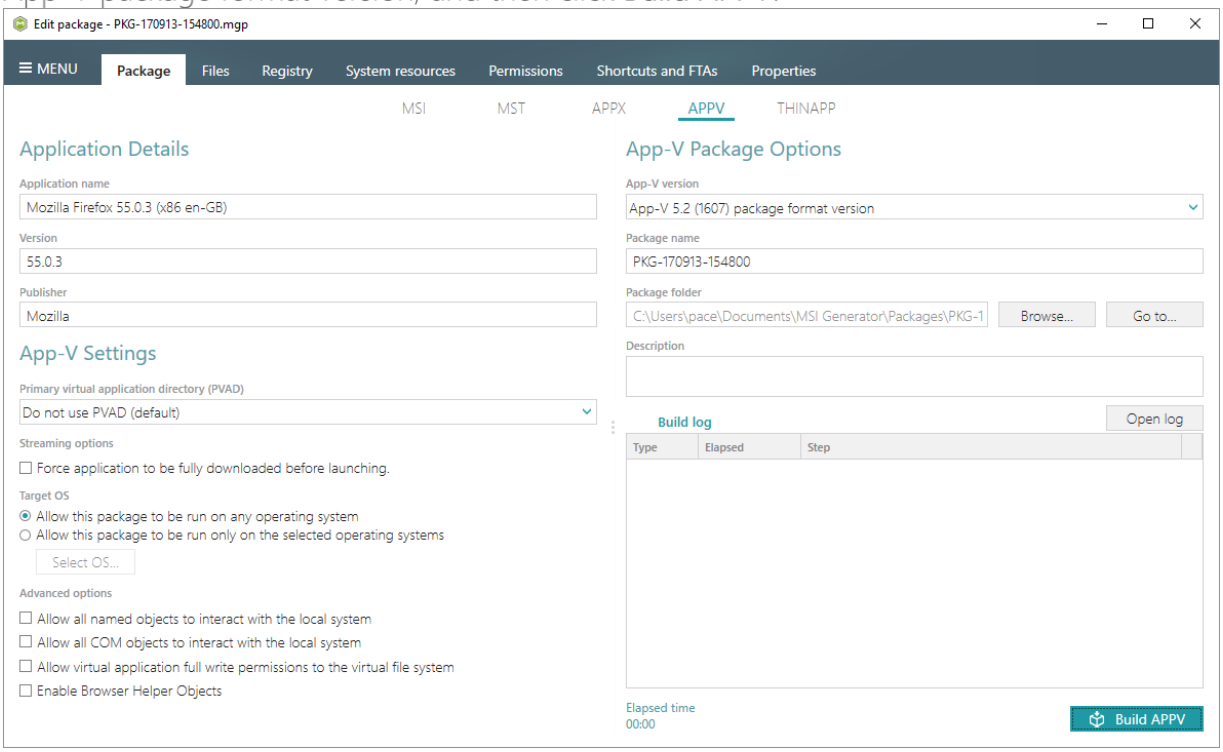
- To associate some file extension with your application, click **New...** in the **File Type Association** section.



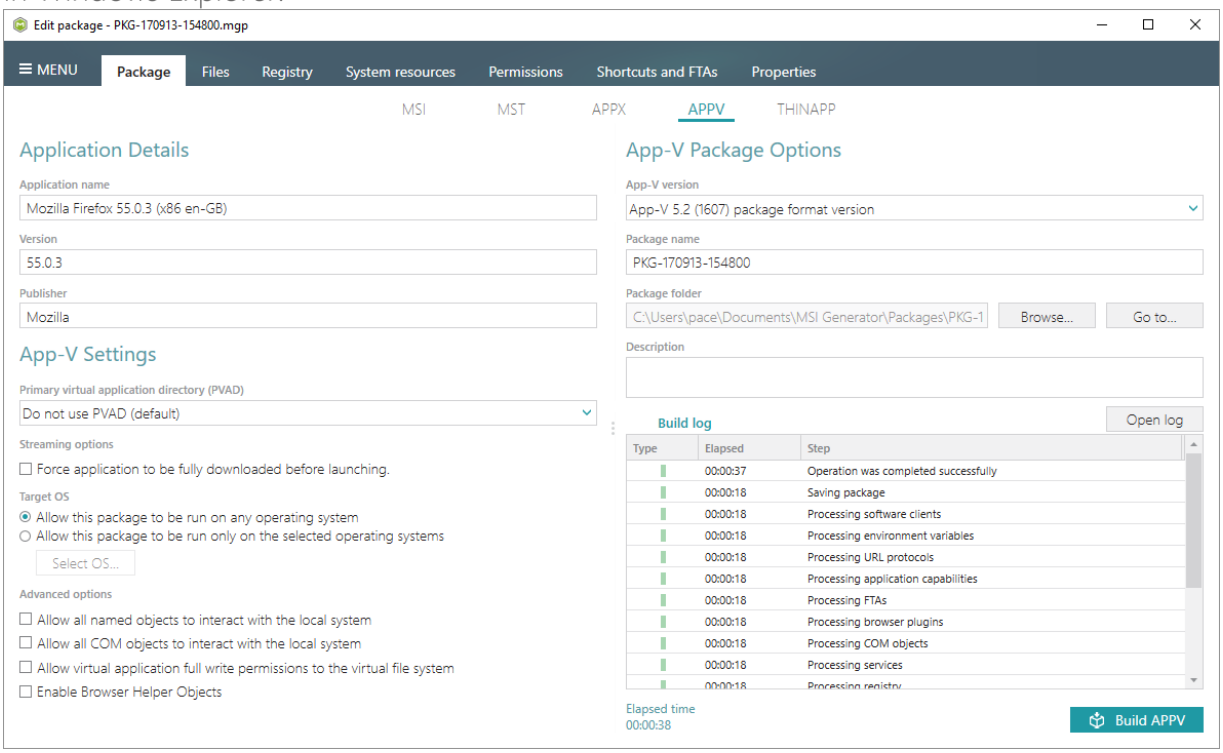

- Enter a file extension and click **Save**. Optinally you can modify other FTA settings according to your needs.



- Finaly, to build APPV package from the project, navigate to the **Package** -> **APPV** tab, review and update (if necessary) **Application Details**, select the necessary **App-V Settings**, App-V package format version, and then click **Build APPV**.




- 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.

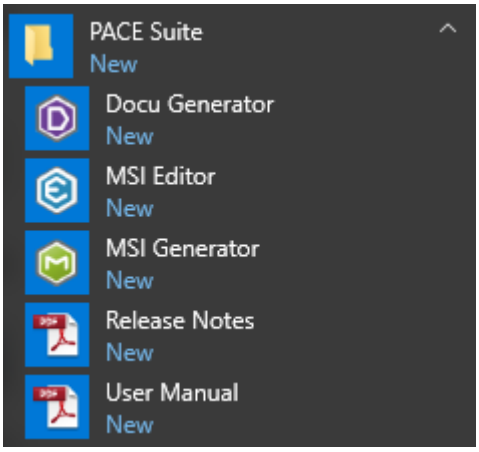
Choose a scenario that better suits your needs:

- 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.
- Convert (Repackage) EXE, MSI to THINAPP, described in section 3.6.2
  Repackage your source installation (EXE, MSI, VBS, CMD, etc.) into THINAPP package using MSI Generator.

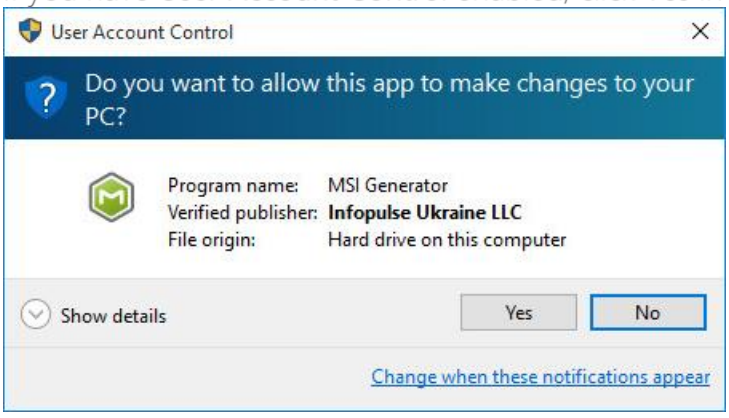
### 3.6.1 Create THINAPP from Scratch

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.

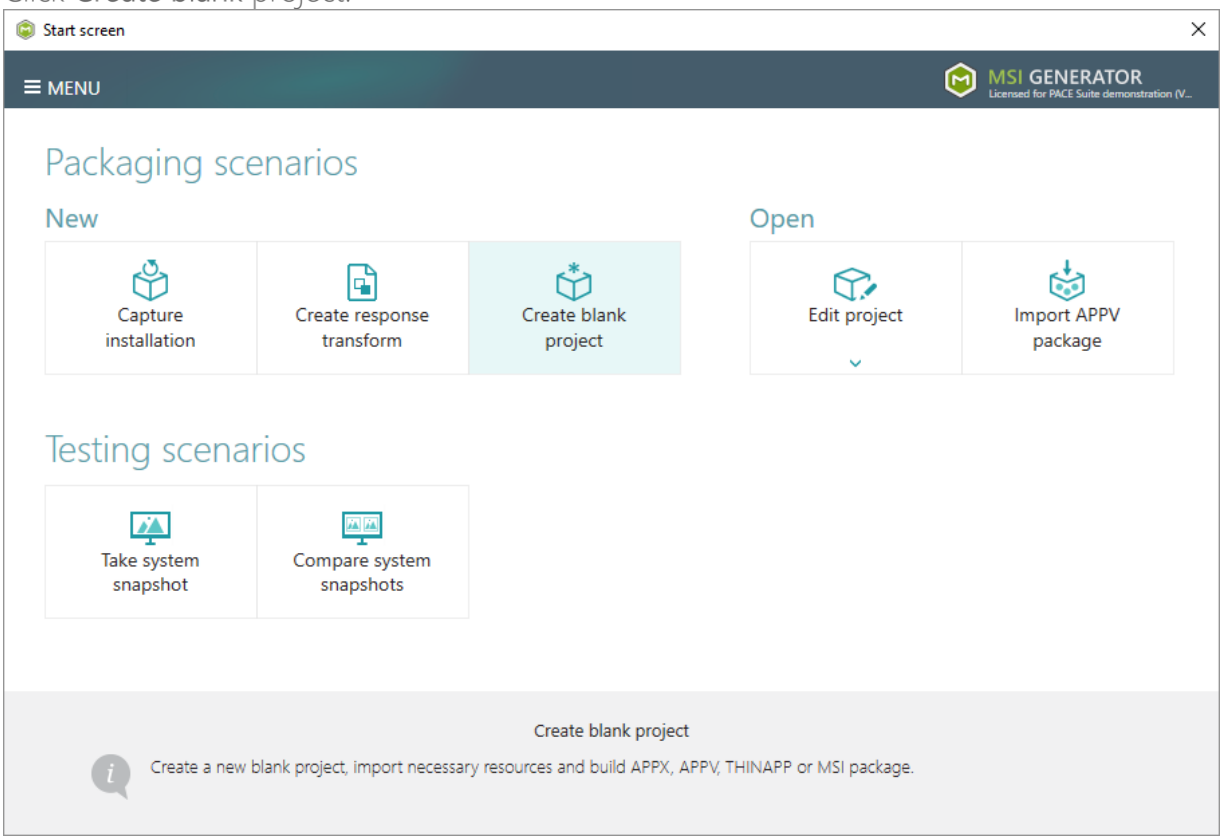
- Launch MSI Generator from the desktop or the start menu shortcut.



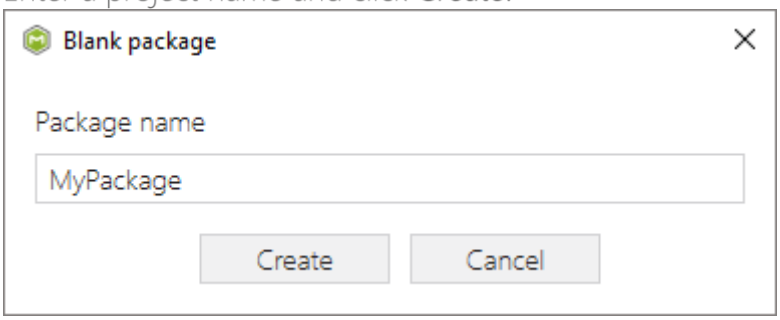

- If you have User Account Control enabled, click Yes in the opened window.



- Click Create blank project.

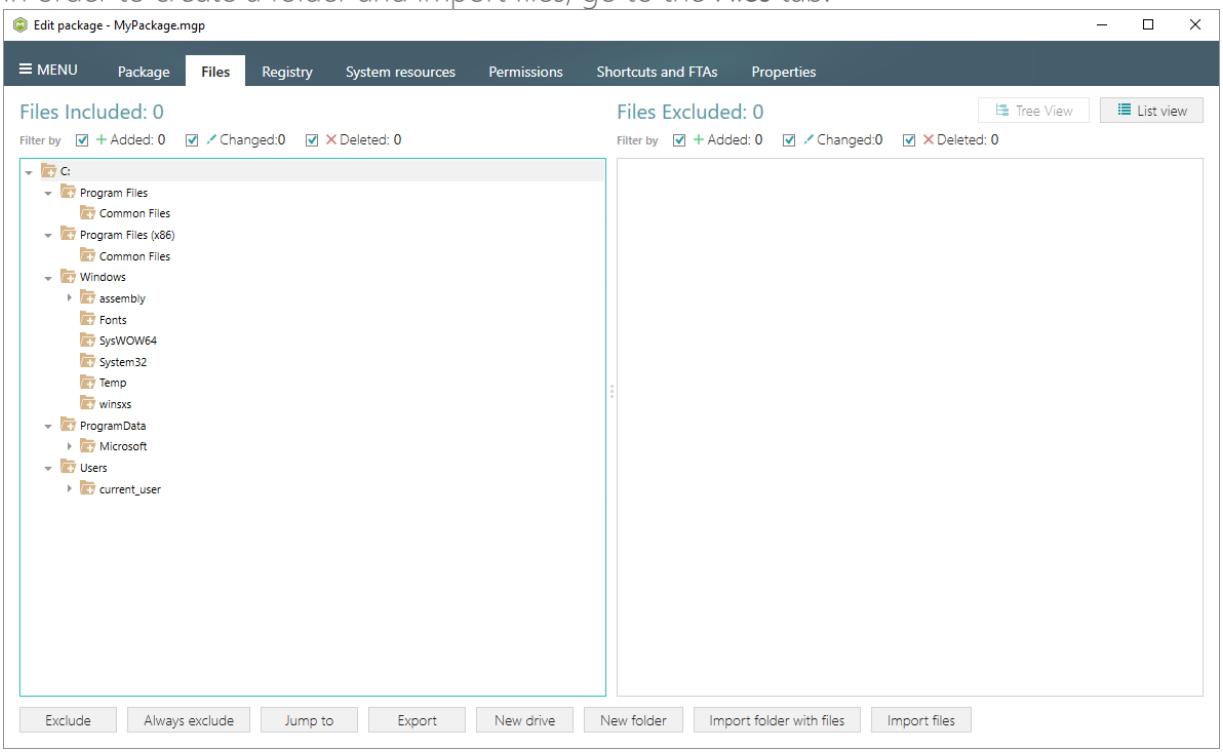



- Enter a project name and click Create.

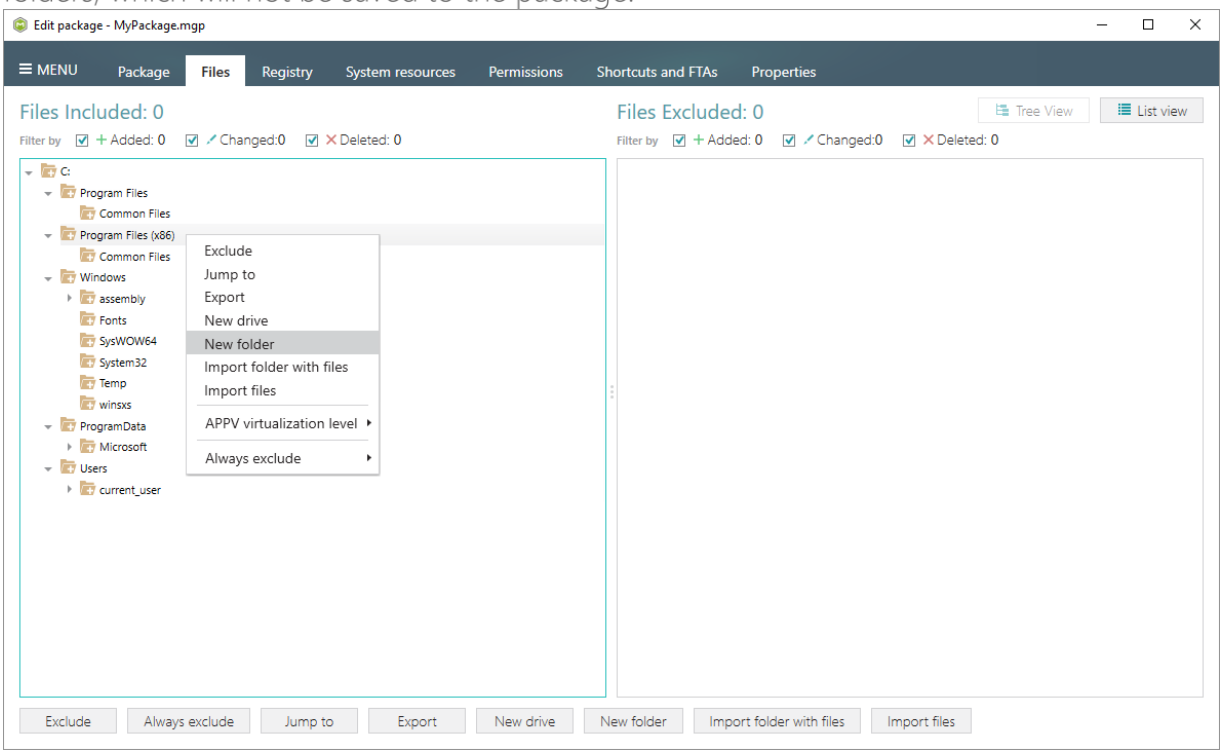



- Once the newly created project is opened automatically, you can start adding the necessary resources using MSI Generator. The steps from [6] to [10] describe how to create a new folder and import files to this folder; the steps from [11] to [13] describe how to import registry entries from the REG file; and the steps [14] and [15] describe how to select entry points and build THINAPP package from the prepared project.

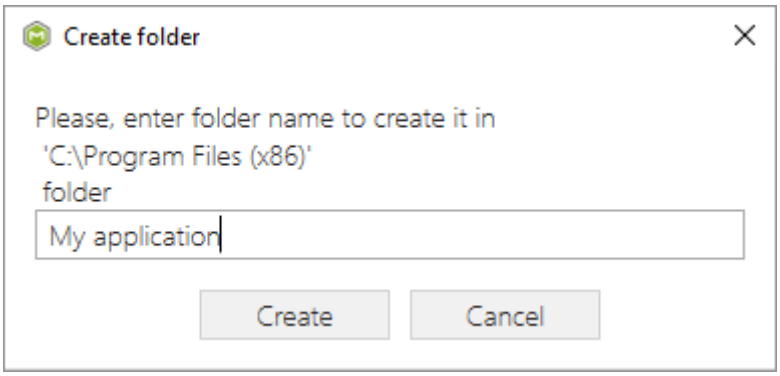
- In order to create a folder and import files, go to the **Files** tab.



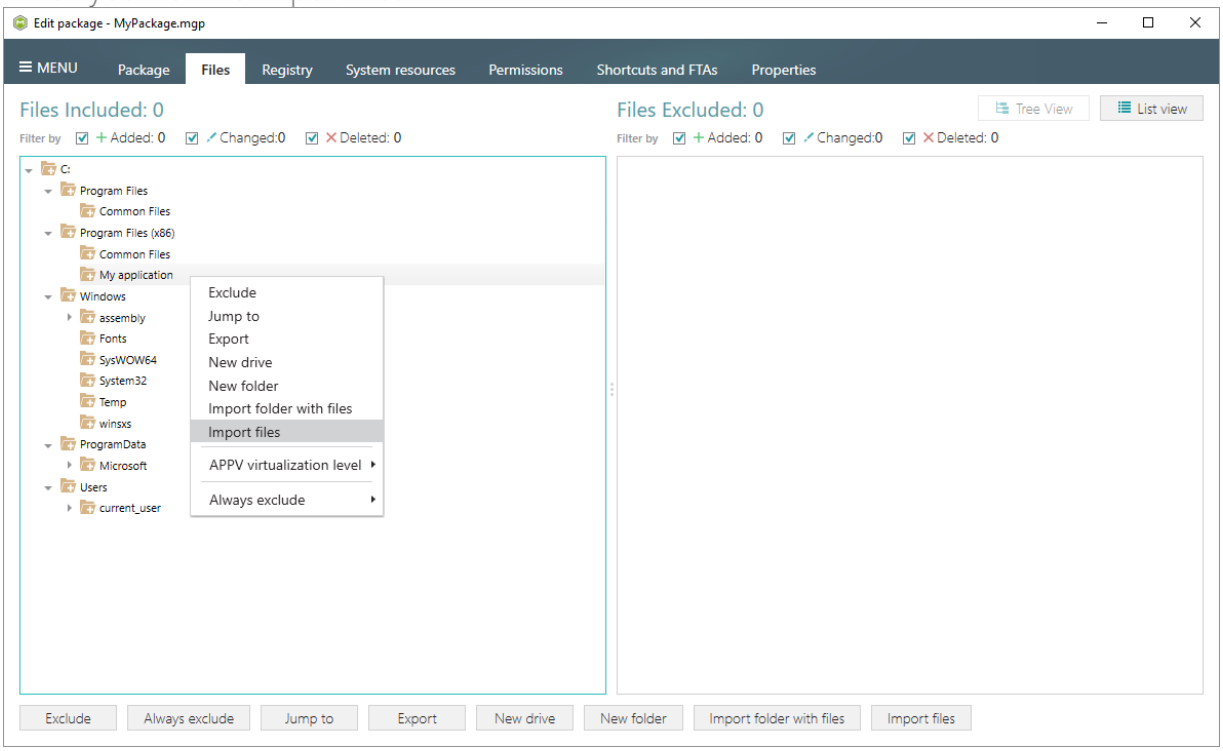

- In the left pane, select **New folder** from the context menu of a folder, where you want to create a new empty folder. Note that the left '**Files Included**' pane displays files and folders, which will be a part of your package, and the right '**Files Excluded**' pane displays files and folders, which will not be saved to the package.



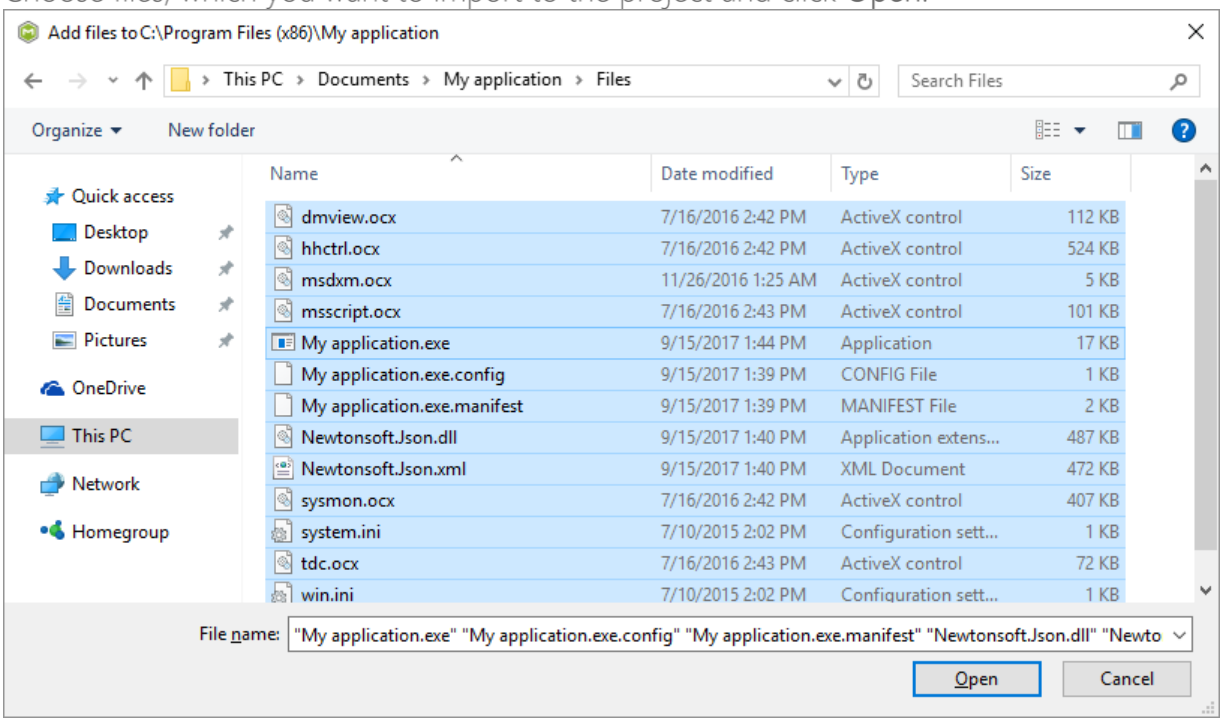
- Enter a folder name and click **Create**.



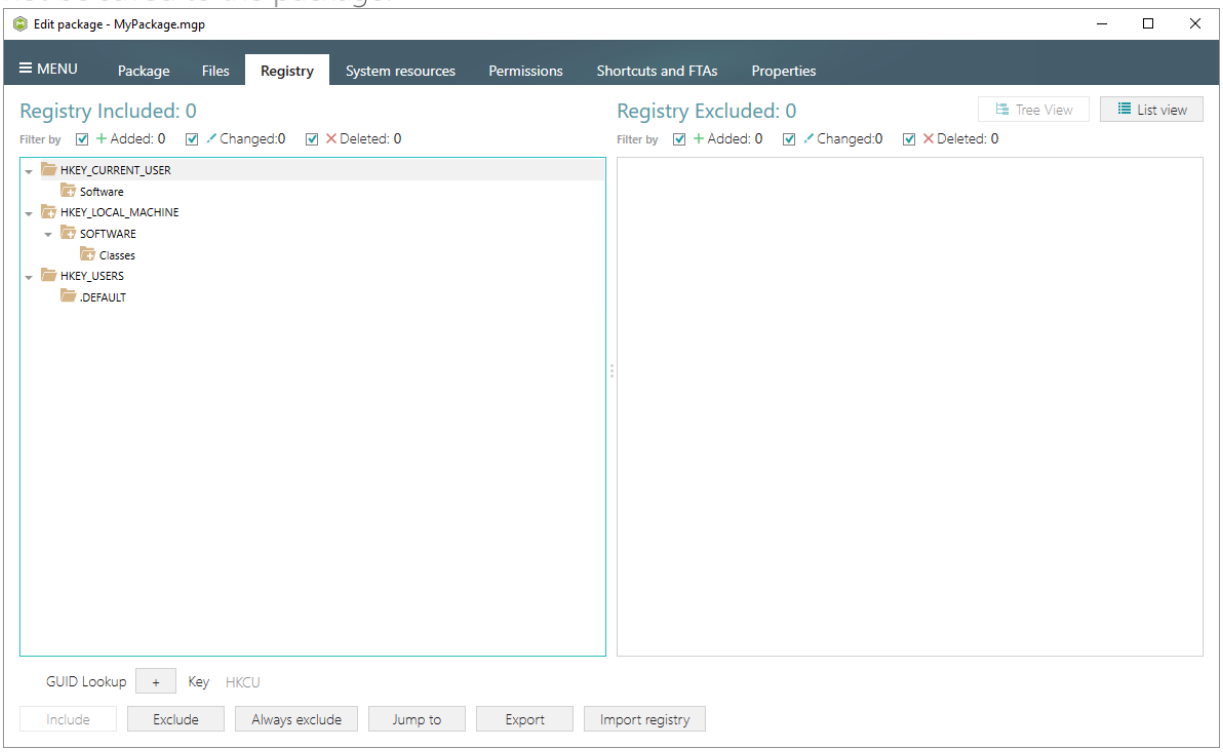

- For adding files to your project, select **Import files** from the context menu of a folder, to which you want to import files.



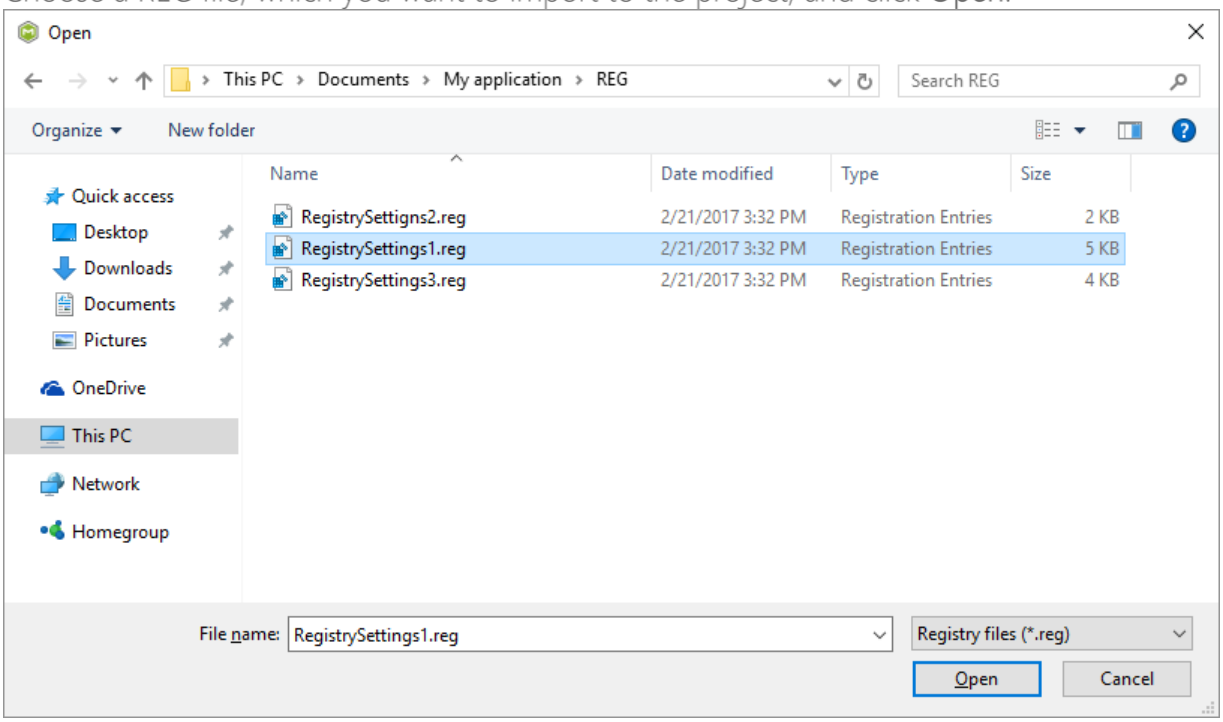
- Choose files, which you want to import to the project and click **Open**.



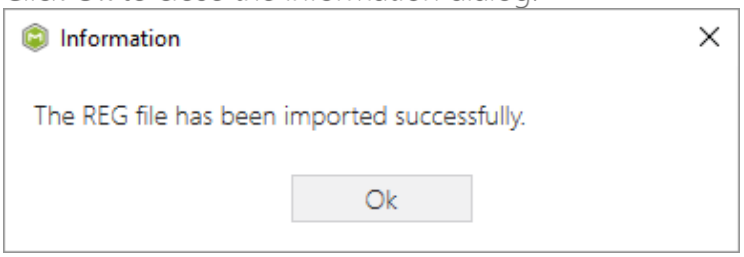

- In order to import registry entries from the REG file, go to the **Registry** tab and click the **Import registry** button, located in the bottom part of the window. Note that the left '**Registry Included**' pane displays registry keys and values, which will be a part of your package, and the right '**Registry Excluded**' pane displays registry keys and values, which will not be saved to the package.



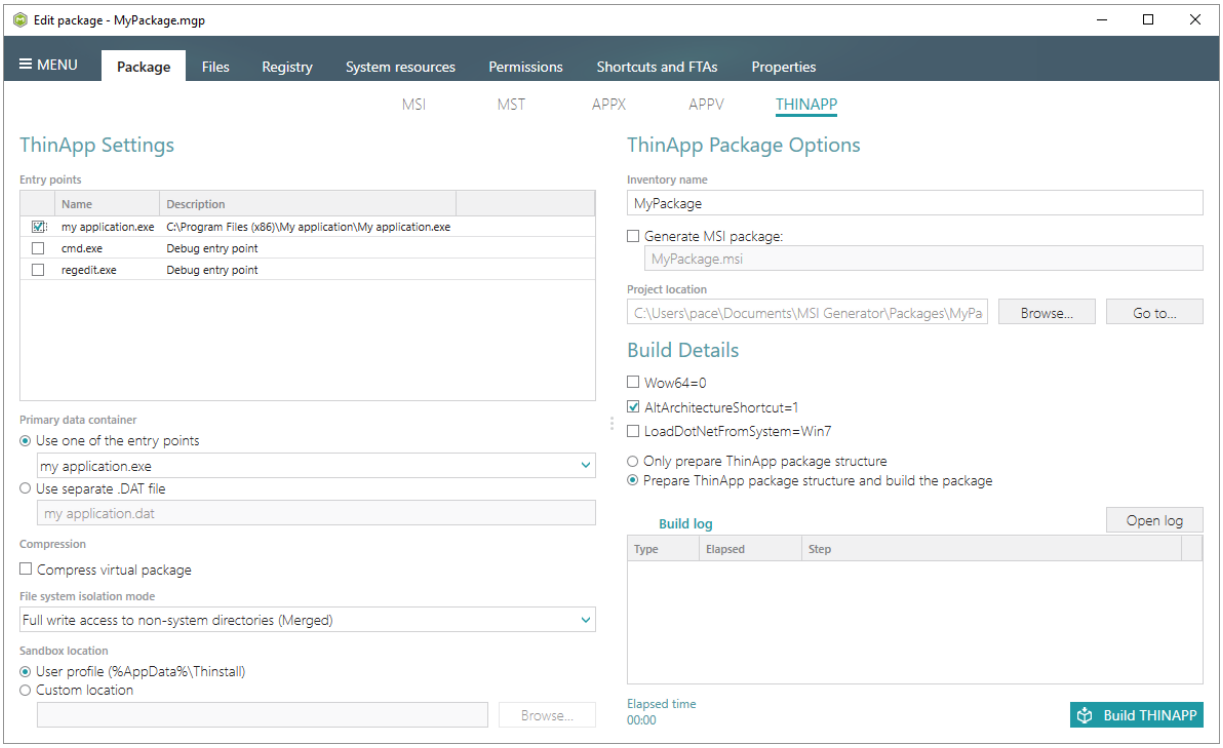
- Choose a REG file, which you want to import to the project, and click Open.




- Click Ok to close the information dialog.

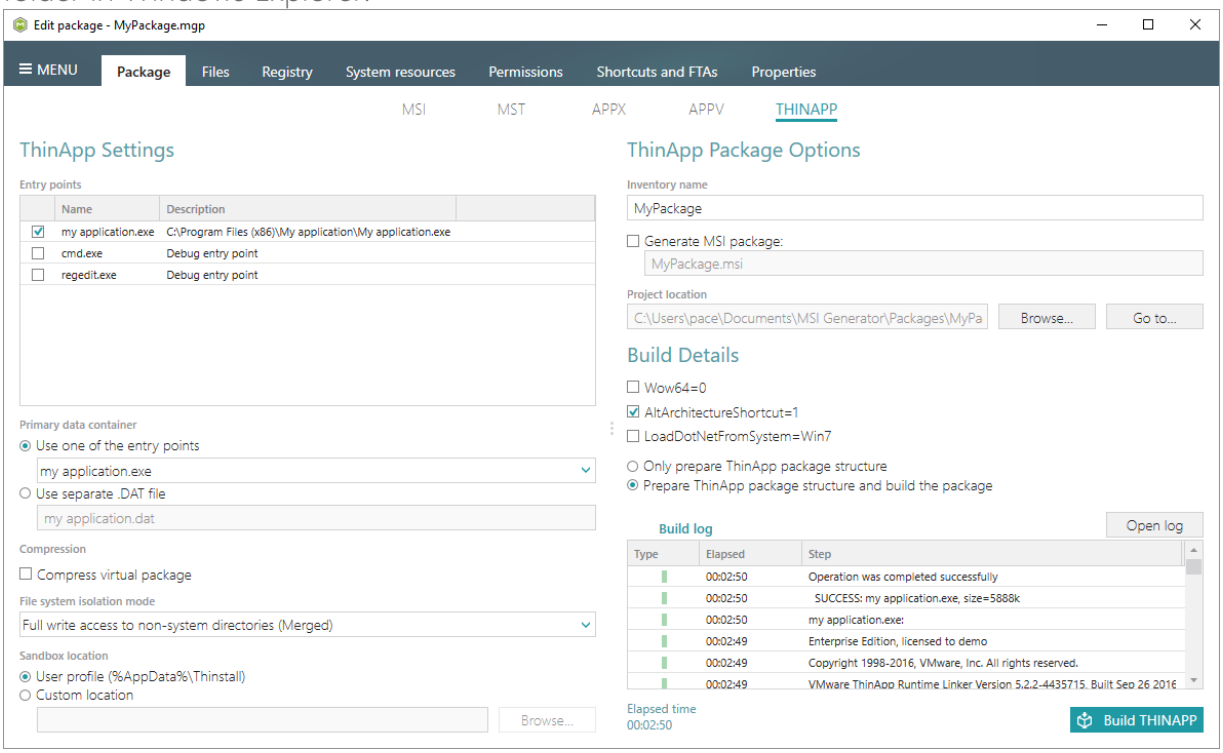



- In order to build THINAPP package from your project, navigate to the Package -> THINAPP tab, select Entry points (shortcuts), primary data container location and other package options, and then click Build THINAPP.




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. If you have installed this tool to the custom location, go to the MENU -> Settings, select the Custom ThinApp install location option and then click Browse... to choose the installation location of the VMware ThinApp tools.

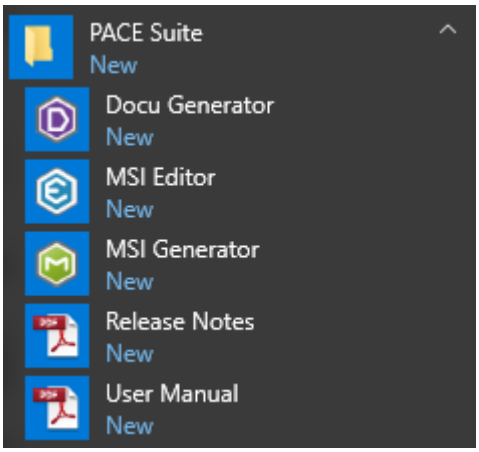
- Click Go to..., located next to the Project location field, to open the package containing folder in Windows Explorer.



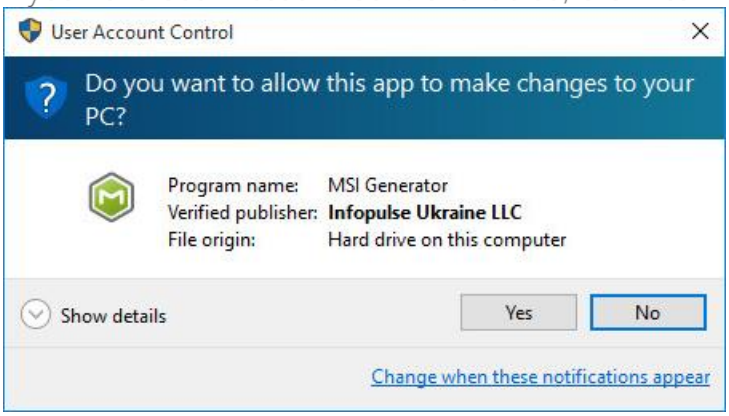
### 3.6.2 Convert (Repackage) EXE, MSI to THINAPP

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

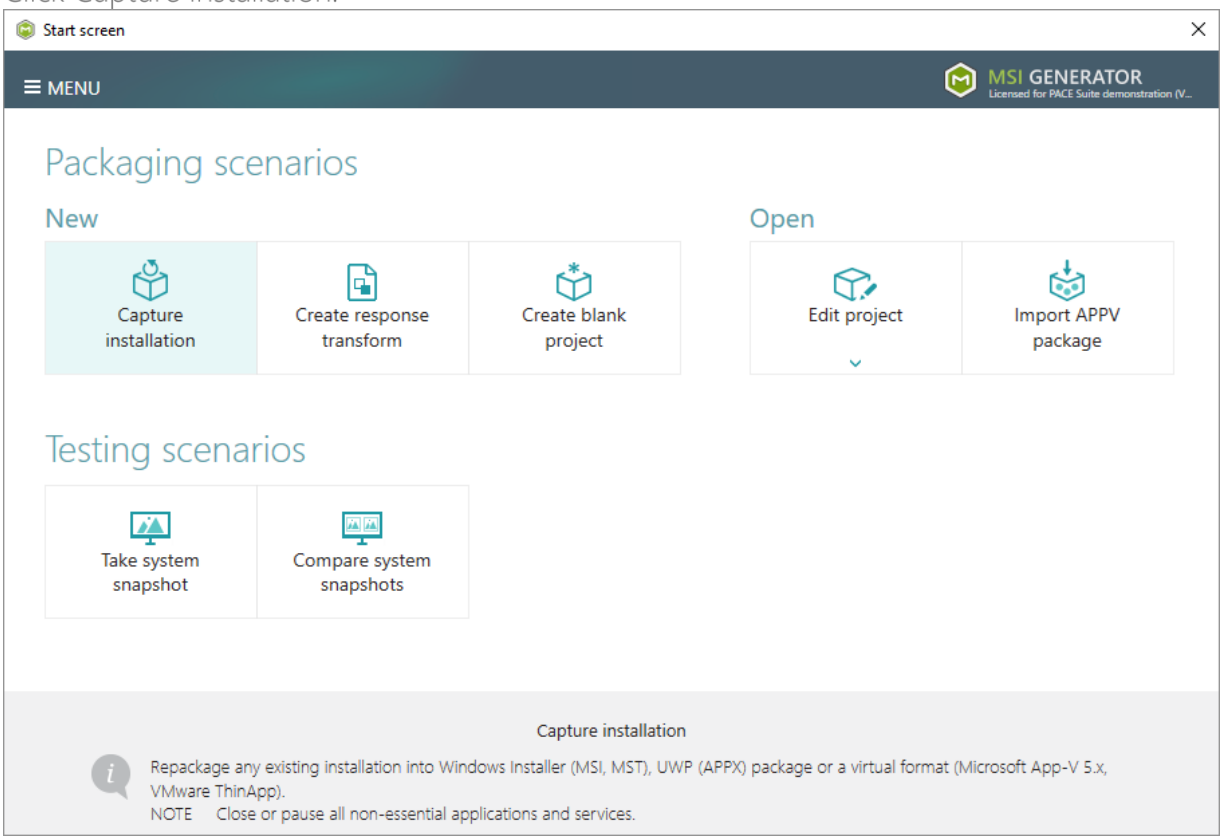
- Launch MSI Generator from the desktop or the start menu shortcut.



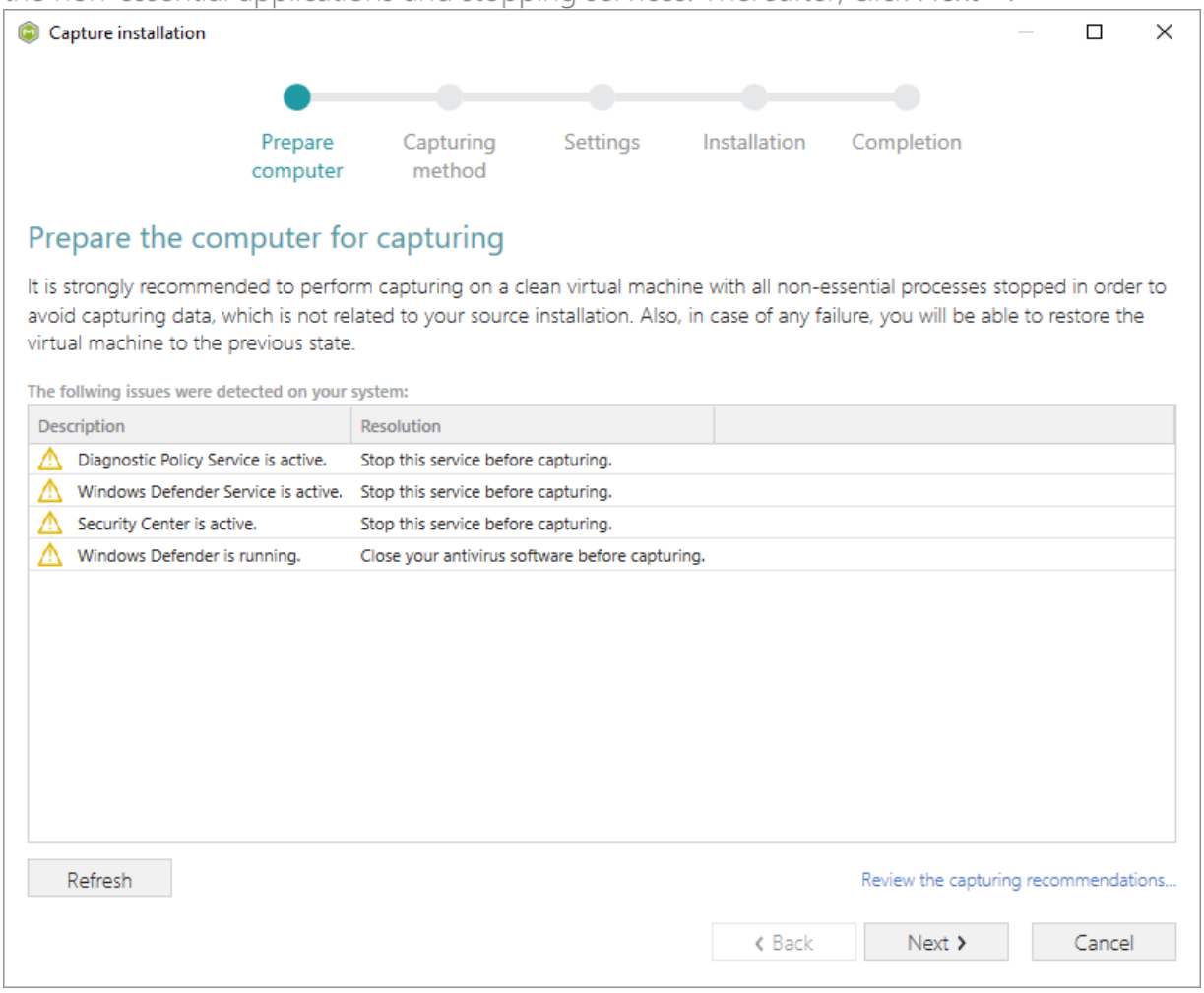

- If you have User Account Control enabled, click Yes in the opened window.



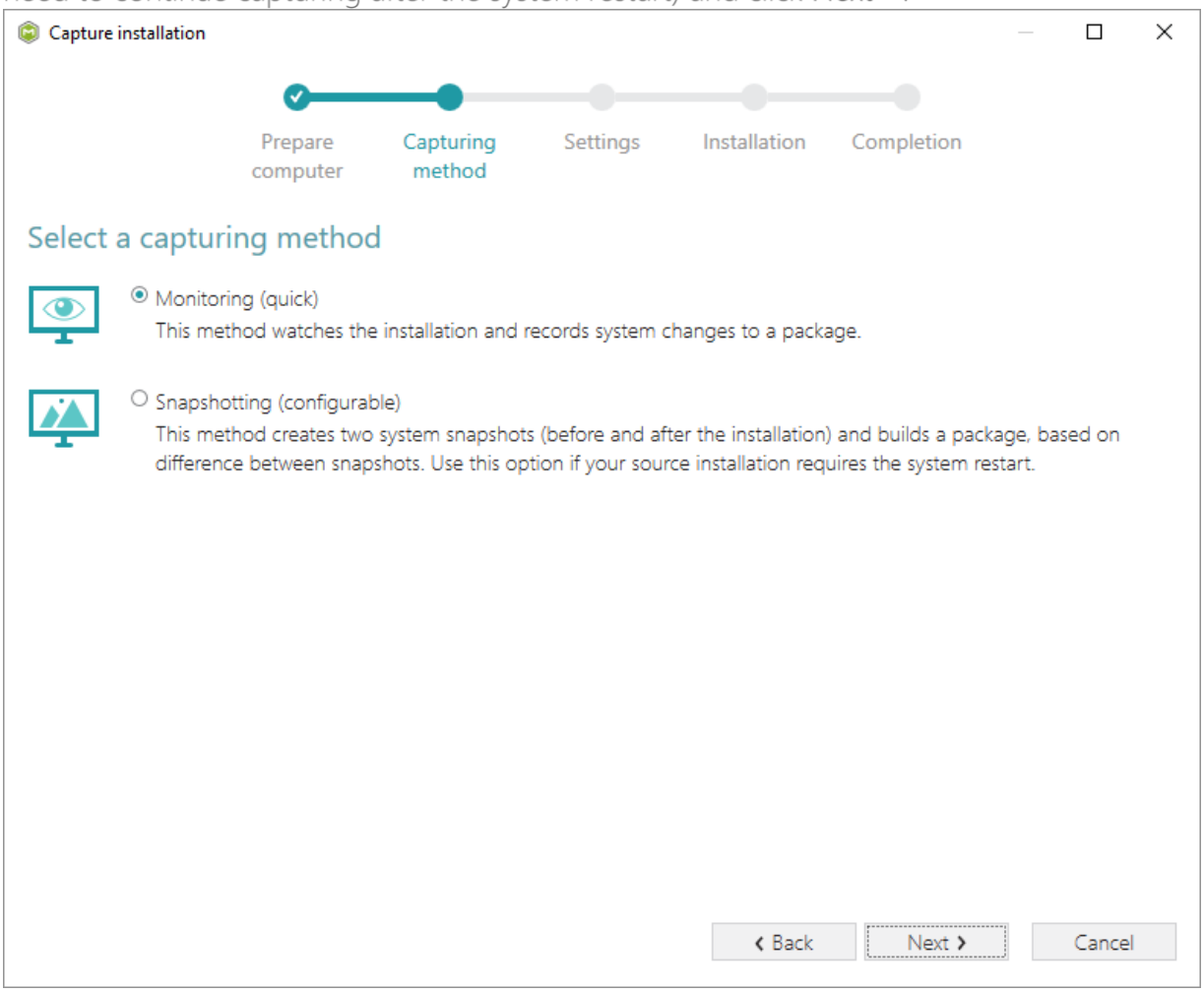
- Click Capture installation.



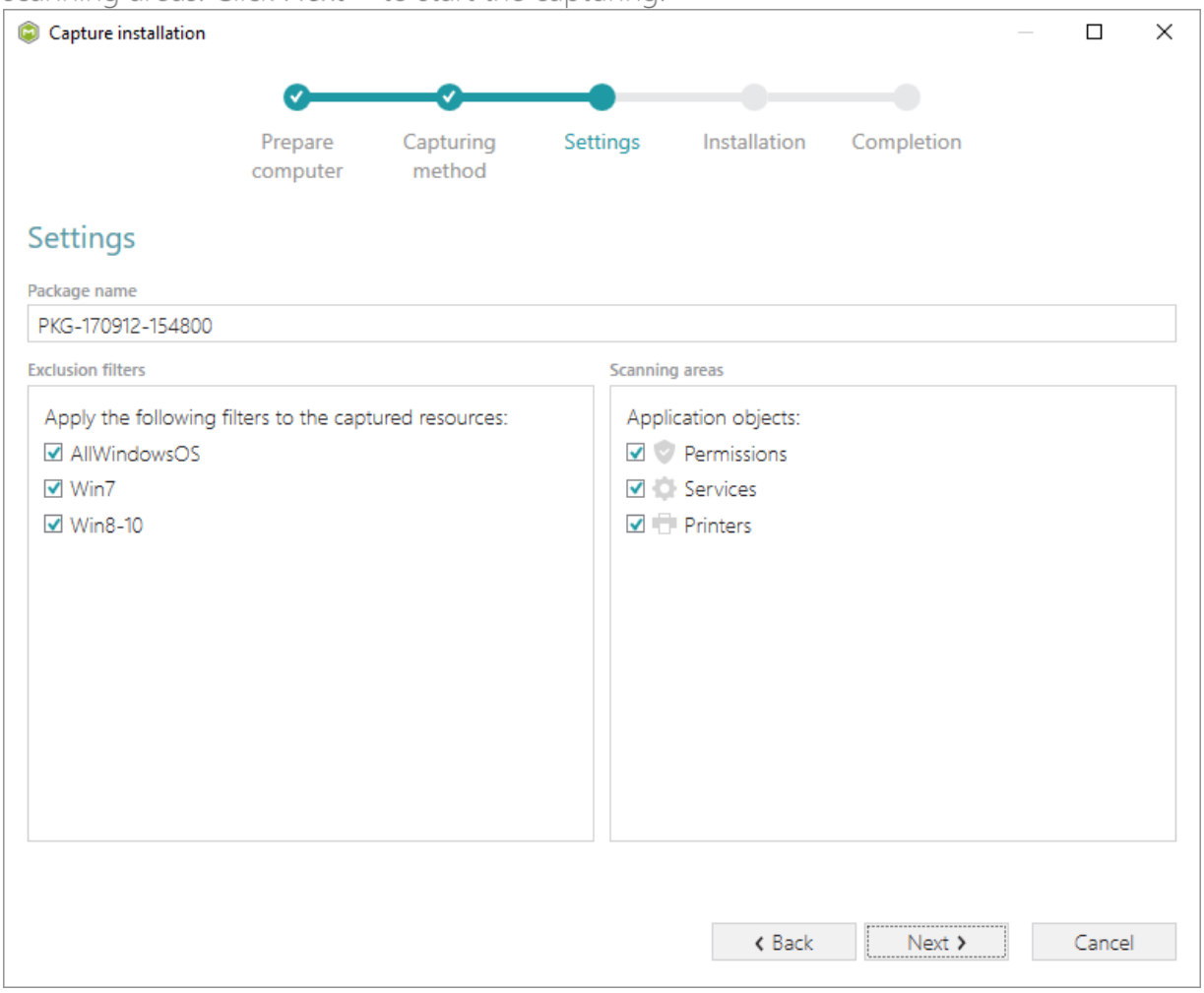
- Review the issues, which were detected on your system, and try to resolve them by closing the non-essential applications and stopping services. Thereafter, click Next >.



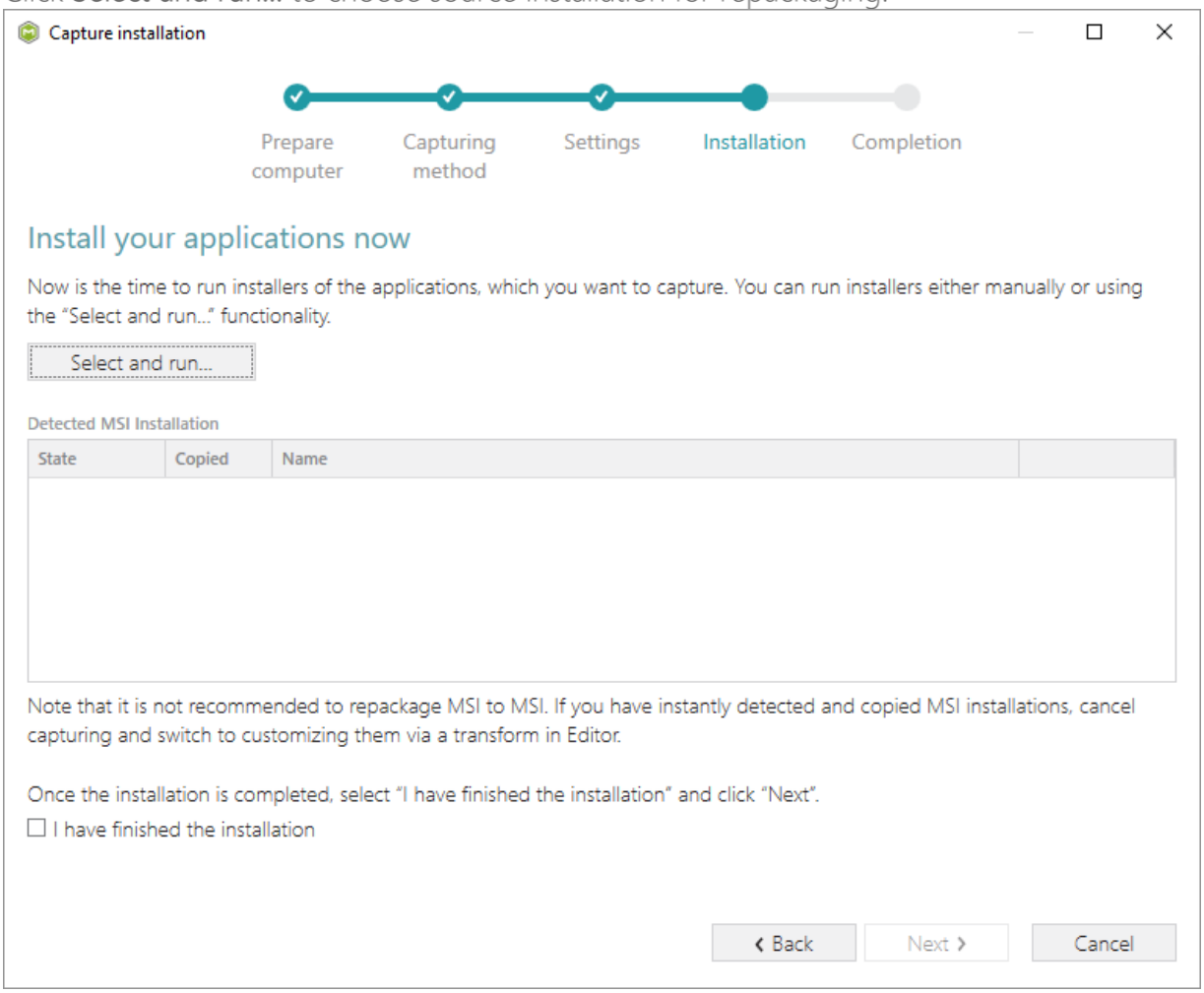
- Select the Monitoring method for the quicker capturing (or use the Snapshotting one if you need to continue capturing after the system restart) and click Next >.



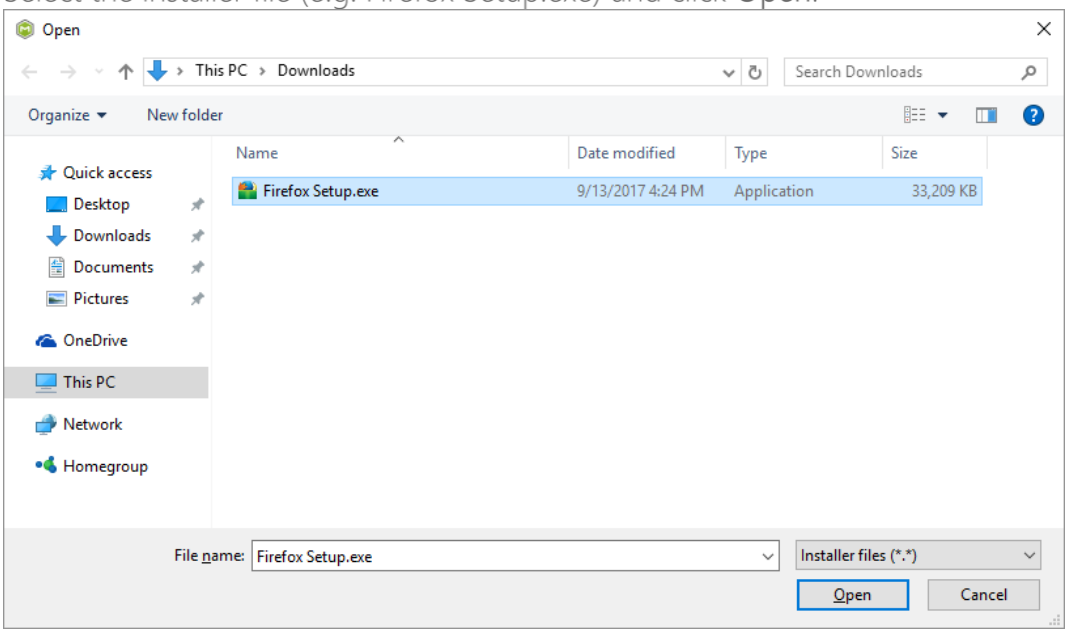
- Review and update package name. If needed, disable unnecessary exlusion filters and scanning areas. Click Next > to start the capturing.



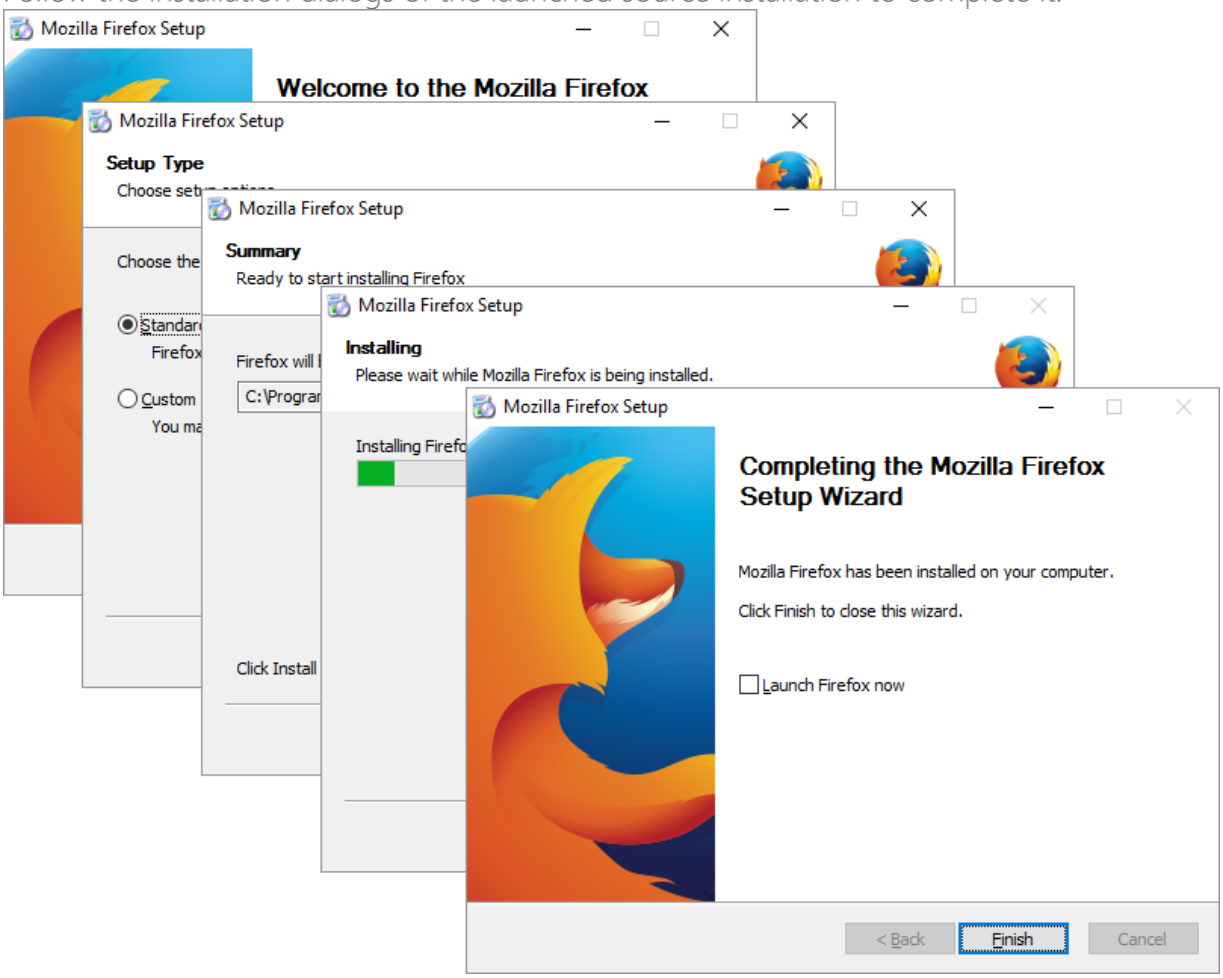
- Click Select and run... to choose source installation for repackaging.




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

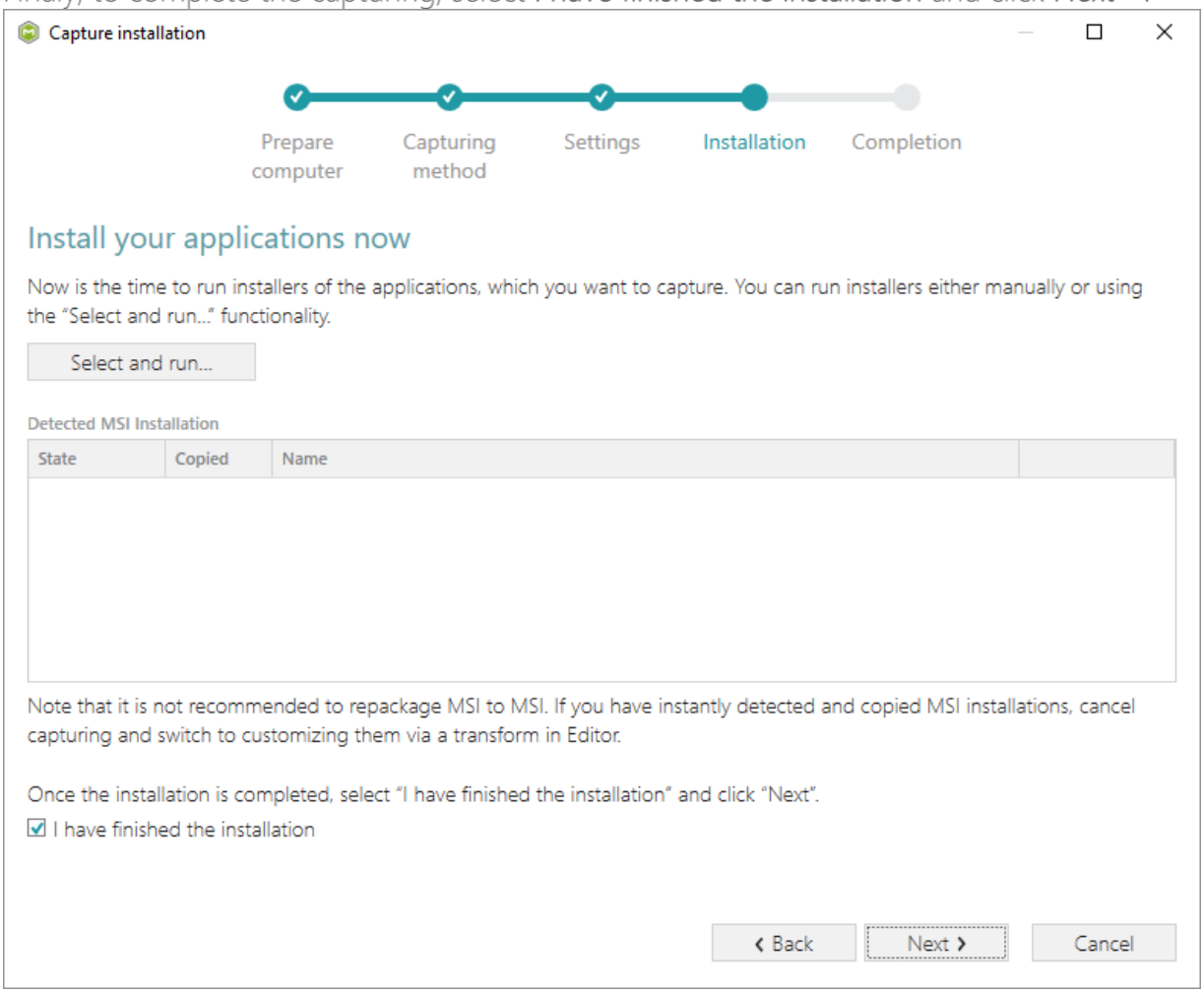


- Follow the installation dialogs of the launched source installation to complete it.

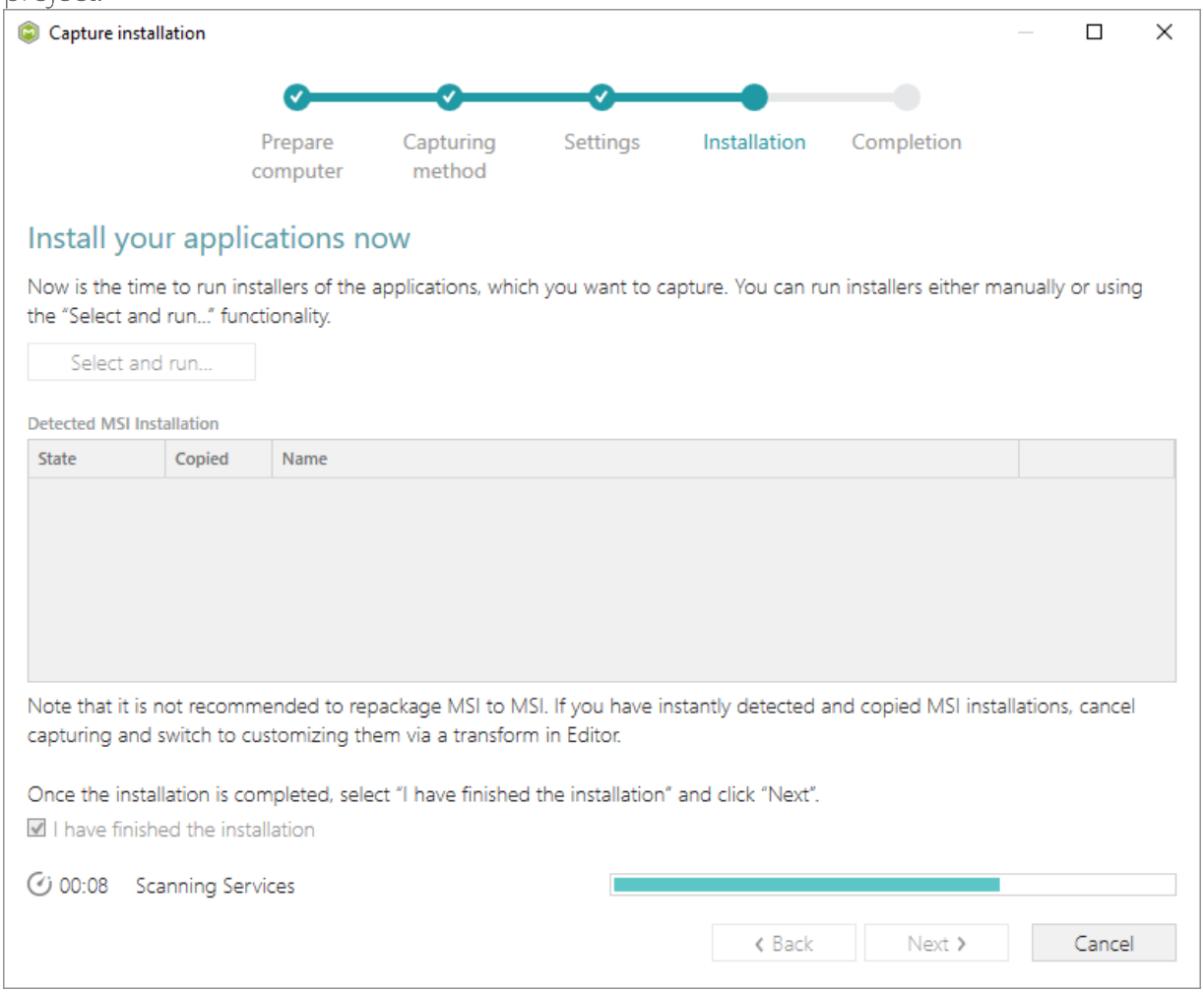



- Now you can make any changes to the file system and registry, which you want to capture and include to the package. For instance, you can create new or copy-paste existing files, import REG file to the system registry, changes permission settings, or launch the installed application in order to capture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

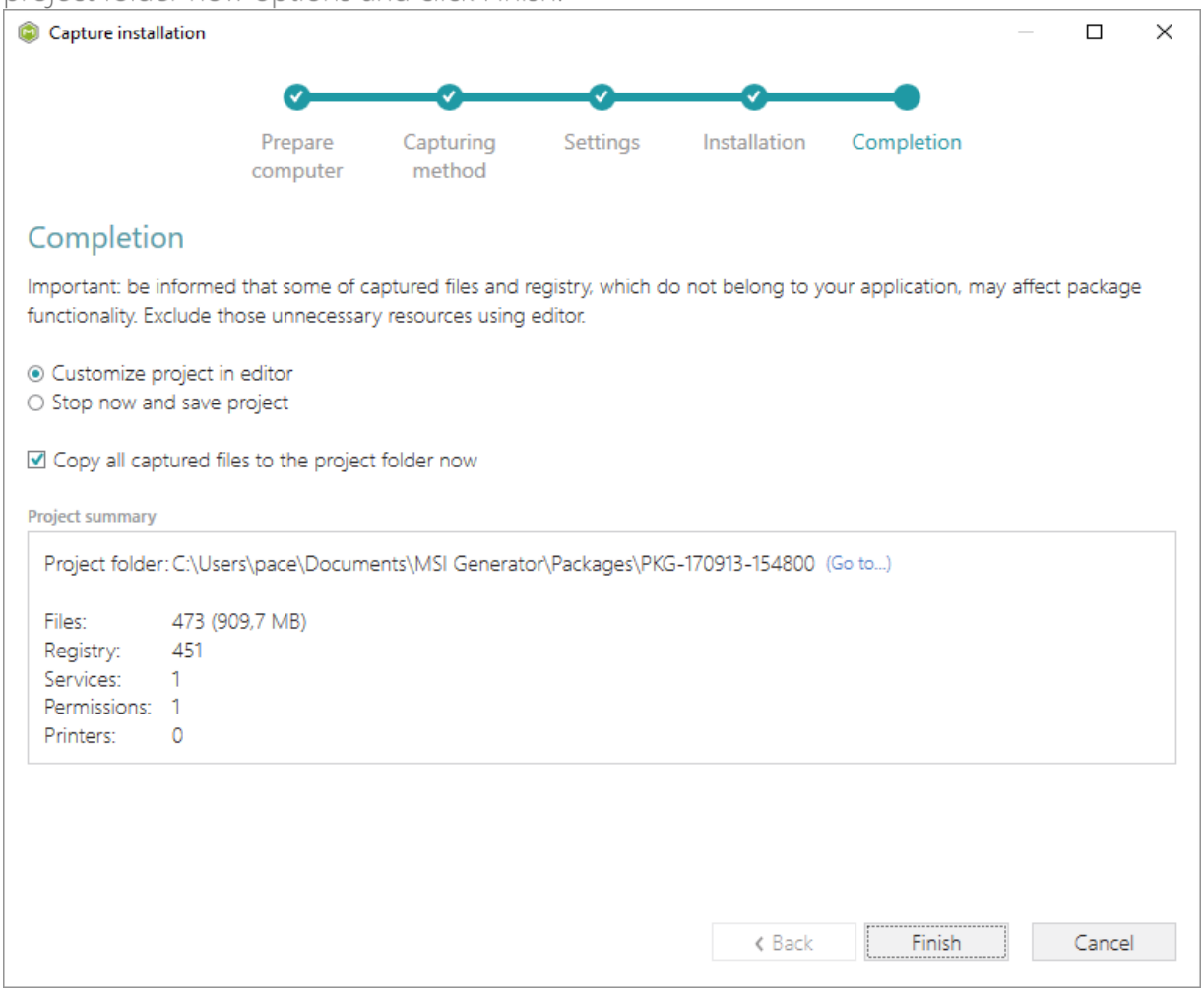
- Finaly, to complete the capturing, select I have finished the installation and click Next >.



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

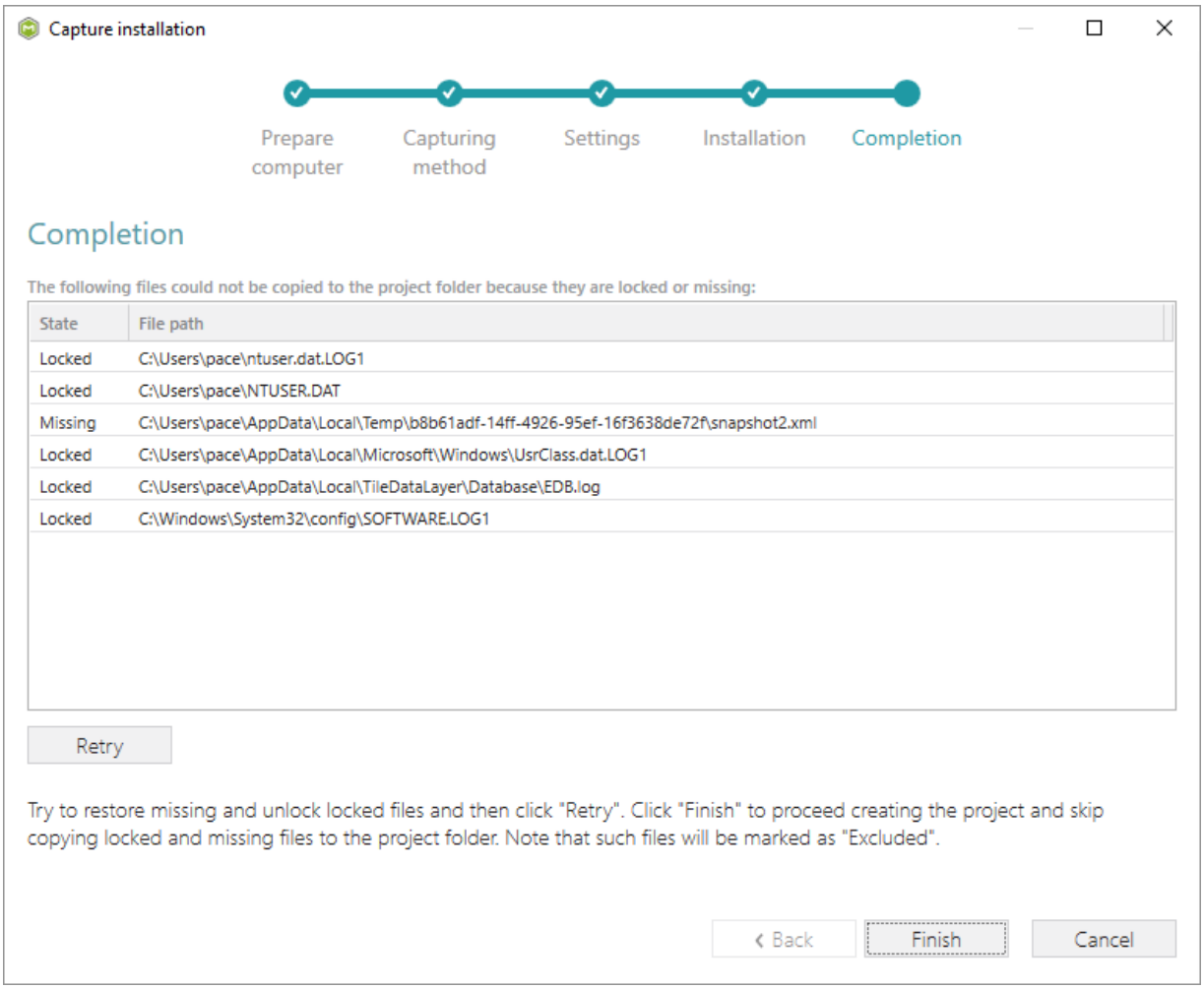


- Leave selected both the Customize project in editor and Copy all captured files to the project folder now options and click Finish.



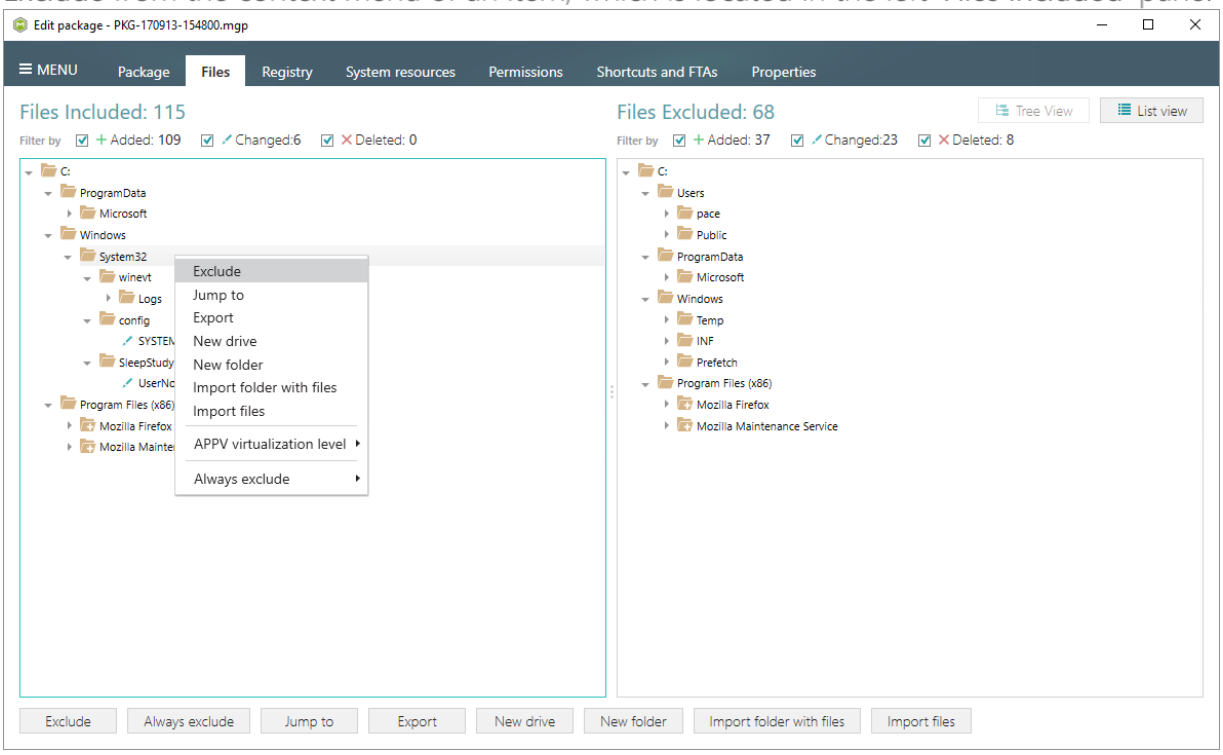

- The following dialog displays captured files, which could not be copied to the project folder because they do not exist anymore or locked by the system or by an application. Try to resolve these issues and then click Retry. Click Finish to skip copying the missing and locked

files and mark them as Excluded resources.

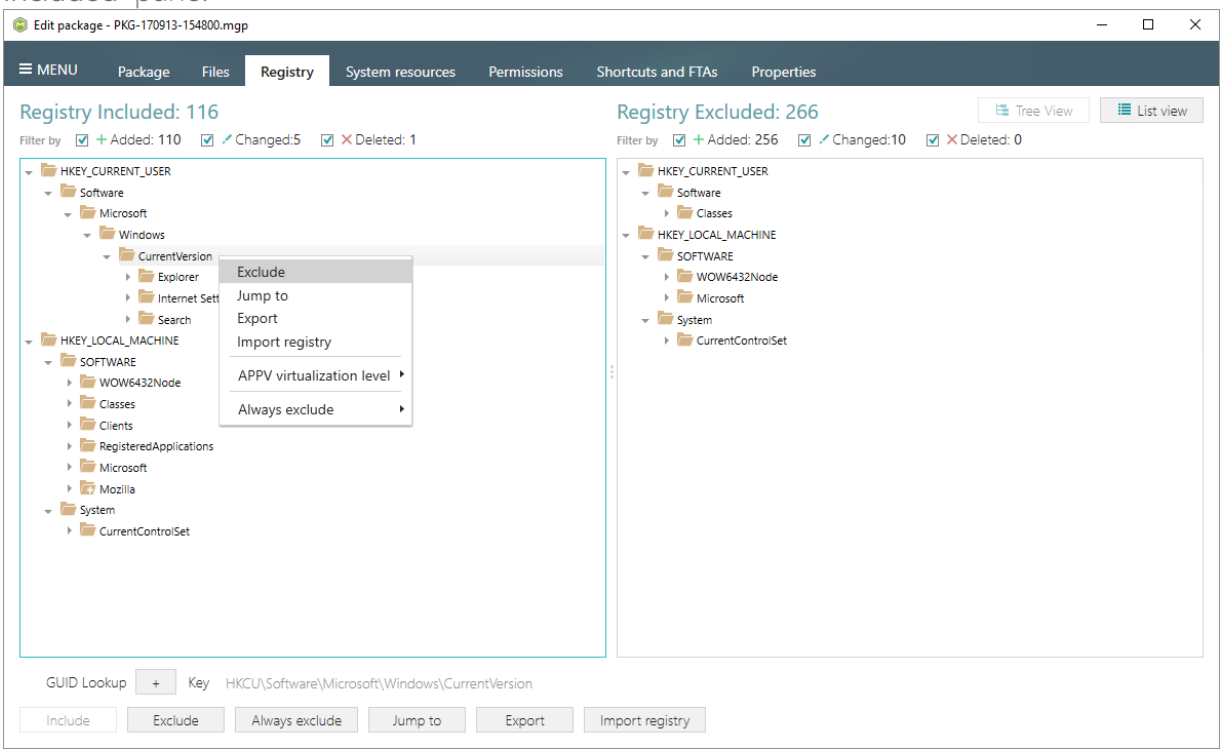



- Once the project is opened in the project editor, review the captured resources at the Files, Registry, System resources, Permissions, and Shortcuts and FTAs tabs and exclude unnecessary ones from the project. Unnecessary resources are files, registry entries, which are usually created or modified in the result of operating system work, and such resources could not be a part of your captured application. Unfortunately, there is no universal rule to discover which of captured files or registry entries should be excluded, so exclude only those ones, which almost 100% do not refer to your captured application (e.g. NOD32 antivirus files couldn't be a part of Firefox application).

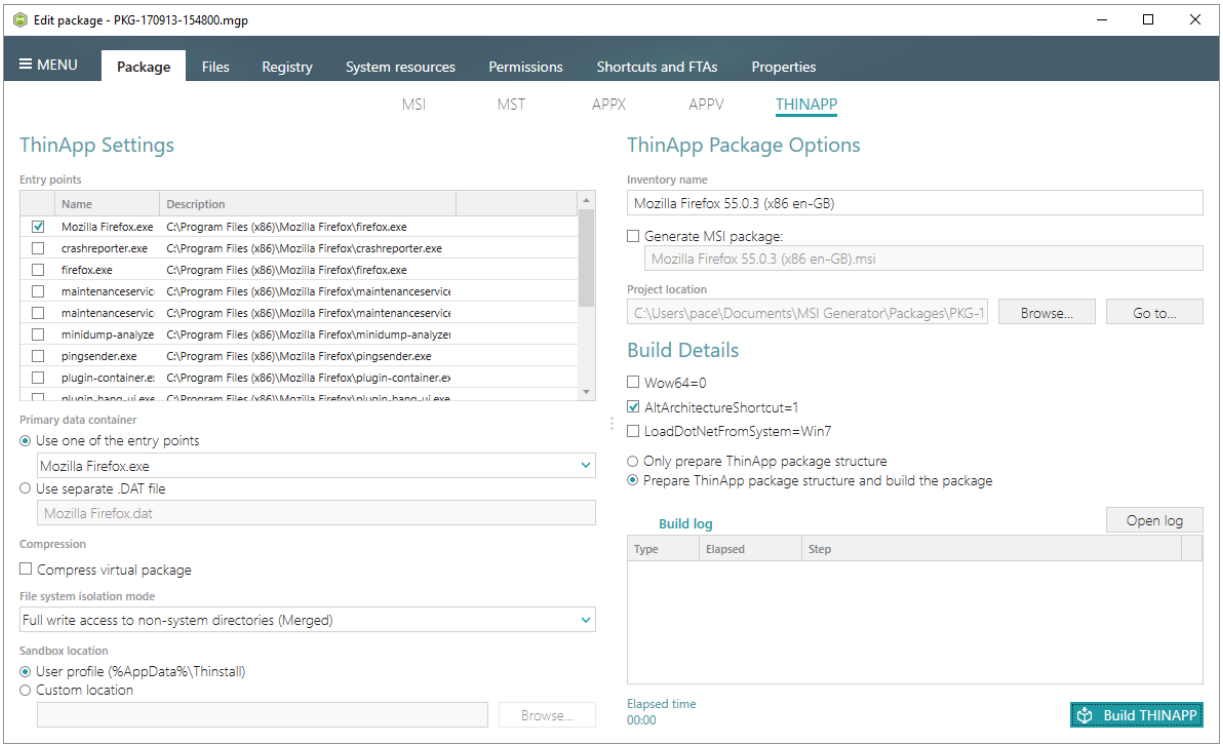
- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.




- 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.

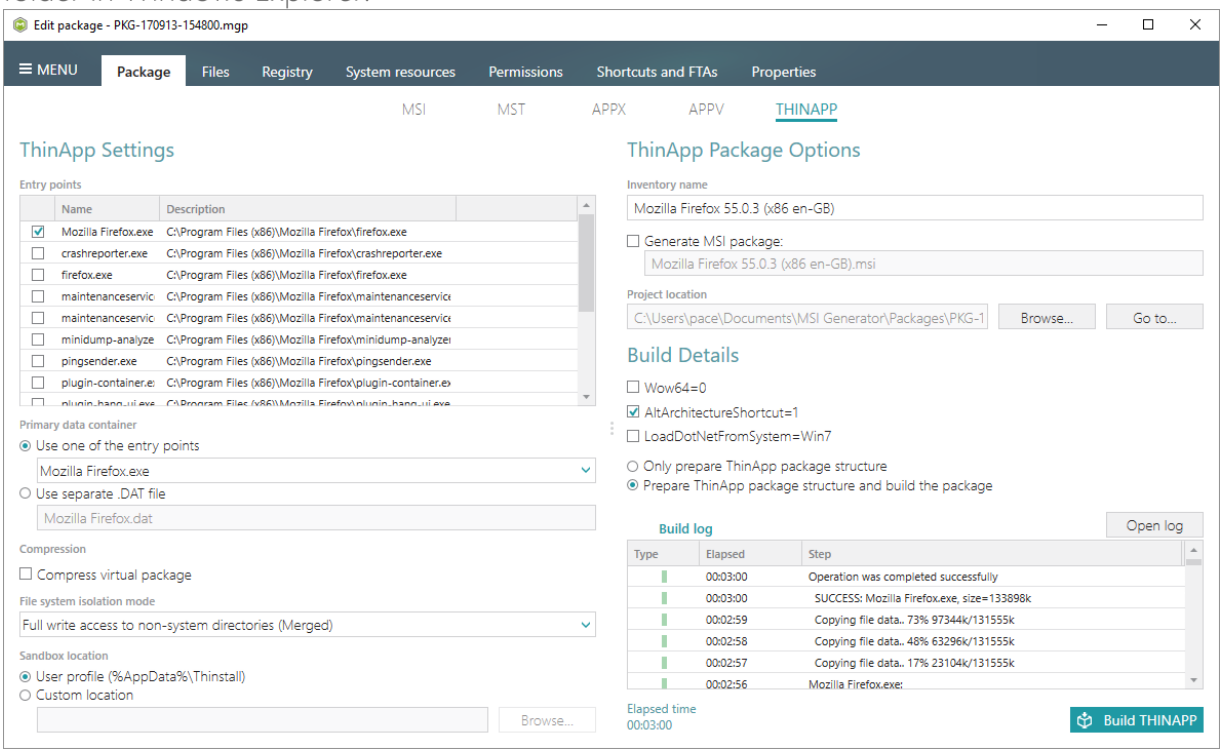



- Finaly, to build THINAPP package from the project, navigate to the **Package** -> **THINAPP** tab, review **Entry points** (shortcuts), primary data container location and other package options, and then click **Build THINAPP**.




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. If you have installed this tool to the custom location, go to the MENU -> Settings, select the Custom ThinApp install location option and then click Browse... to choose the installation location of the VMware ThinApp tools

- Click Go to..., located next to the Project location field, to open the package containing folder in Windows Explorer.



## 3.7 Edit MSI/MST

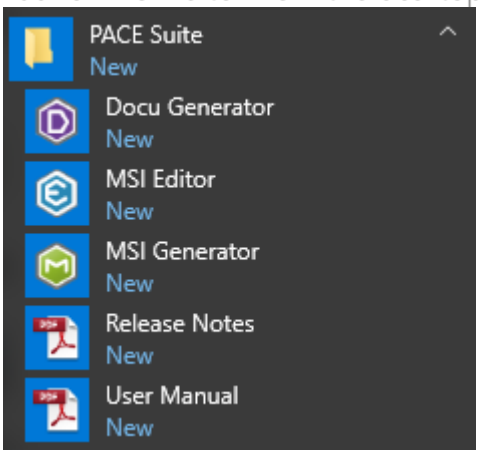
Choose a scenario that better suits your needs:

- Manage Files and Folders, described in section 3.7.2
  Review files and folders of the package, add new system and user-defined folders (named subfolders), set permission settings, import files, extract COM information and .NET assemblies from the imprted files to register them. Move files and folders to another folders, set the INSTALLDIR folder, and edit file and folder properties.
- Manage Shortcuts, described in section 3.7.3
  Review shortcuts of the package, add new shortcuts to files, folders, URLs or import them from the file system. Move shortcuts to another folder, manage shortcut details.
- Manage Registry, described in section 3.7.4
  Review registry of the package, add new registry keys and values, import registry from the REG file, and export registry to the REG file. Integrate ActiveSetup, manage registry key permissions, move keys and values to another keys, edit key and value details.
- Manage Custom Actions, described in section 3.7.5
  Review custom and standard actions of the package, edit and add new actions to any sequences of all installation modes. Run PowerShell scripts and so on.
- Manage Device Drivers, described in section 3.7.6
  Review device drivers of the package, edit and add new drivers.
- Manage Features and Components, described in section 3.7.7
  Review the feature-component structure of the package, component resources, add new features and components, manage their properties.
- Manage Properties, described in section 3.7.8
  Review properties of the package, their initial values and values, which will be defined in runtime via the CustomAction, AppSearch, Control, ComboBox, CheckBox, RadioButton tables. Add new and modify existing properties.
- Edit Summary Information, described in section 3.7.9
  Edit the Summary Information properties of the opened MSI package or applied MST file.
- Import Upgrades, described in section 3.7.10
  Integrate the automatic upgrade of packages of the previous versions.
- Add and Drop Tables, described in section 3.7.11
  Add standard, import standard and custom tables with or without content to the opened package. Drop tables from the package controling which of linked resources also will be removed
- Import Merge Modules, described in section 3.7.12
  Import Merge Modules (MSM) to the package to deliver the shared resources properly.

### 3.7.1 Pre-Condition

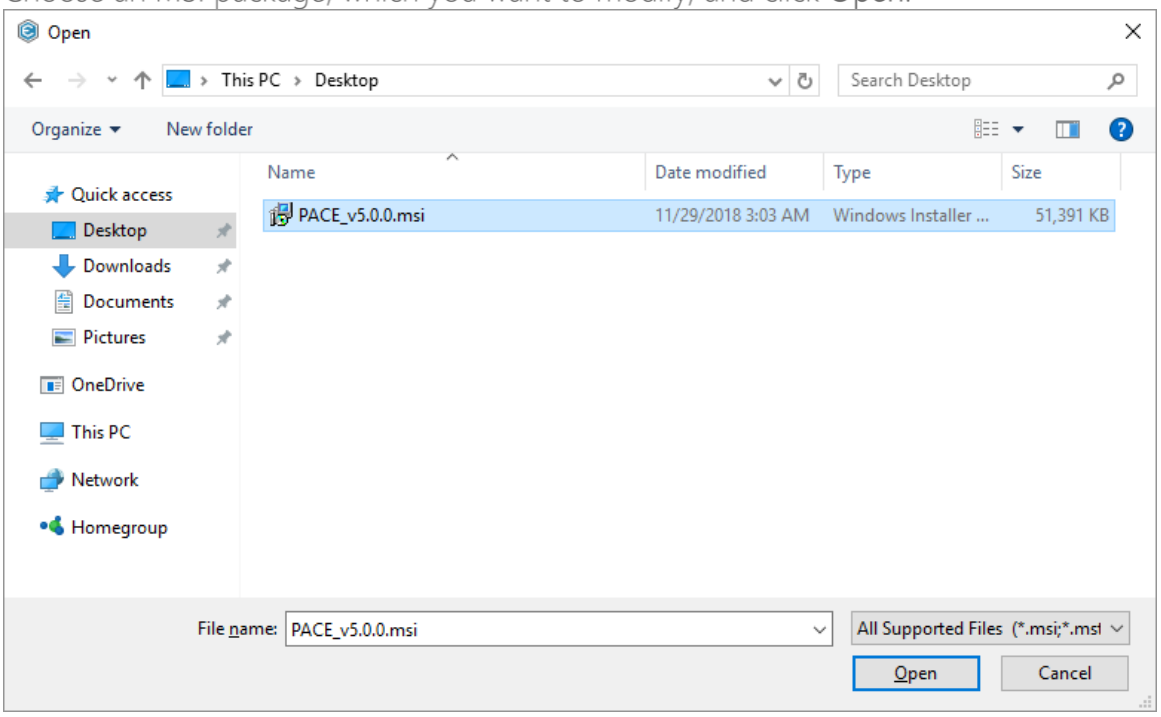
Open an MSI package following the instructions below:

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

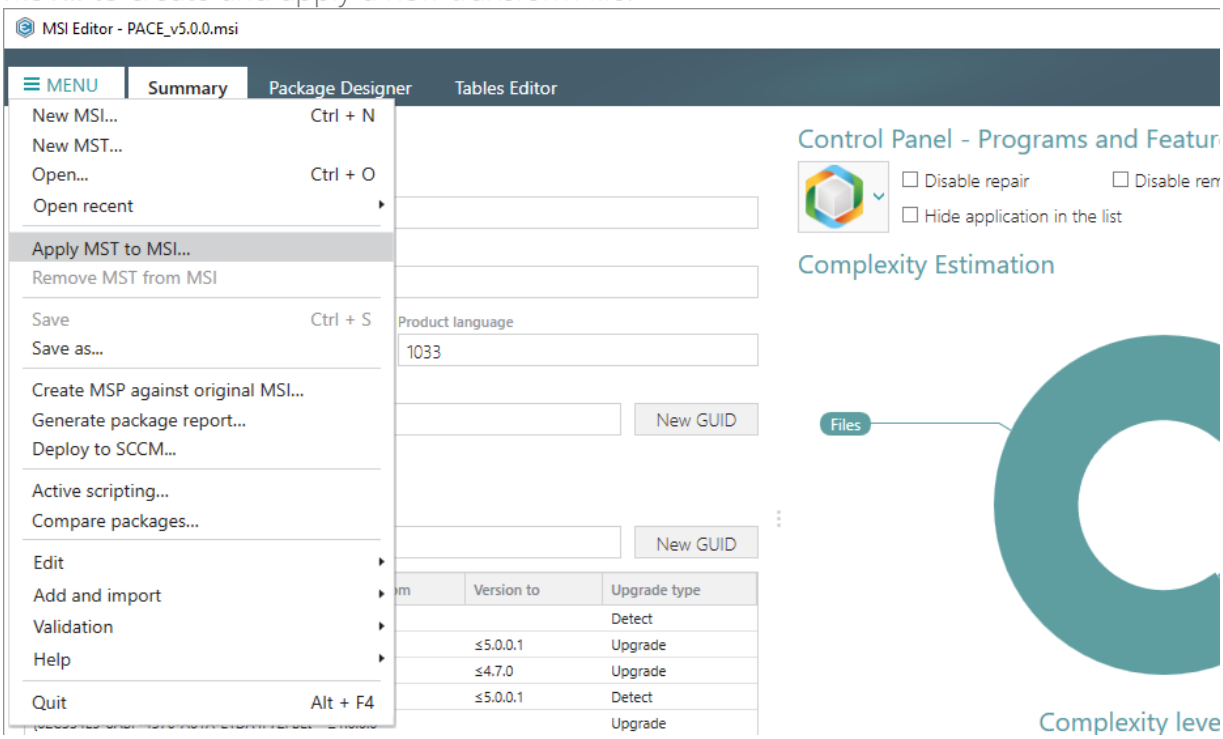



- Select Open... from MENU.

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




- If you want to modify MSI package via Transform (MST), you can either apply the existing MST by selecting **Apply MST to MSI...** from **MENU** and choosing an MST file or select **New MST...** to create and apply a new transform file.



### 3.7.2 Files and Folders

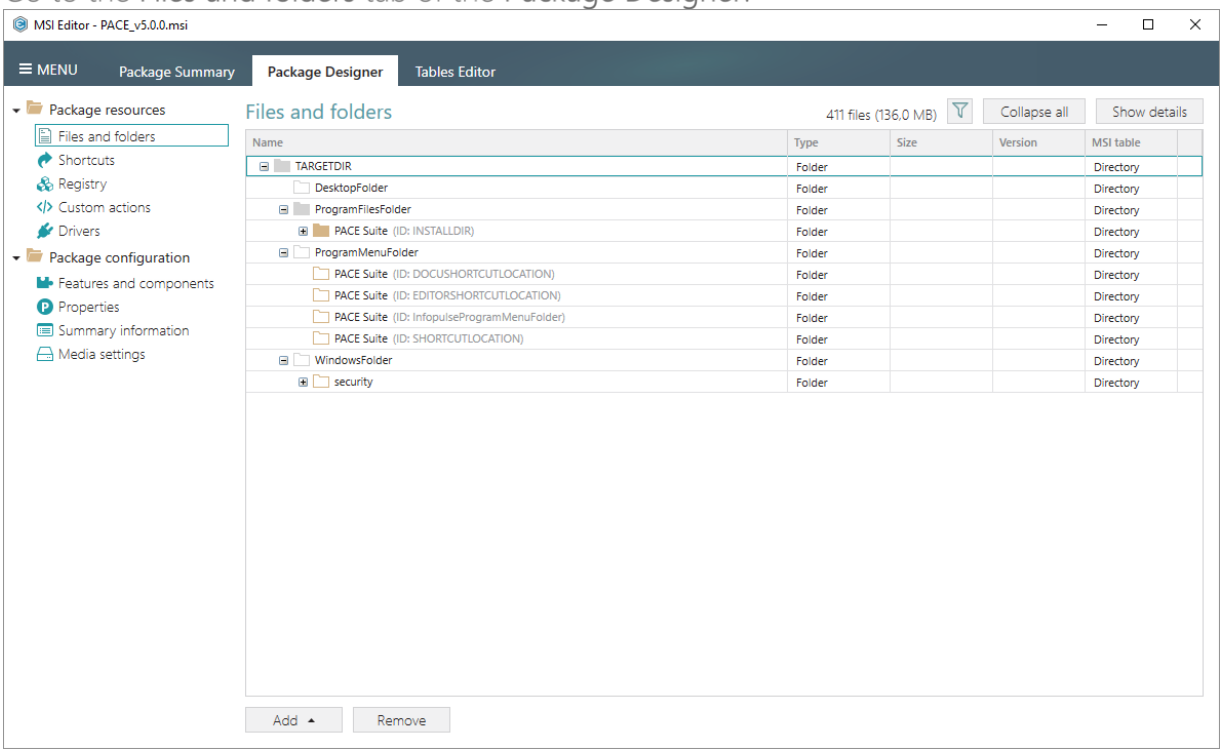
Choose a scenario that better suits your needs:

- Add System Folder, described in section 3.7.2.1
  Add a system (standard) folder from the list into the package.
- Add Subfolder, described in section 3.7.2.2
  Add a subfolder (user-defined folder) to the package.
- Import Files and COM, described in section 3.7.2.3
  Import files with the extracted on-the-fly COM information.
- Change Compression Type, described in section 3.7.2.4
  Rebuild the whole package with the selected compression type.
- Set Permissions, described in section 3.7.2.5
  Set custom permissions for the file or the folder.
- Remove Permissions, described in section 3.7.2.6
  Locate files and folders with custom permissions and remove the unnecessary permissions.
- Move Files, Folders, described in section 3.7.2.7
  Move a file or a folder from one folder to another.
- Set INSTALLDIR, described in section 3.7.2.8
  Set a subfolder (user-defined folder) of the package as the INSTALLDIR.
- Edit INI File, described in section 3.7.2.9
  Edit a name, a destination location and a content of the INI file, which has been read from the IniFile table.
- Edit File Properties, described in section 3.7.2.10
  Edit a file name, an identifier, a destination location, a component and many other attributes of the binary file, which has been read from the File table.
- Edit Folder Properties, described in section 3.7.2.11
  Edit a folder target and source name, an identifier, a parent folder.

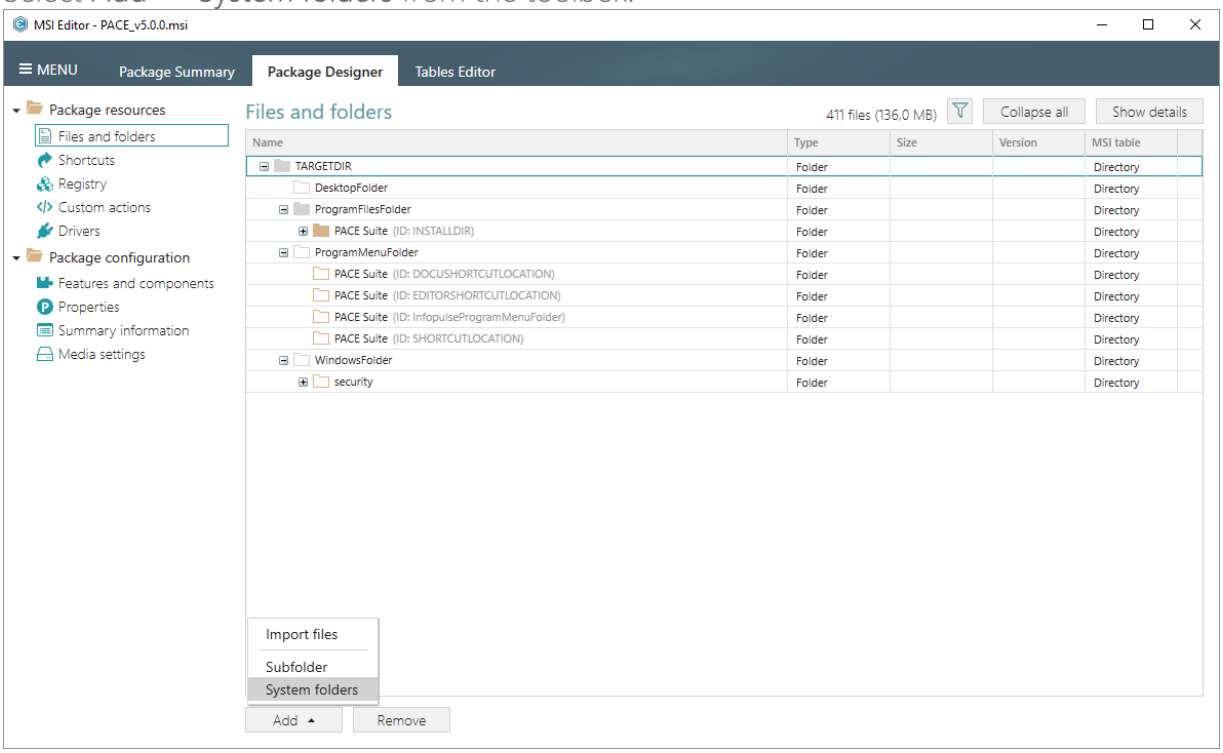
### 3.7.2.1 Add System Folder

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

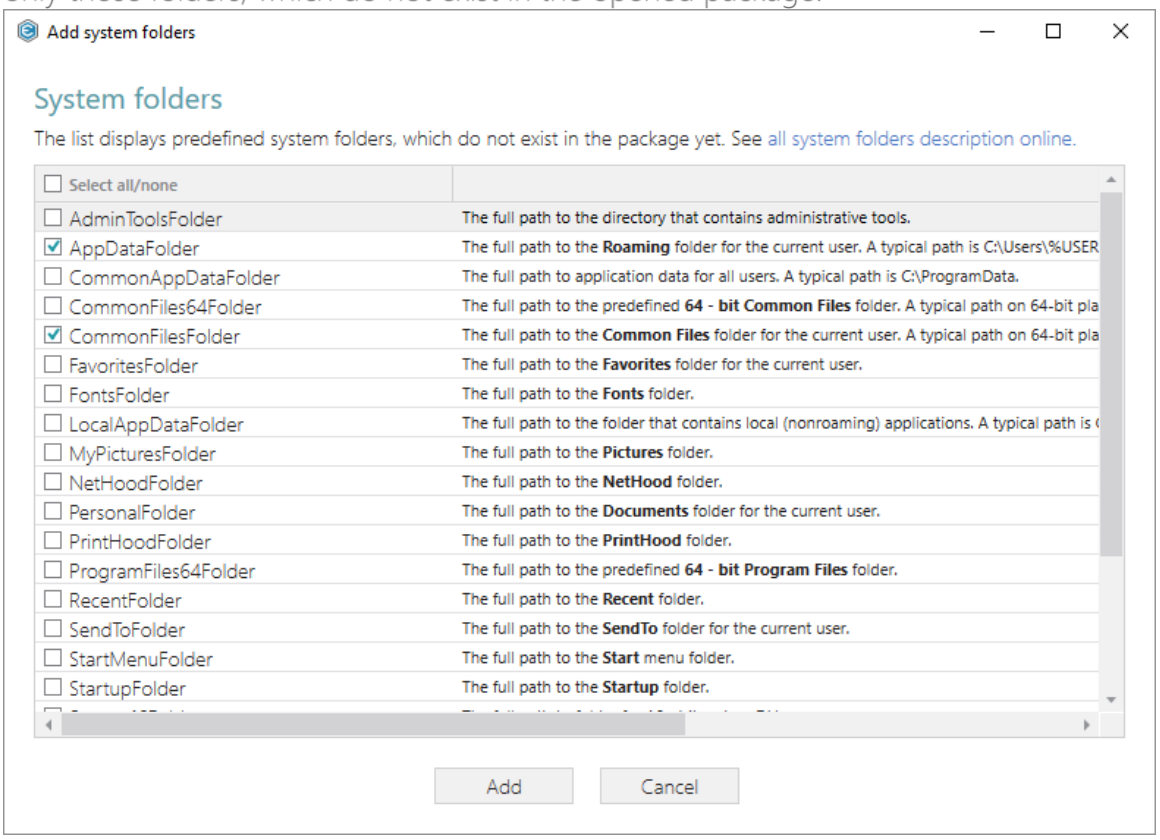
- Go to the Files and folders tab of the Package Designer.



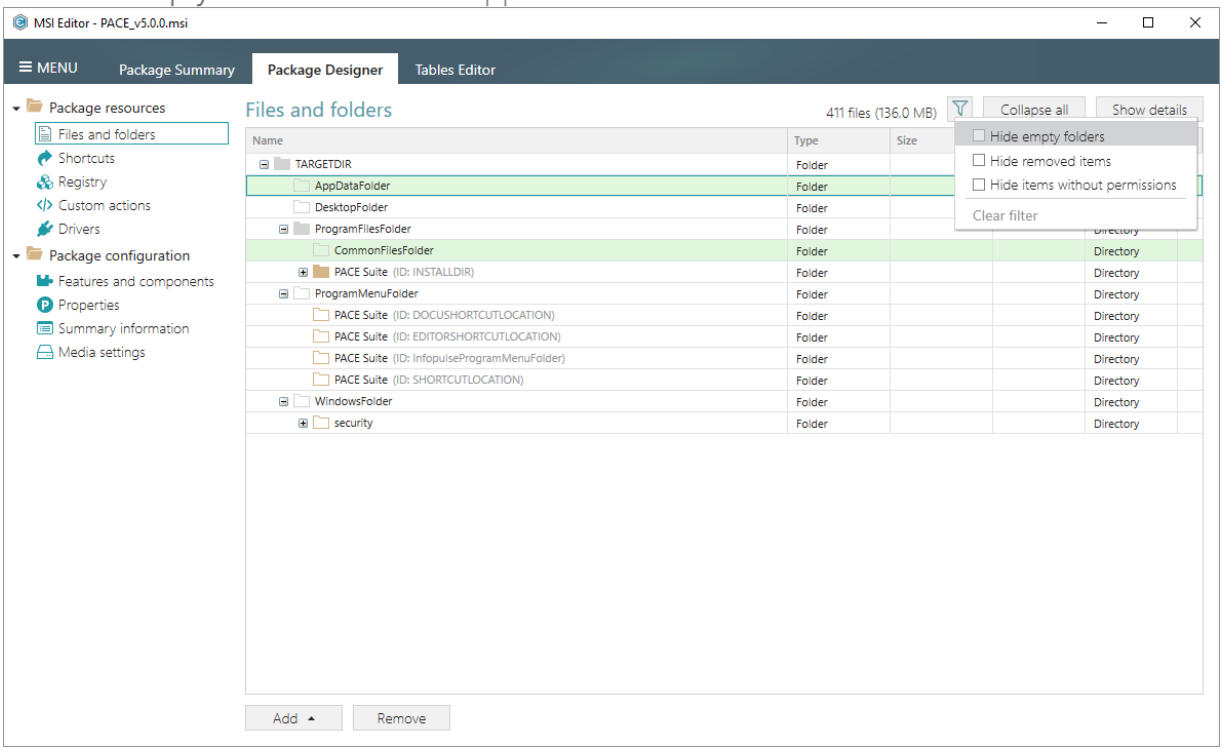

- Select Add -> System folders from the toolbox.



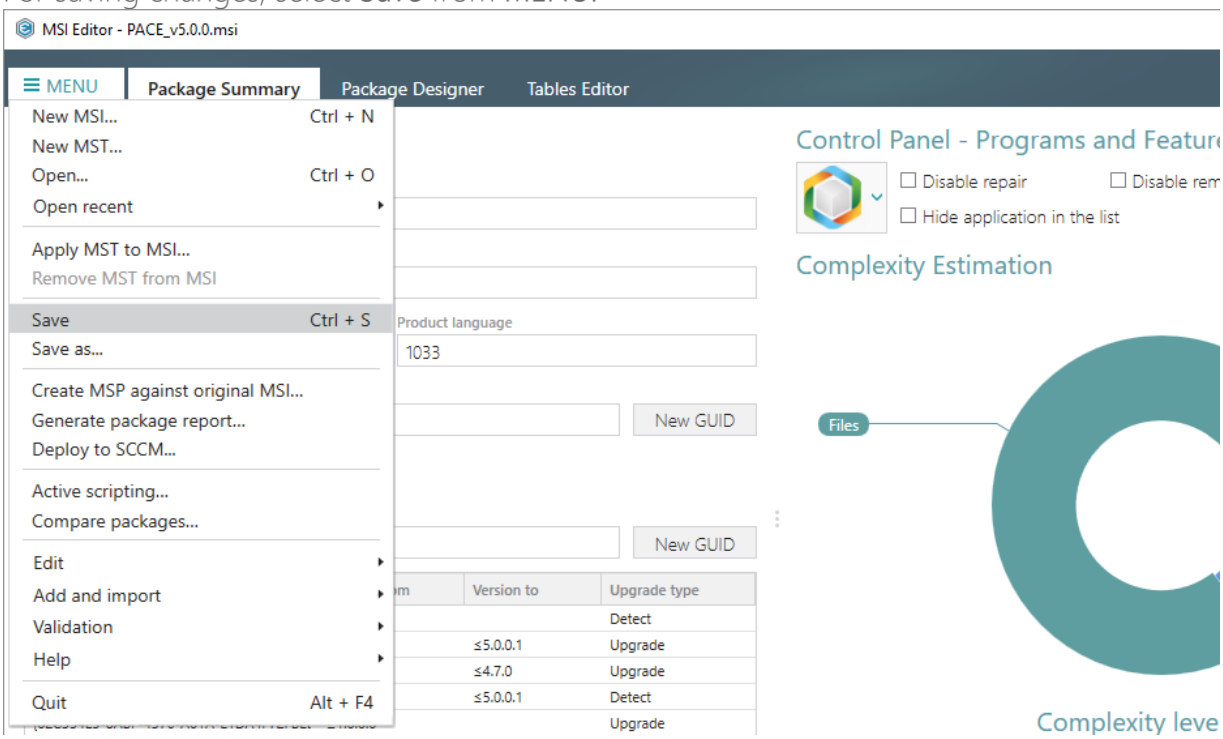
- Choose the necessary system folders from the list and click **Add**. Note that this list displays only those folders, which do not exist in the opened package.




- If you cannot find the specific folder in the list and on the **Files and folders** tab, ensure that the **Hide empty folders** filter is not applied.



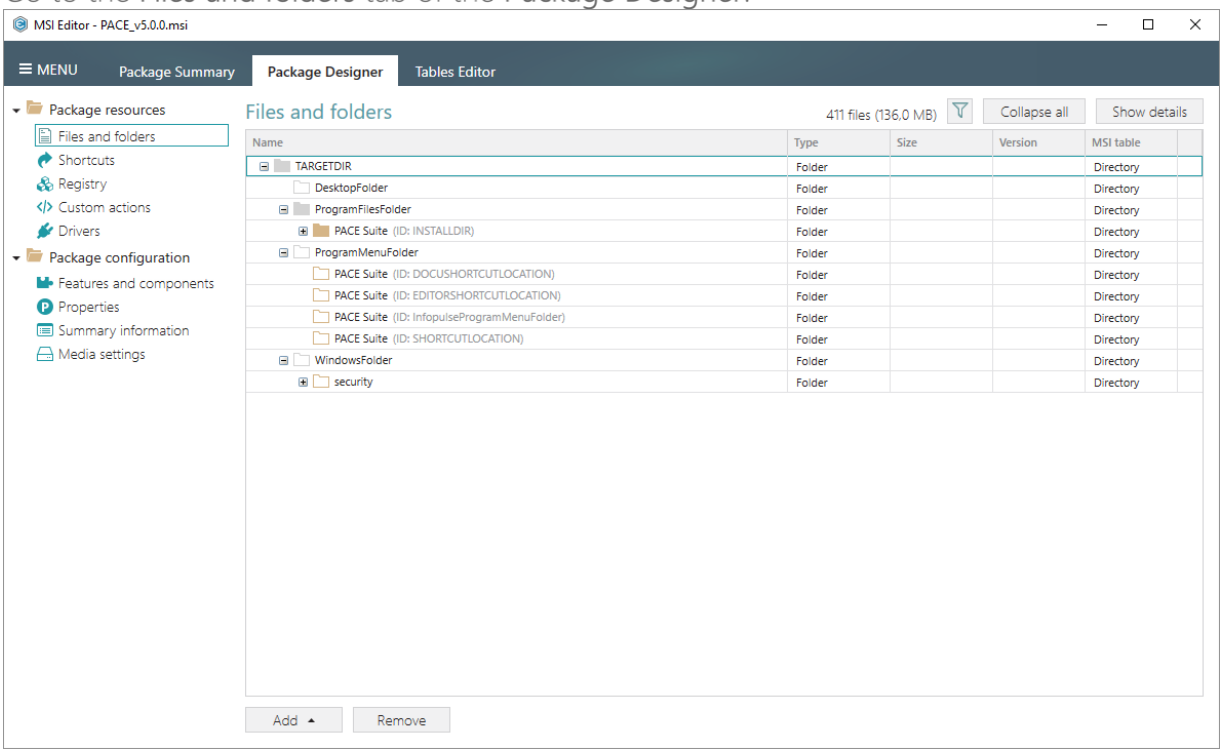
- For saving changes, select Save from MENU.




### 3.7.2.2 Add Subfolder

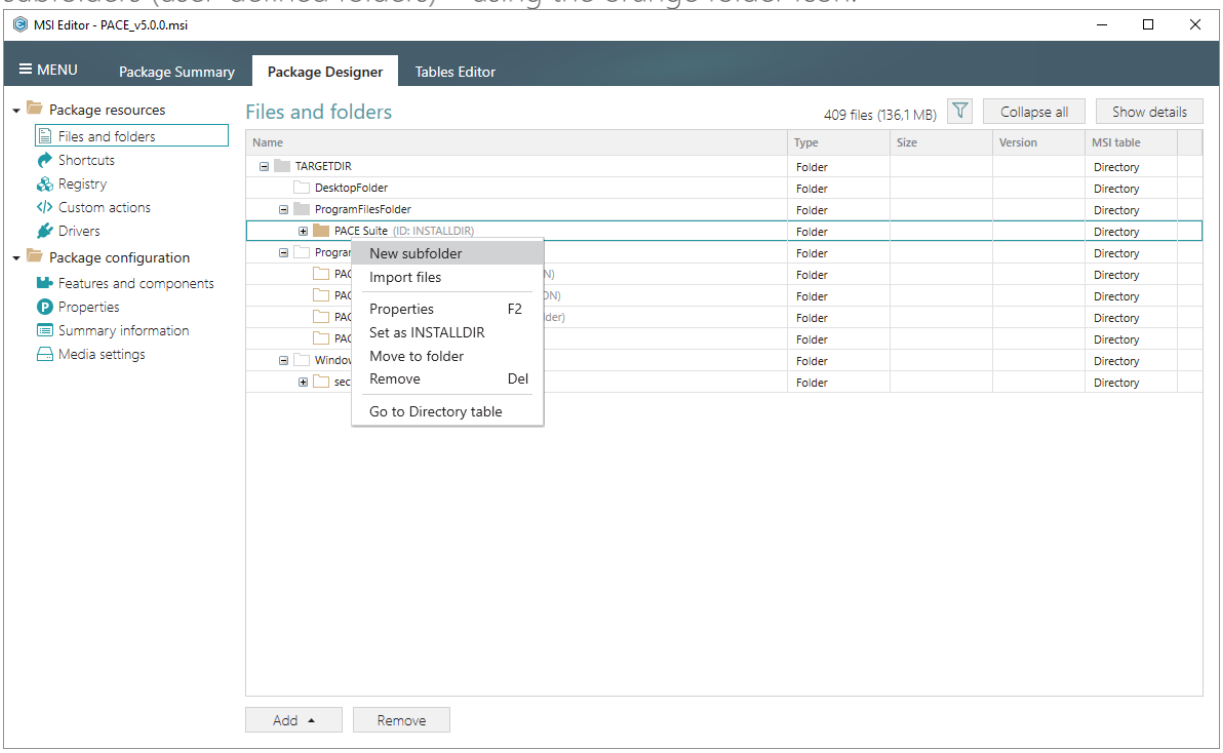
Add a subfolder (user-defined folder) to the package.

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

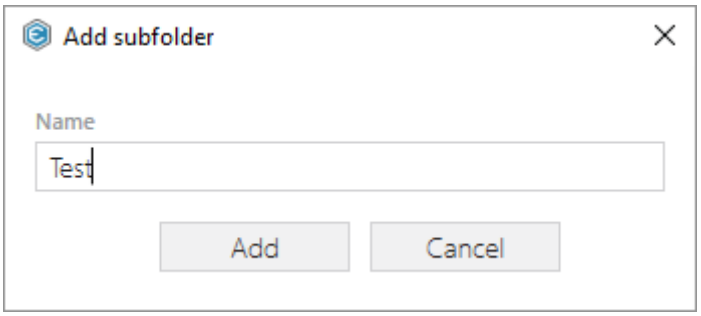



- Select New subfolder from the context menu of a folder, where you want to create a new subfolder. Note that system folders are displayed using the grey folder icon and the

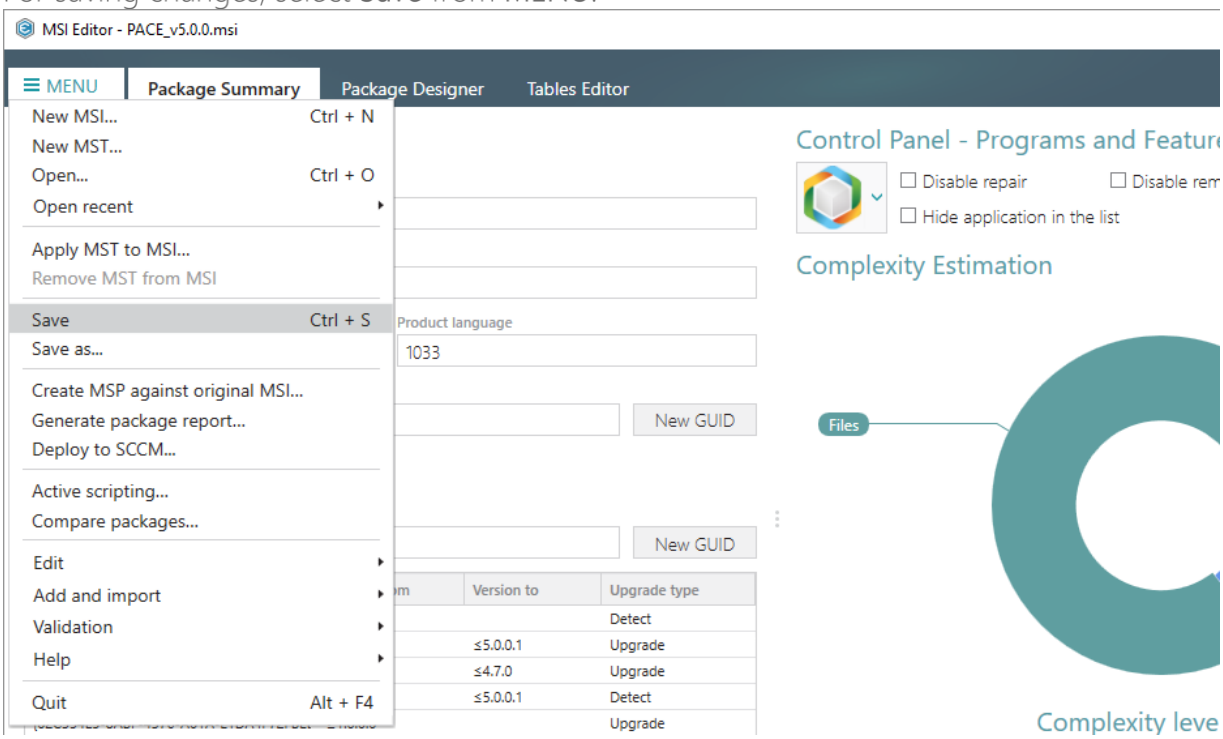
subfolders (user-defined folders) – using the orange folder icon.




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



- For saving changes, select Save from MENU.

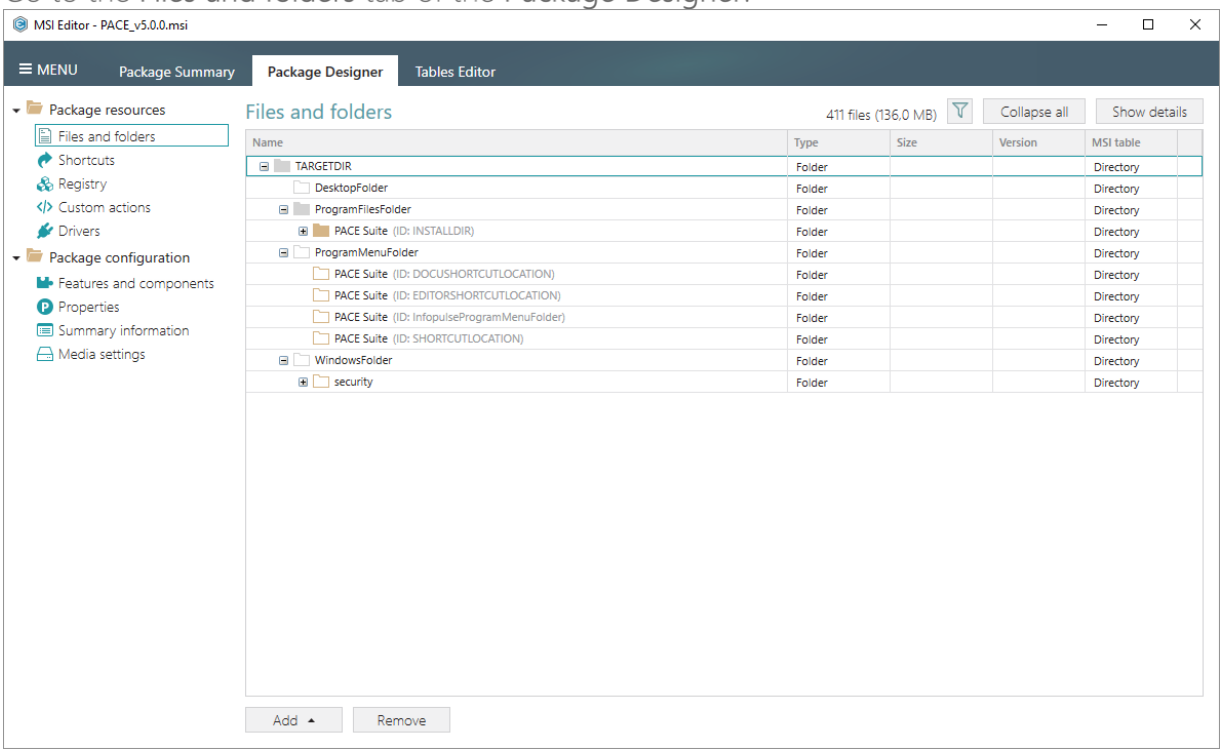



### 3.7.2.3 Import Files and COM

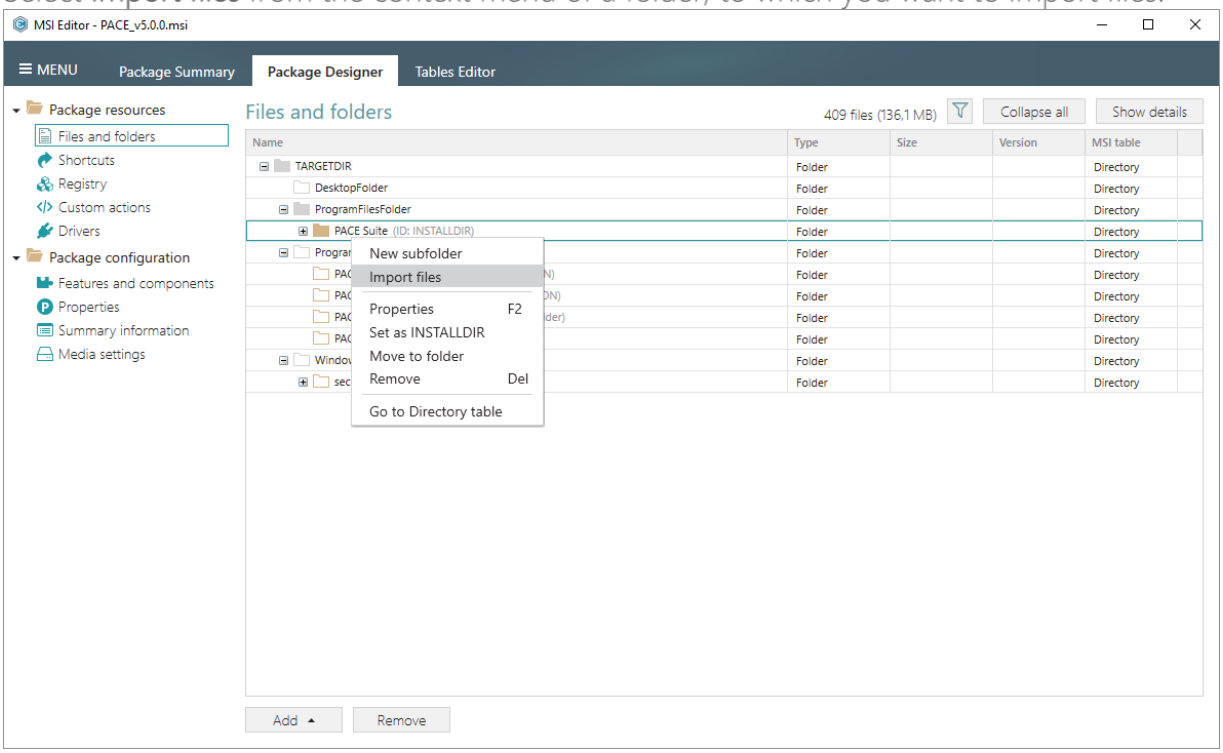
Import files with the extracted on-the-fly COM information.

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

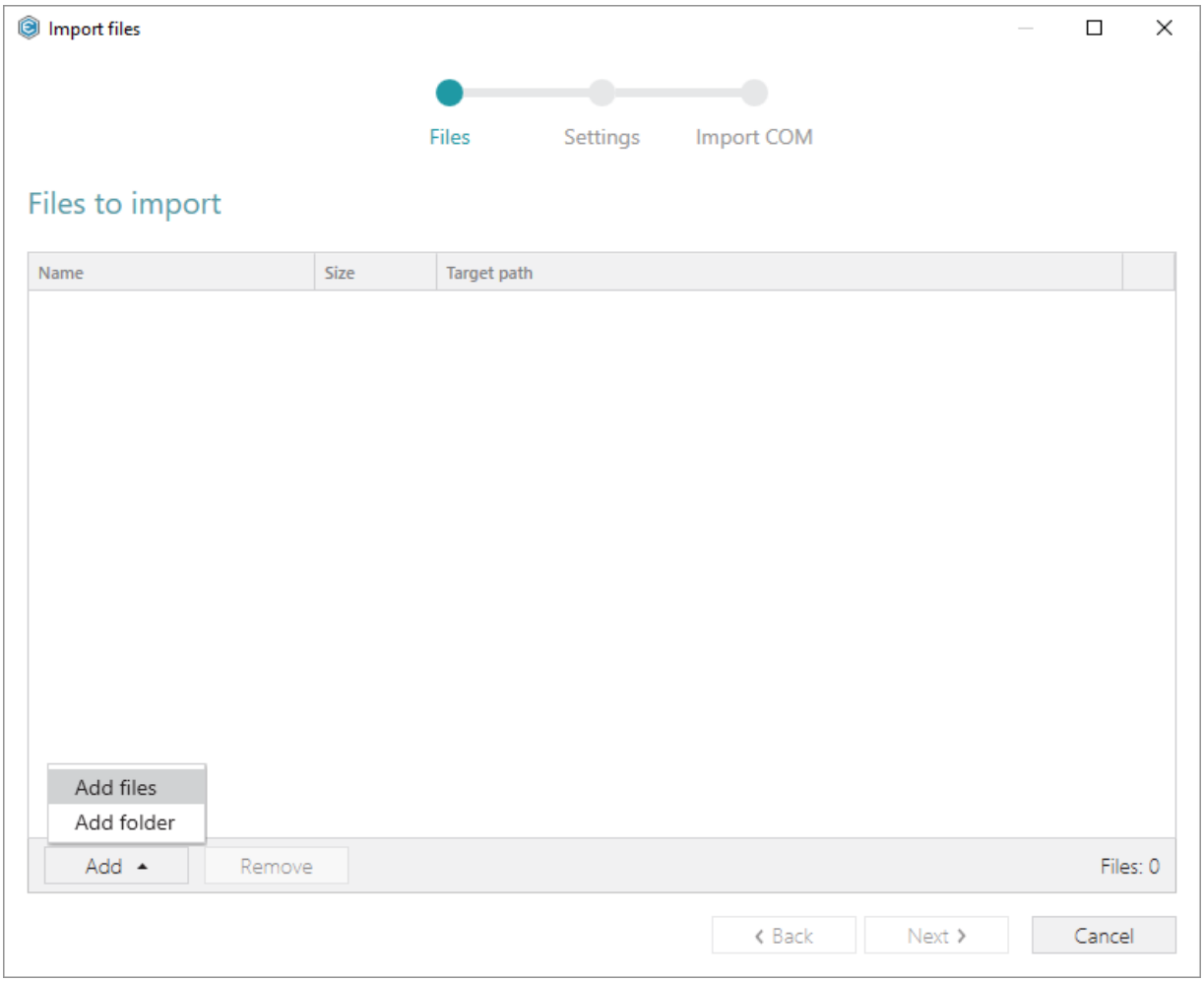
- Go to the Files and folders tab of the Package Designer.



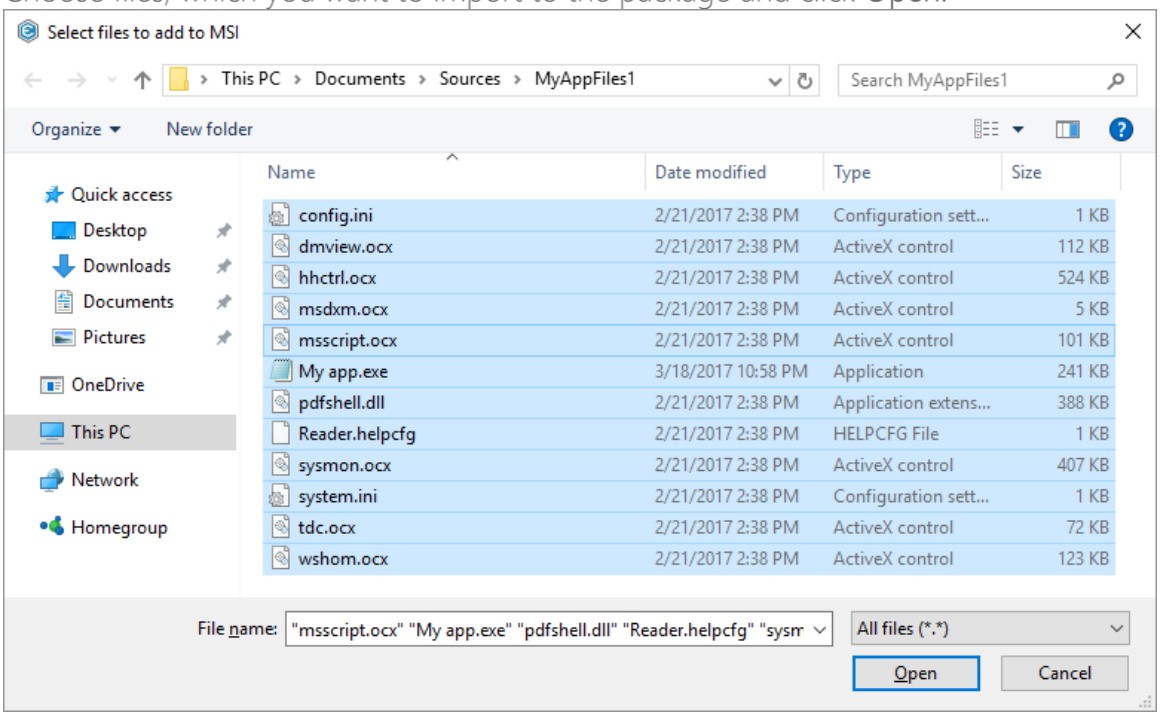
- Select **Import files** from the context menu of a folder, to which you want to import files.



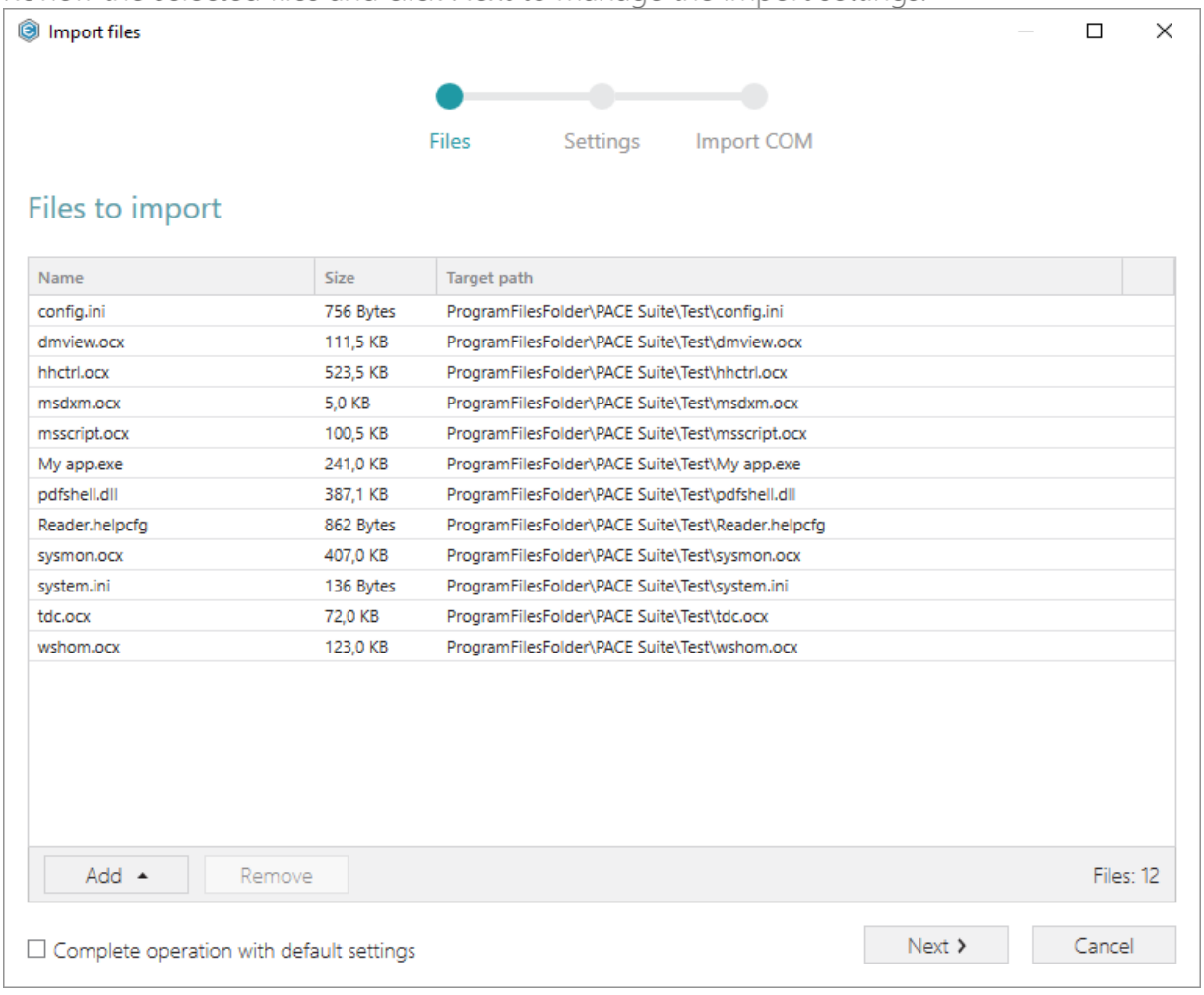

- Click **Add** -> **Add files** to choose files in a folder or **Add** -> **Add folder** to choose a folder with subfolders and files.



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

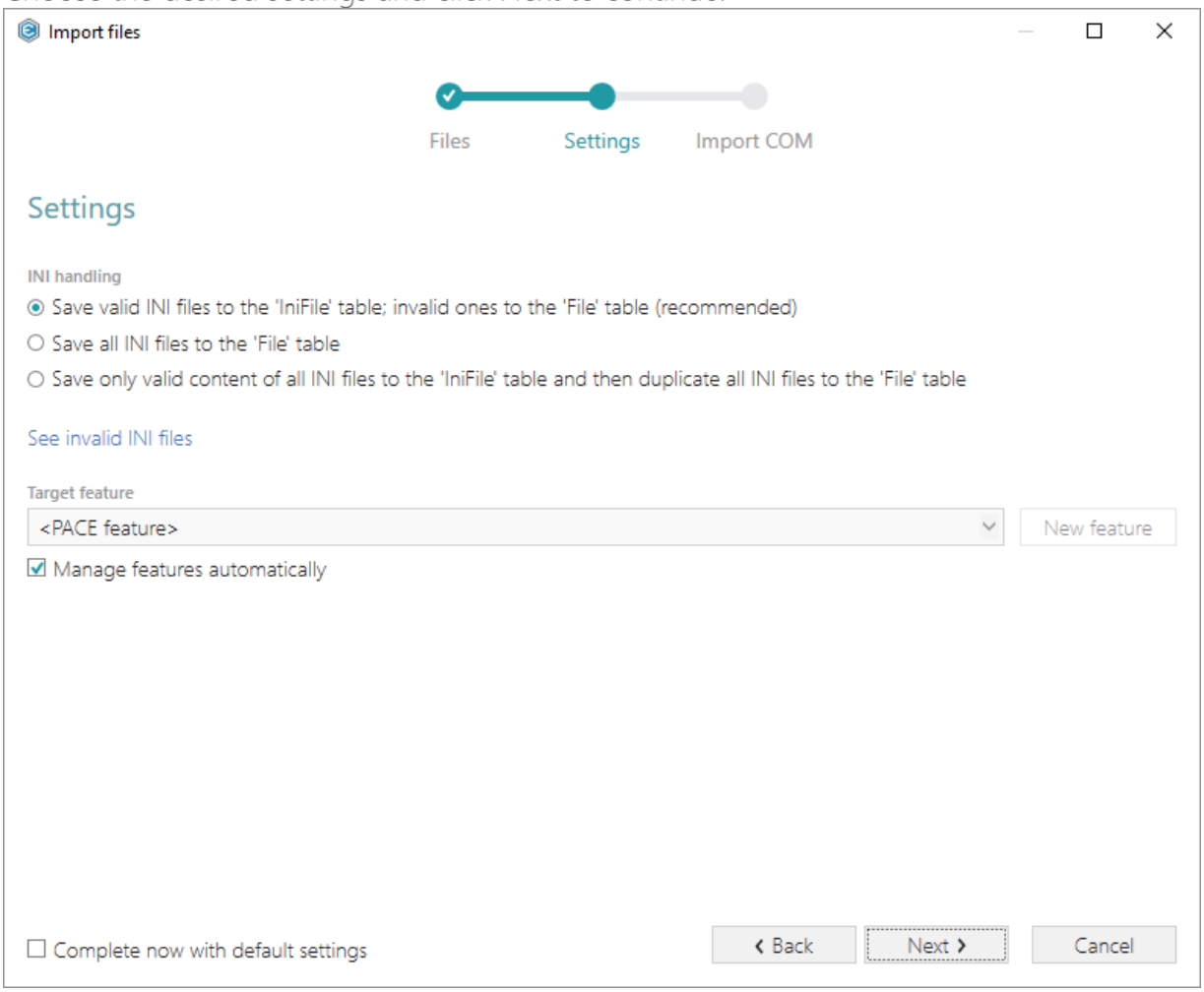



- Review the selected files and click **Next** to manage the import settings.




- At this step, you can manage how to save INI files, select an MSI Feature, to which components with files will be assigned. Below, in the tables, you will find the detailed

description of these settings. Note that files compression types and other Media settings are represented on the separate **Media settings** tab of the **Package Designer**.
Choose the desired settings and click **Next** to continue.



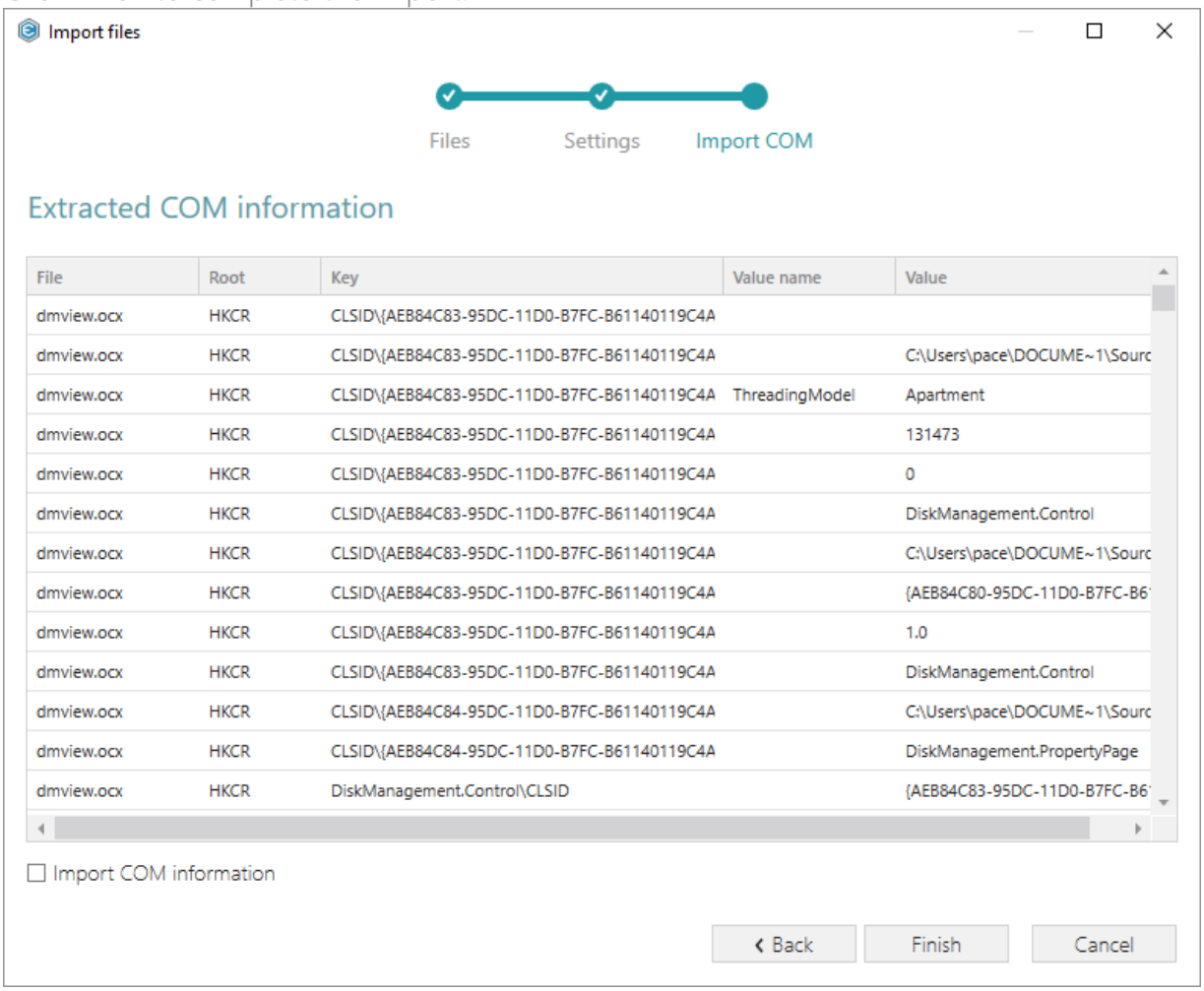

## INI handling

| | |
|---|---|
| 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.<br>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.<br>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

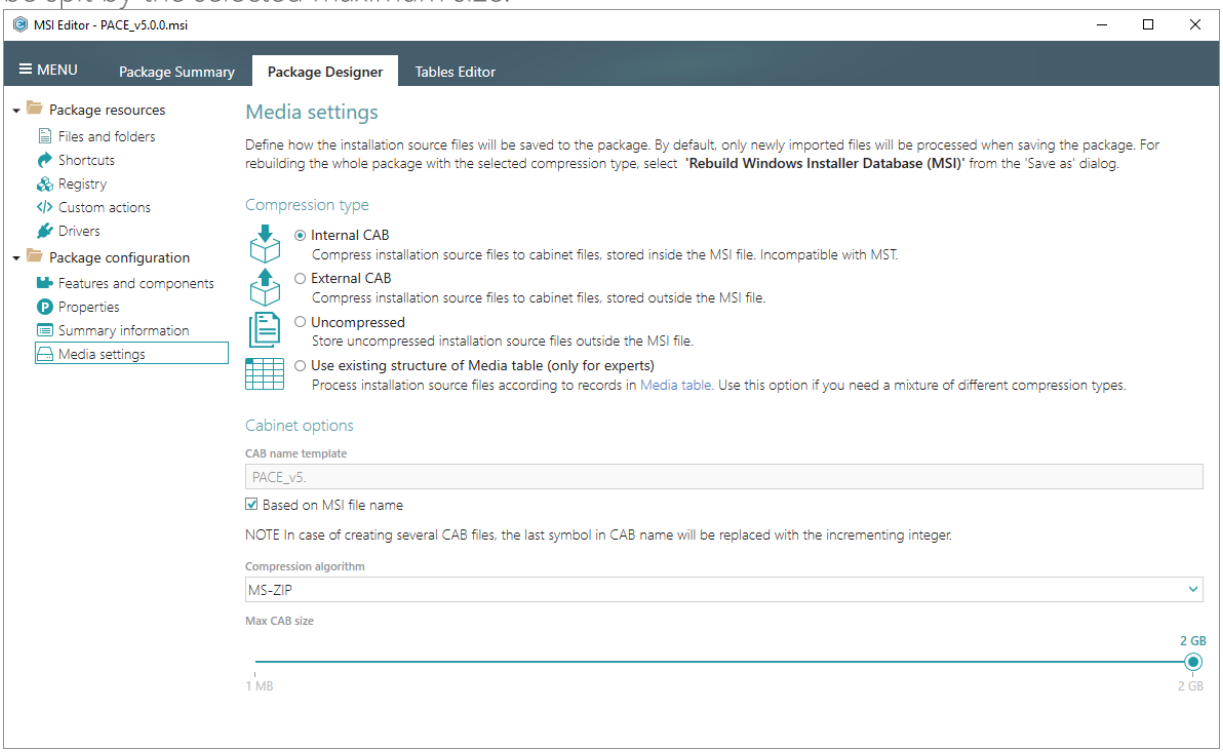
| | |
|---|---|
| Manage features automatically | The specific PACE features will be created (or used if they exist) for the components, containing the imported files. Feature structure is based on the packaging best practices and depends on the destination location of files. |
| <a feature, selected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

- On the last step you can review COM information, extracted from the selected files. For importing this COM information along with files, select the Import COM information option. Click Finish to complete the import.

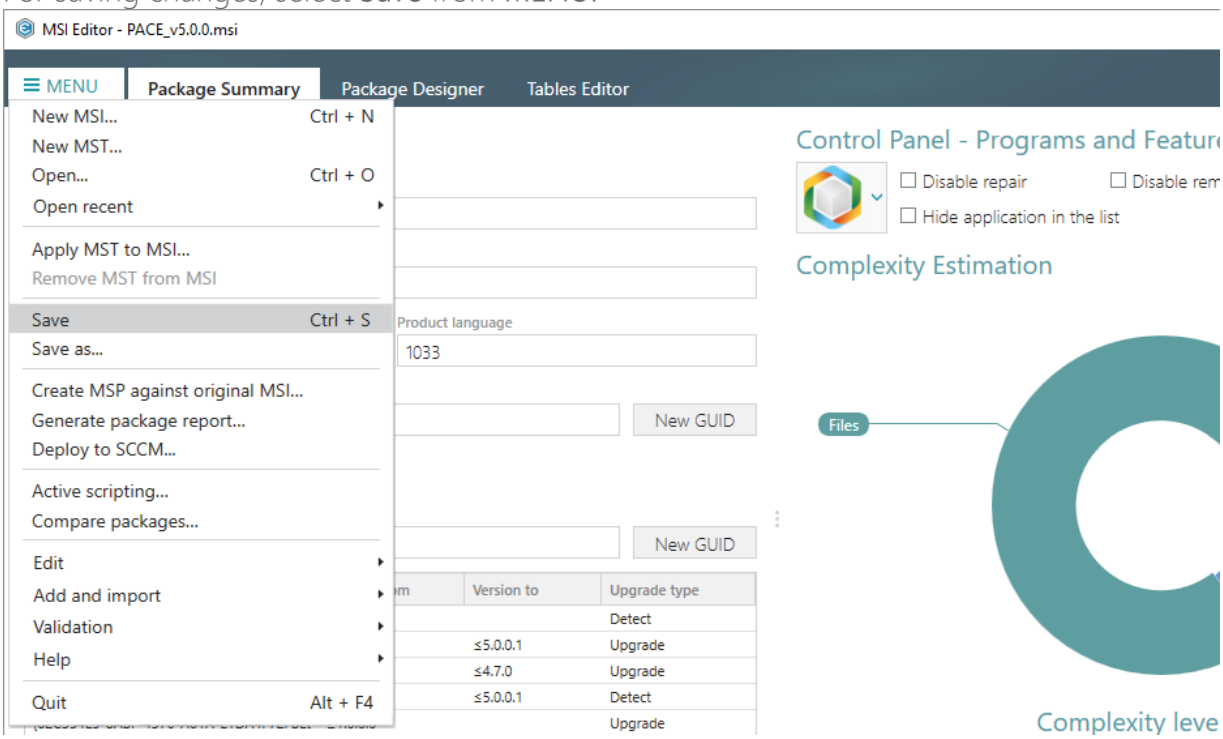



- Go to the Media settings tab of the Package Designer and choose the necessary compression type for the imported files. For the compressed files you can manage cabinet options such as CAB name, compression algorithm and maximum CAB size. Cabinets will

be spit by the selected maximum size.



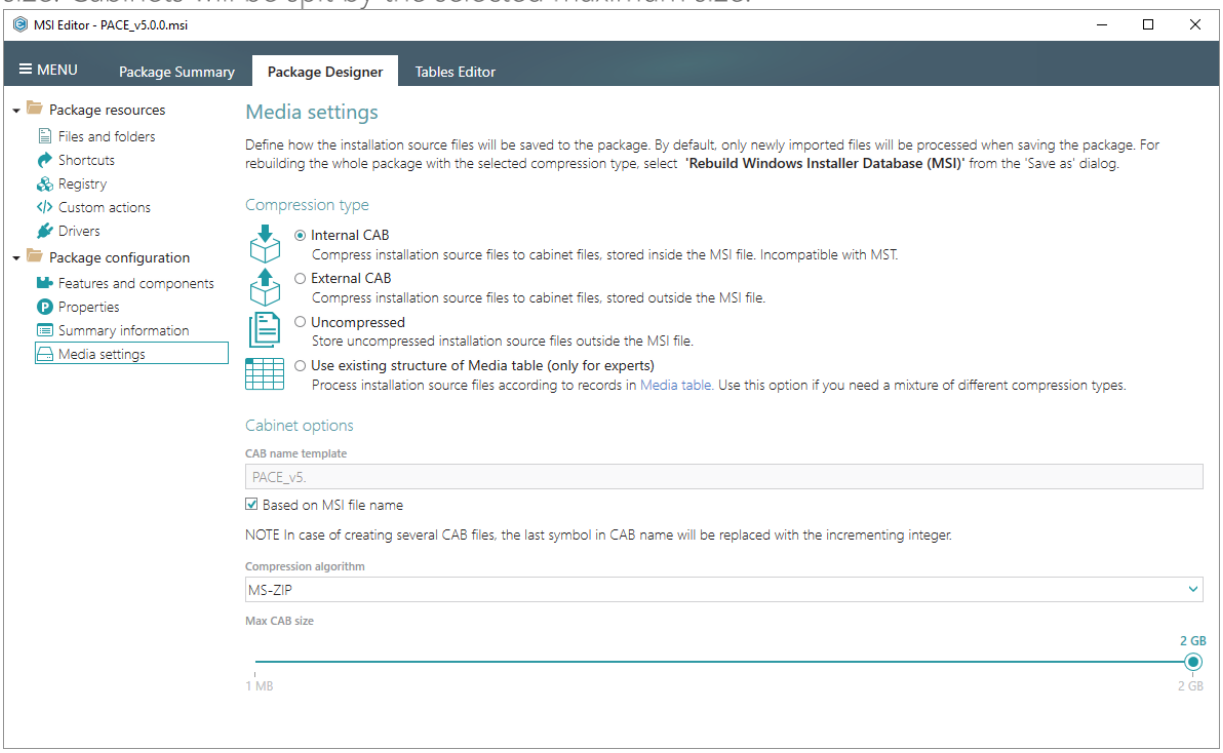

- For saving changes, select Save from MENU.



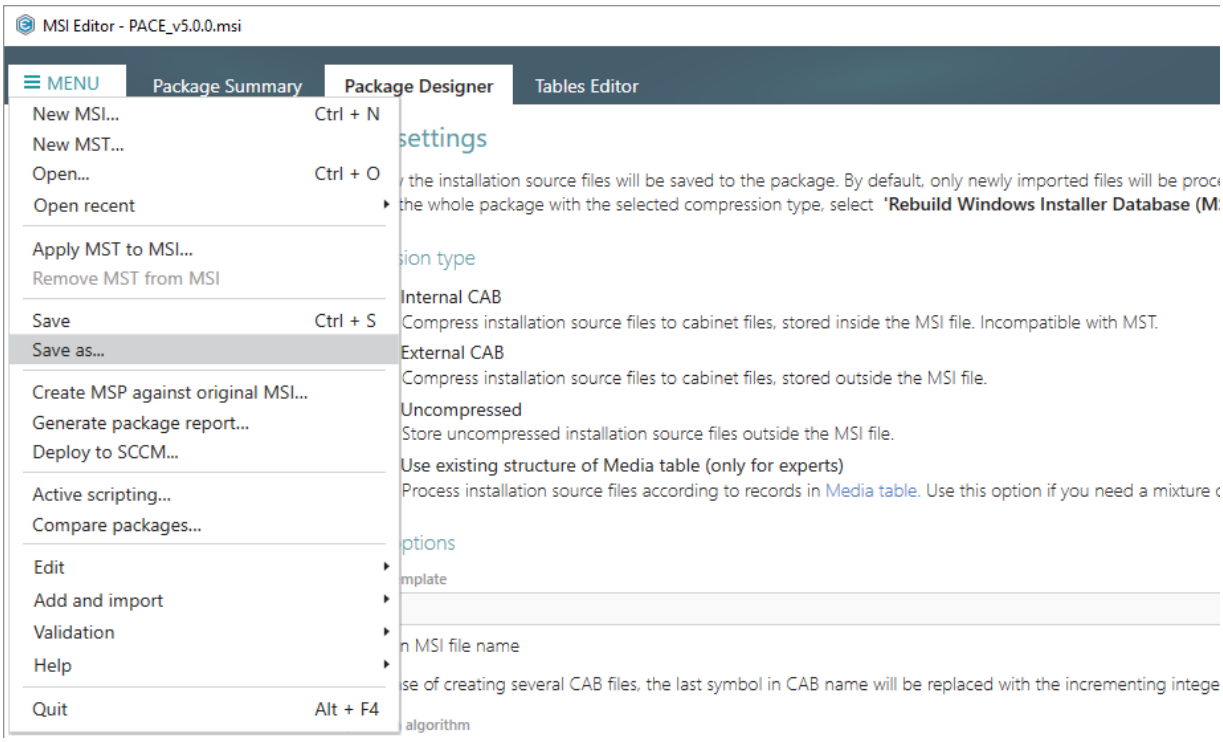

### 3.7.2.4 Change Compression Type

Rebuild the whole package with the selected compression type. For this, all your changes will be merged with MSI and saved as a new MSI package in the location, which you select. Then, all files of this MSI package will be extracted and then saved again using the selected compression type and cabinet options.

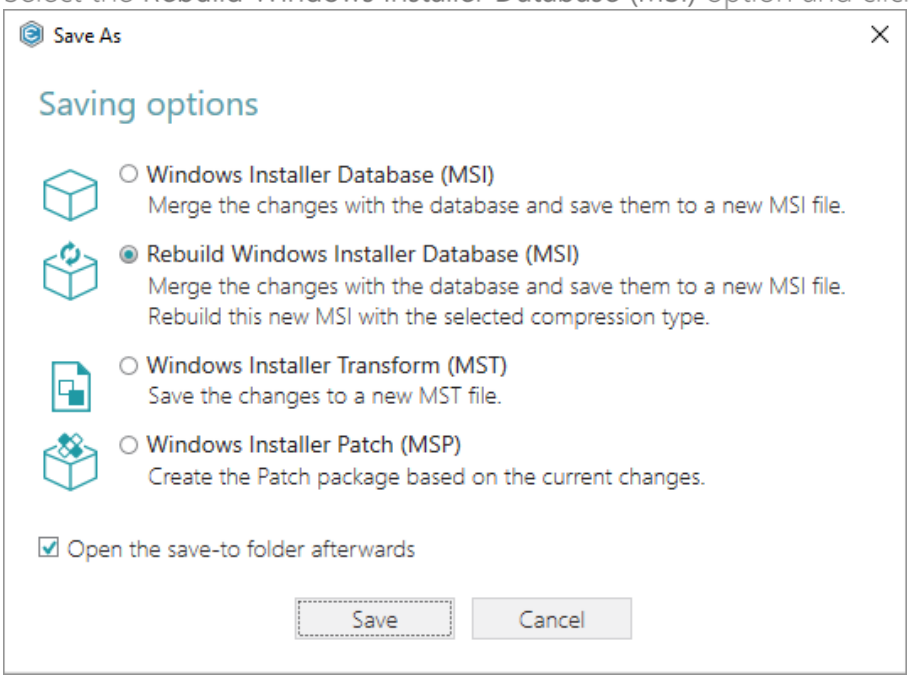
- Go to the Media settings tab of the Package Designer and select the desired compression type that will be applied to all package files. For the first two compression types you can manage cabinet options such as CAB name, compression algorithm and maximum CAB size. Cabinets will be spit by the selected maximum size.



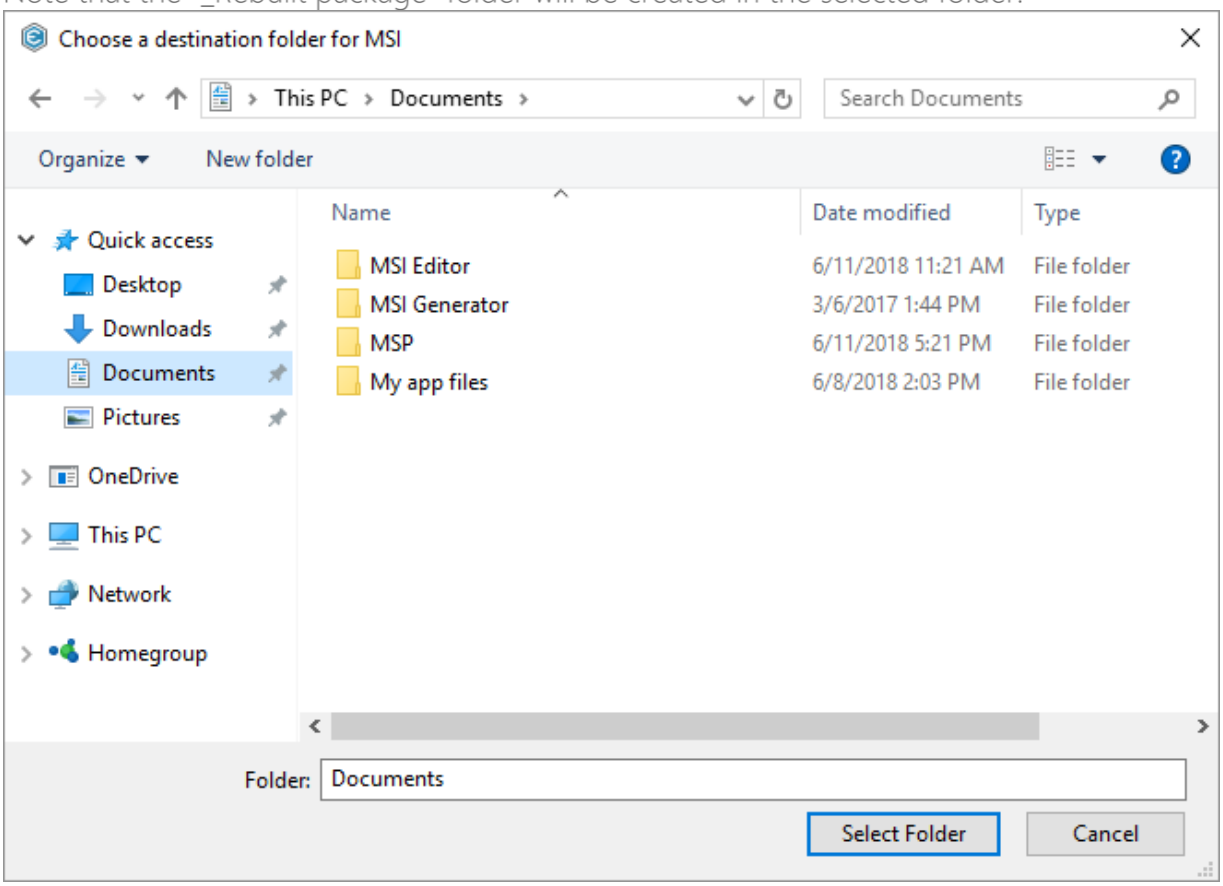

- Select Save as from MENU.



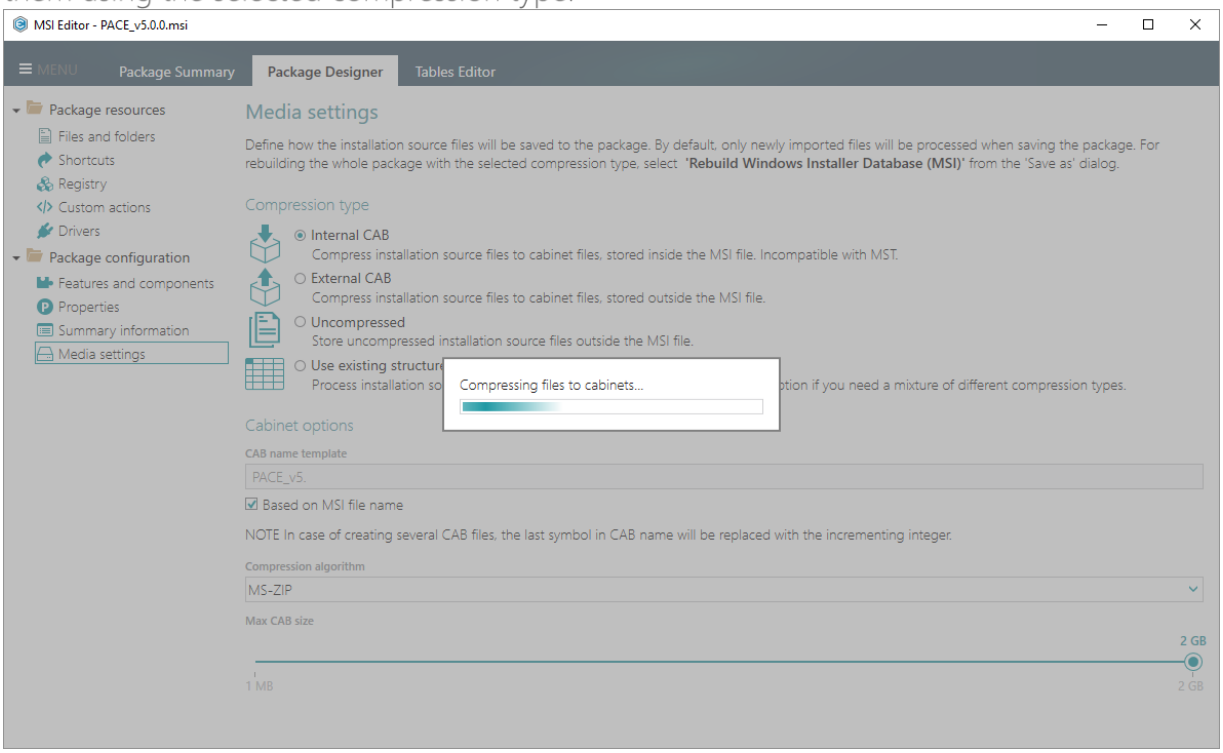
- Select the Rebuild Windows Installer Database (MSI) option and click Save.



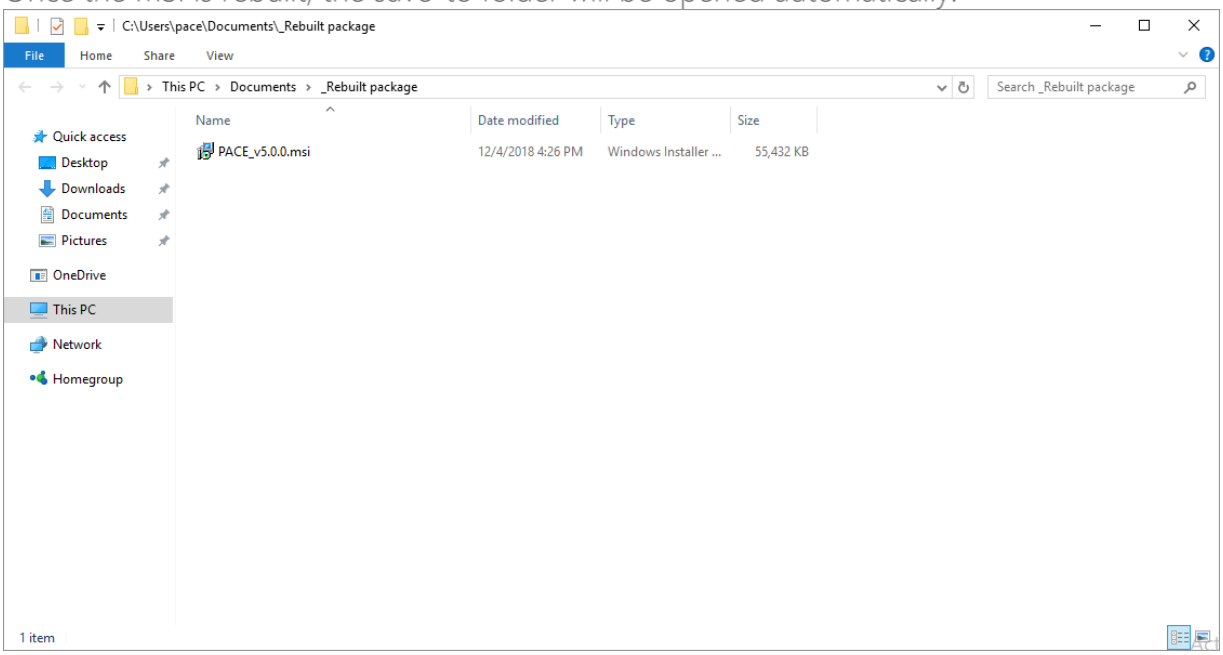

- Choose a destination folder, where the rebuilt MSI will be saved and click Select Folder. Note that the "_Rebuilt package" folder will be created in the selected folder.



- Wait a little, while the MSI Editor is extracting all files from the package and processing them using the selected compression type.



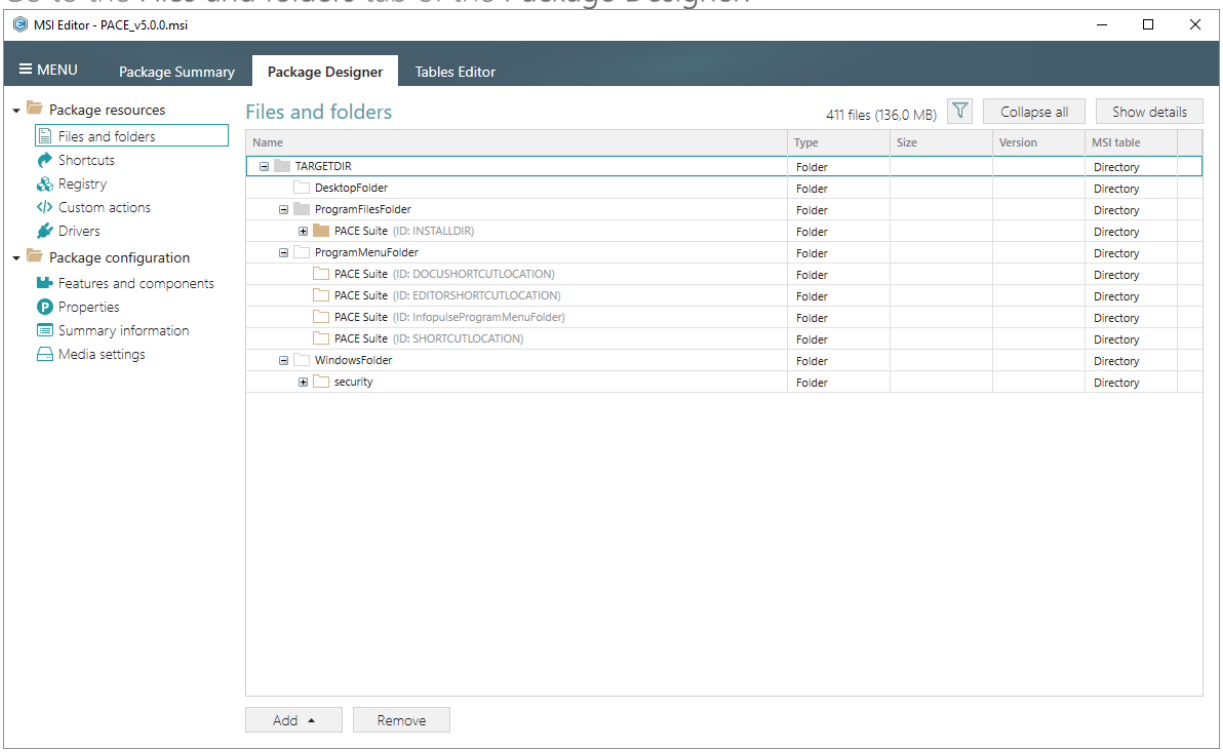

- Once the MSI is rebuilt, the save-to folder will be opened automatically.



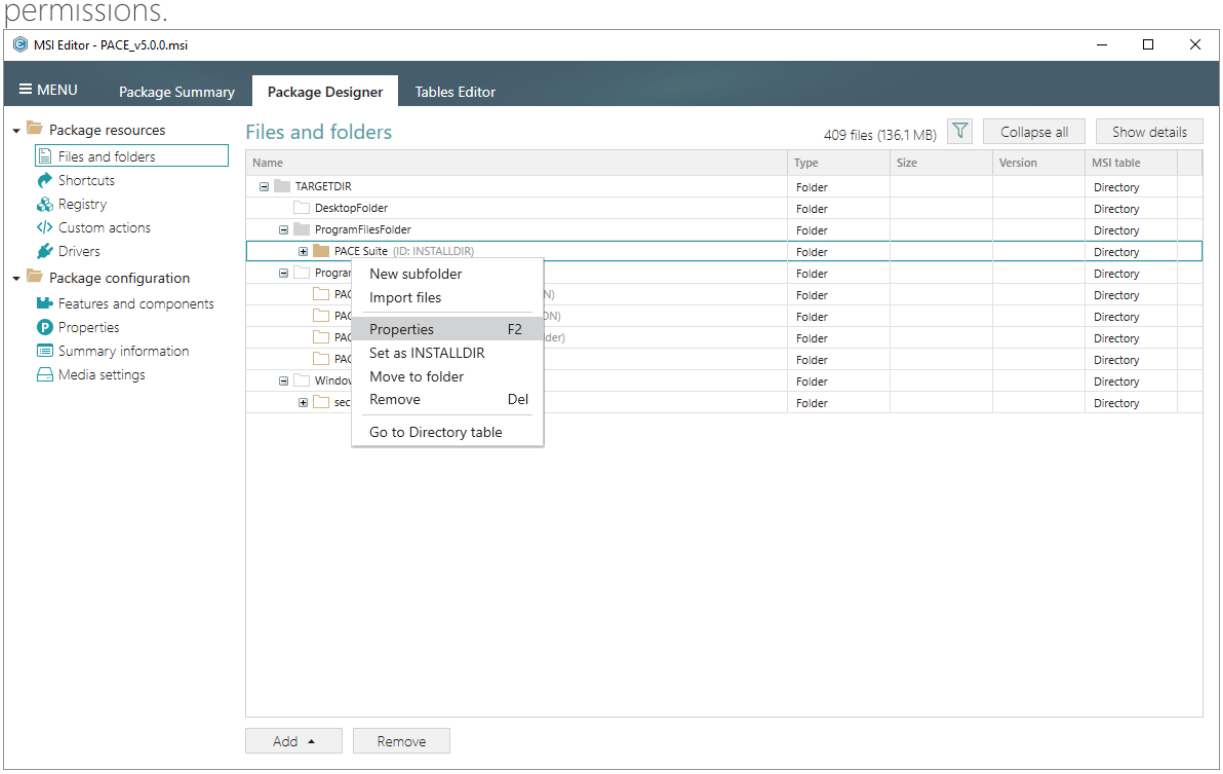
### 3.7.2.5 Set Permissions

Change permissions for the file or the folder.

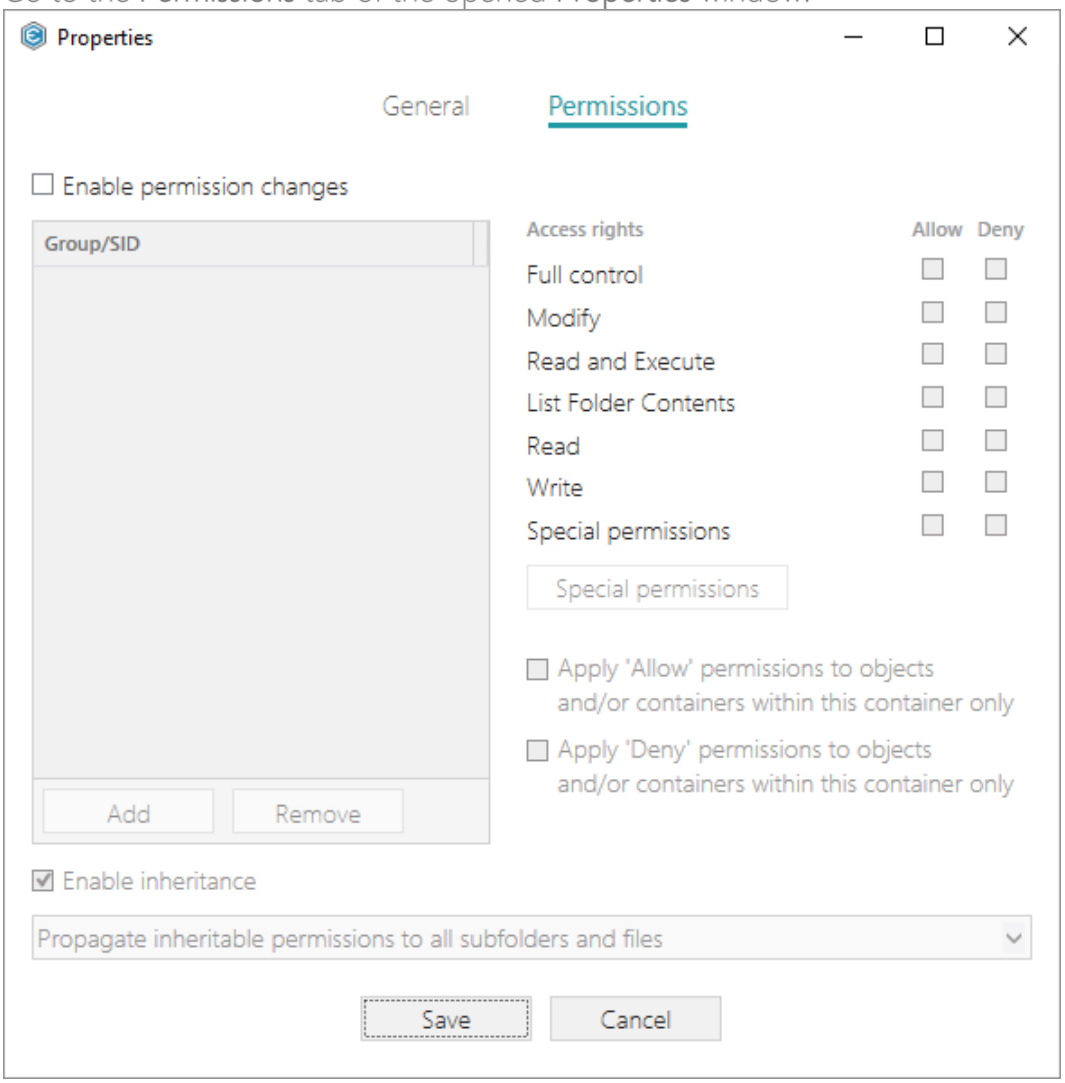
- Go to the **Files and folders** tab of the **Package Designer**.



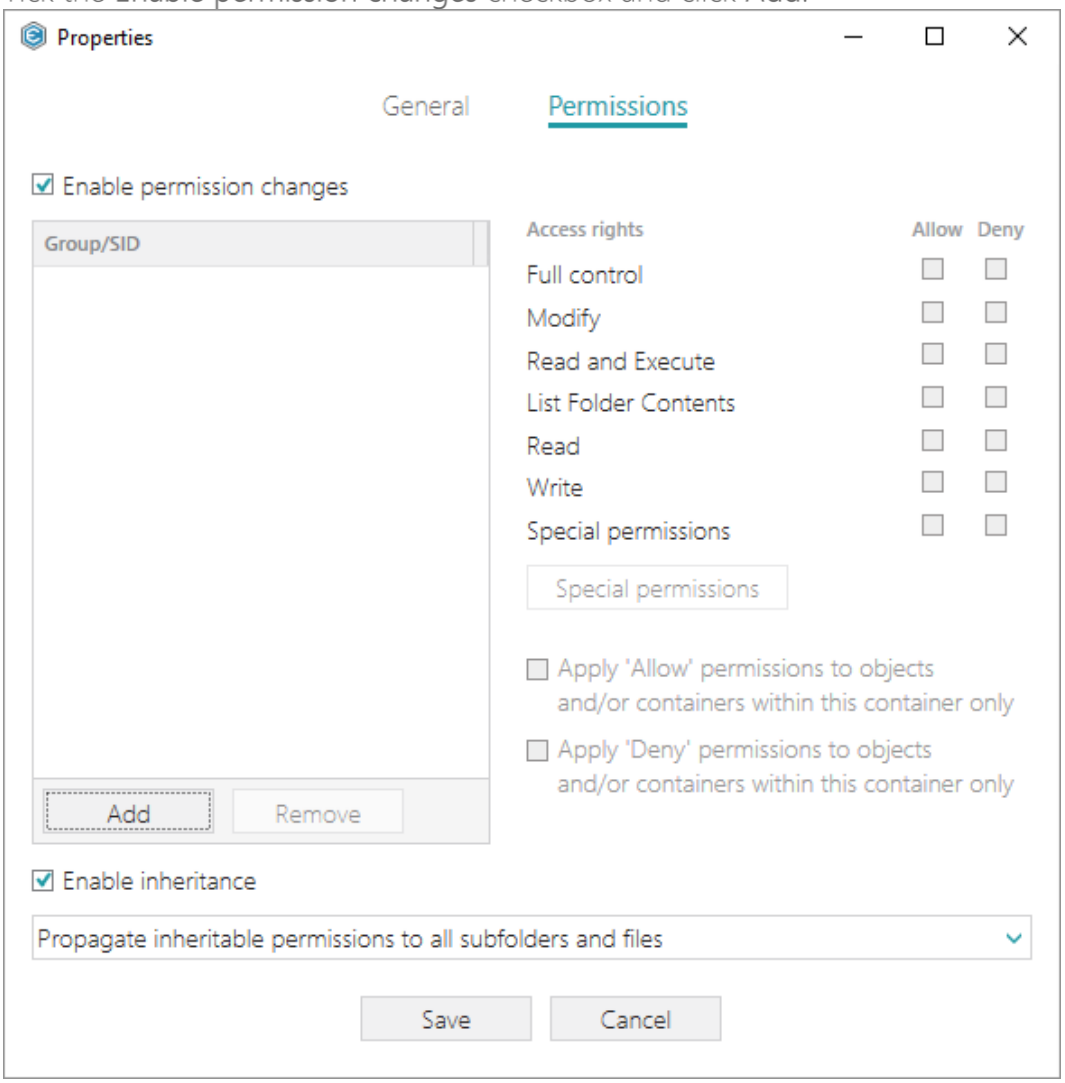

- Select **Properties** from the context menu of a folder or a file, for which you want to set permissions.



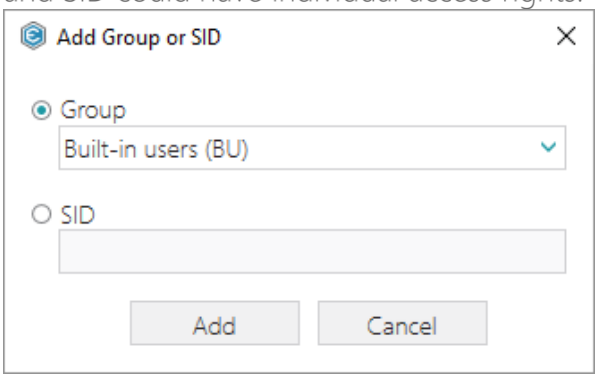
- Go to the Permissions tab of the opened Properties window.



- Tick the Enable permission changes checkbox and click Add.

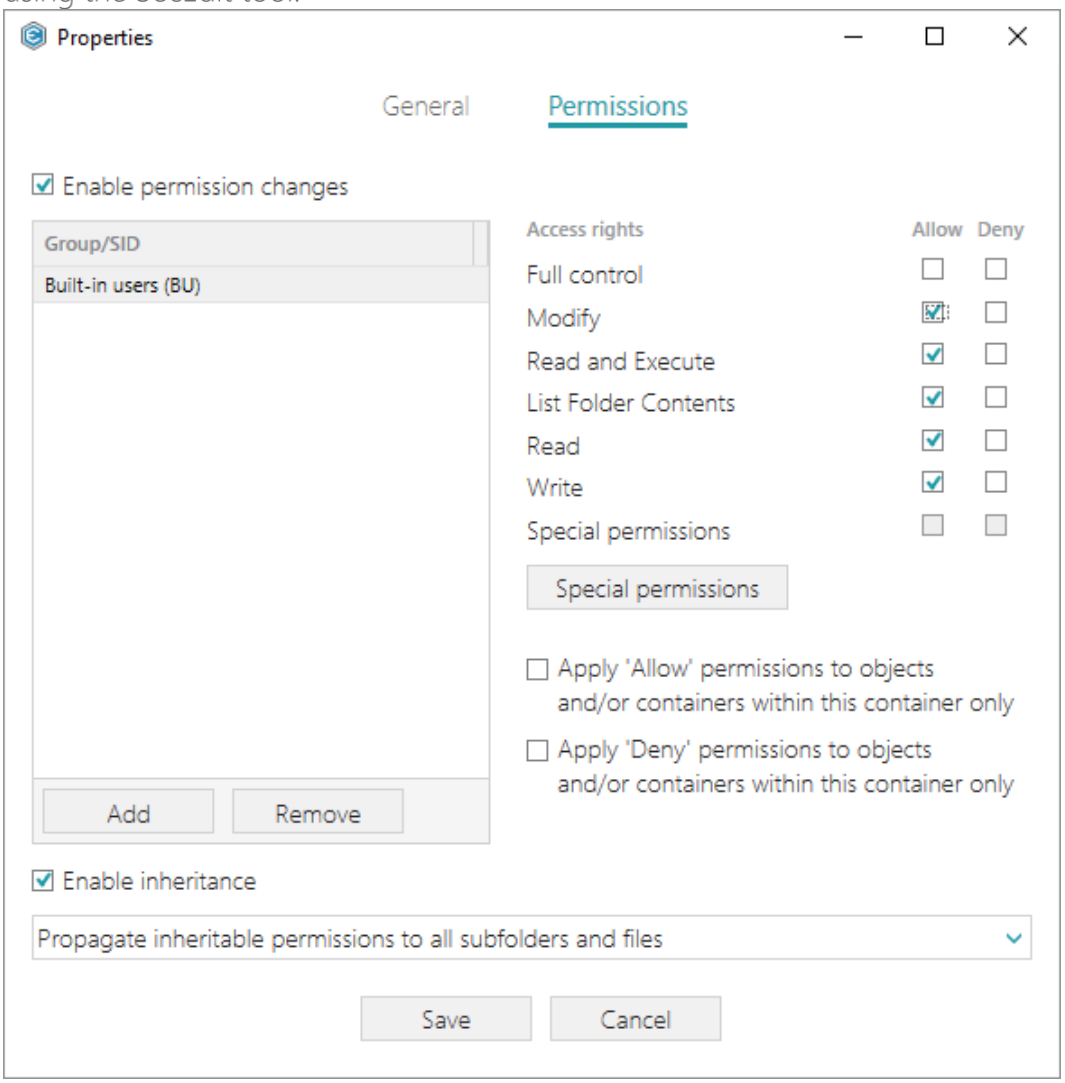



- Select a group from the list or enter SID manually, for which you want to set permissions, and click Add. Note that you can add as many groups and SID as you need. Each group and SID could have individual access rights.

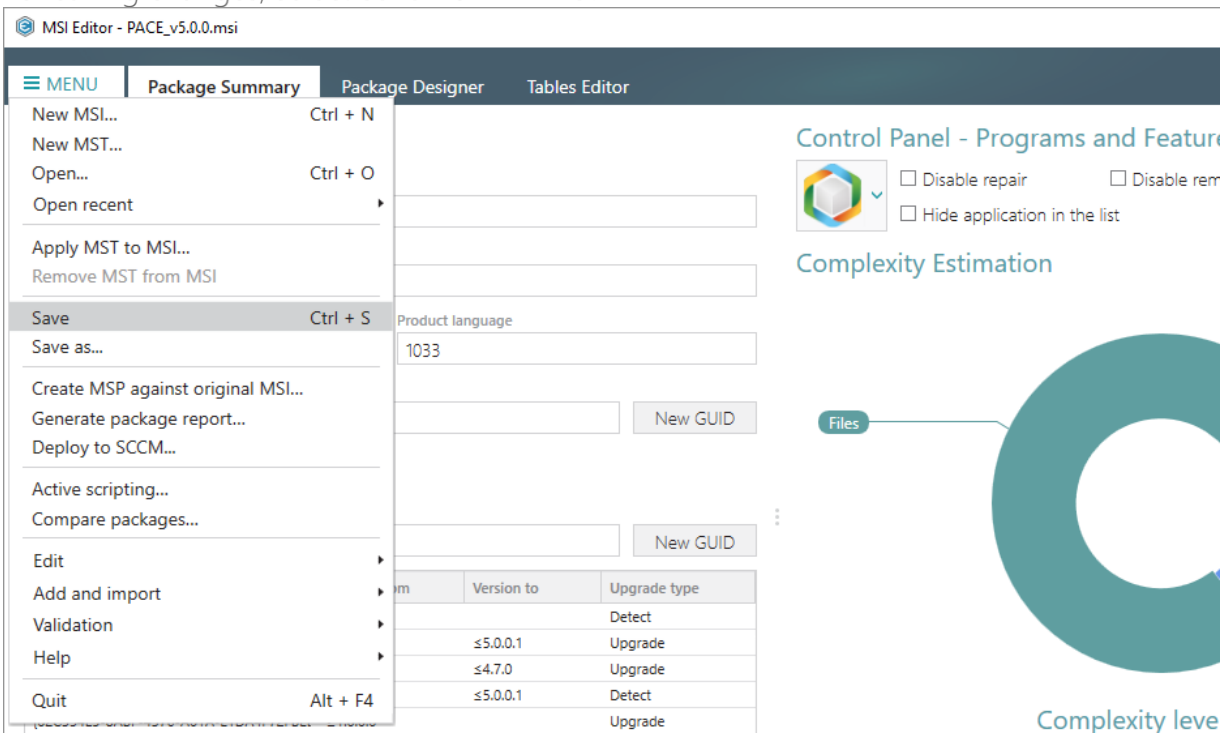



- Tick the necessary access rights for the selected group or SID, change (if needed) special permissions, inheritance options and behavior of applying the permissions. Click Save to save the changed permission settings to MSI. Note that permissions will be propagated

using the SecEdit tool.



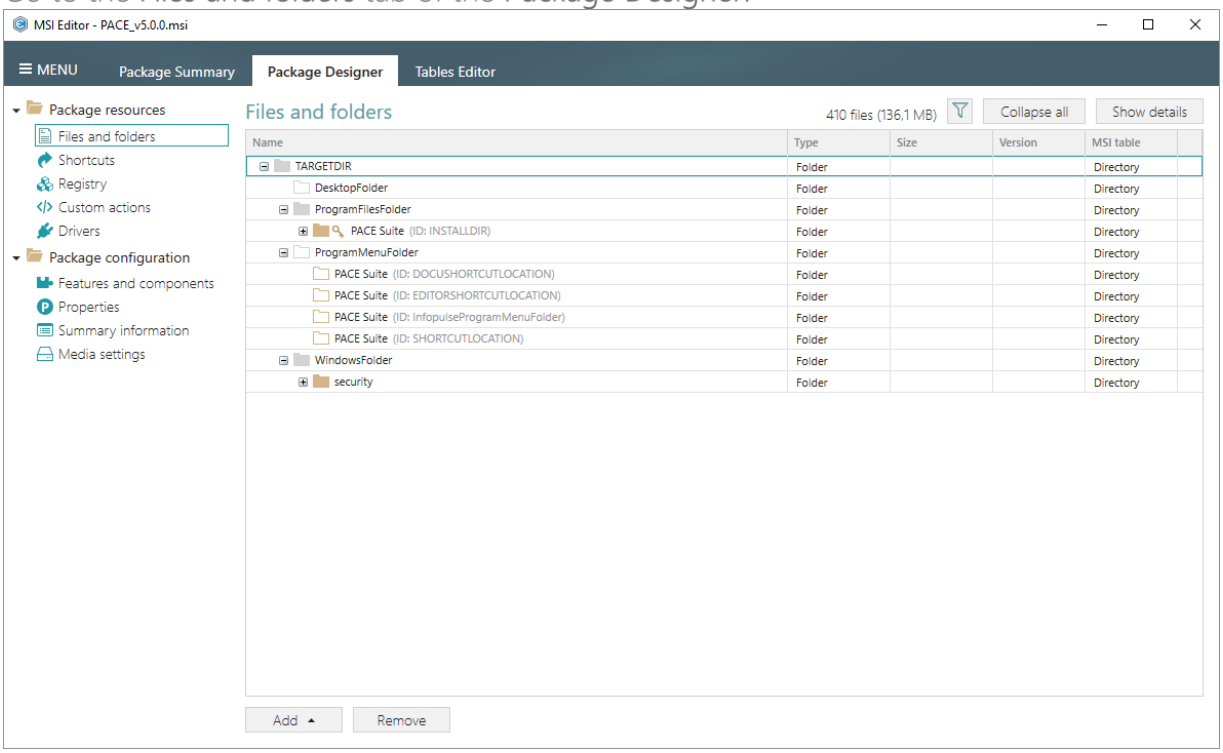

- For saving changes, select Save from MENU.



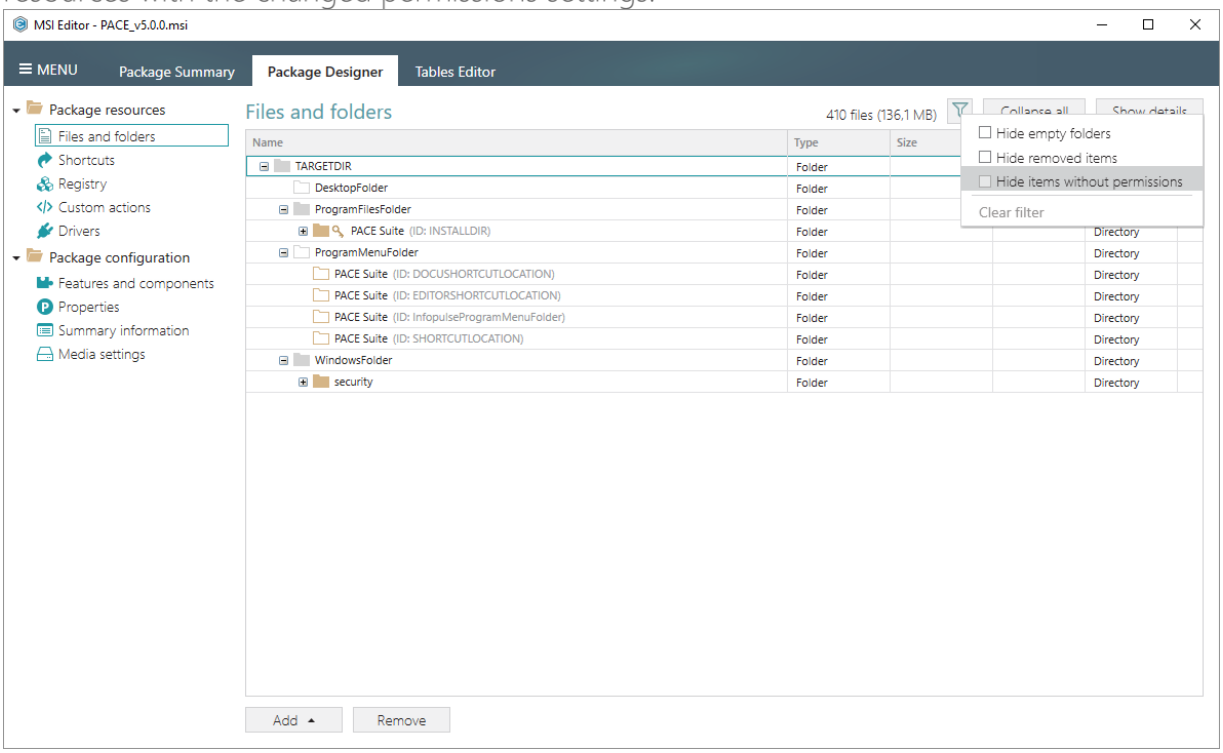
### 3.7.2.6 Remove Permissions

Locate files and folders with custom permissions and remove the unnecessary permissions.

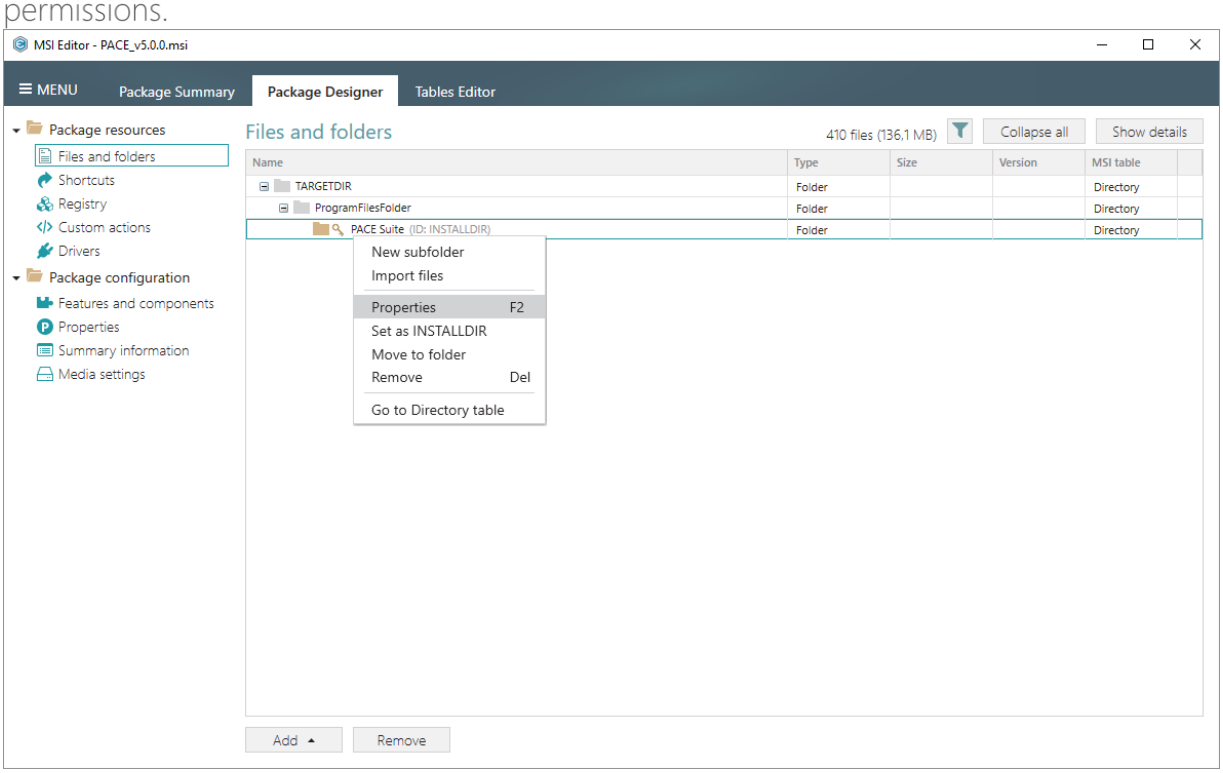
- Go to the **Files and folders** tab of the **Package Designer**.



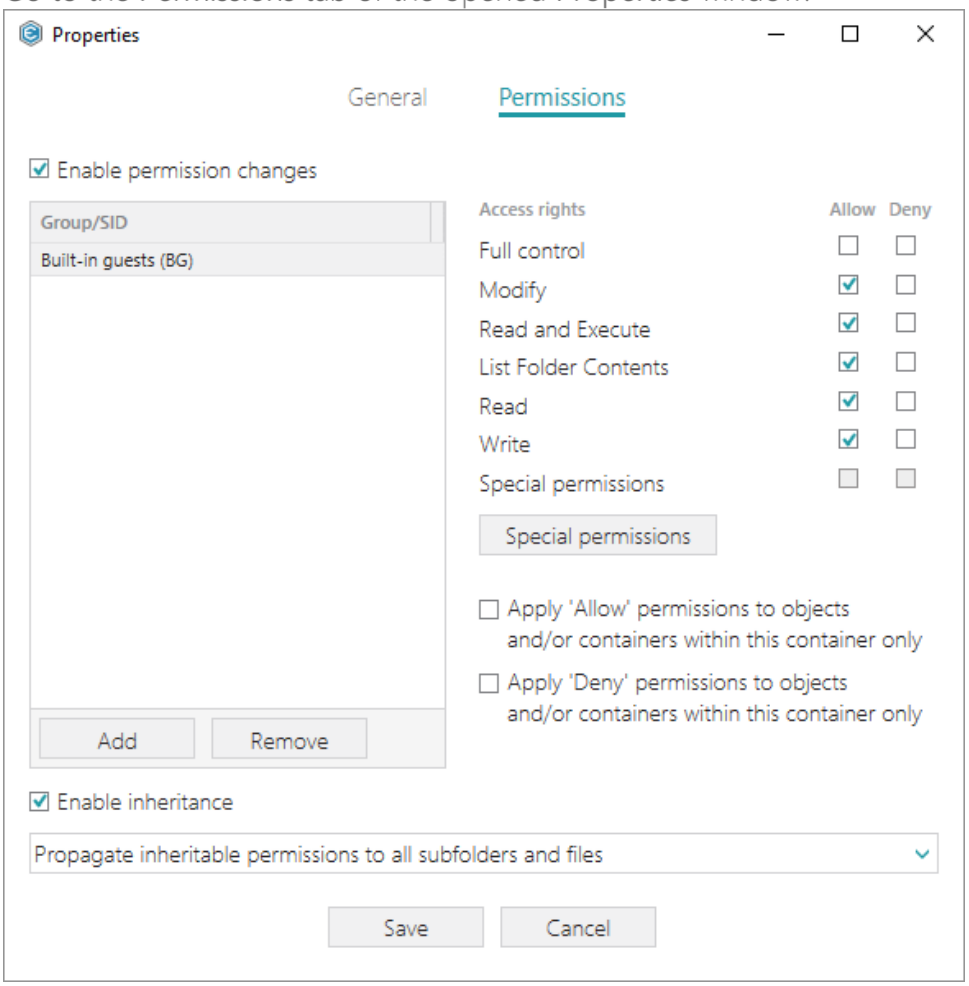

- (Optional) Click on the **Filter** and apply the **Hide items without permissions** filter to locate resources with the changed permissions settings.



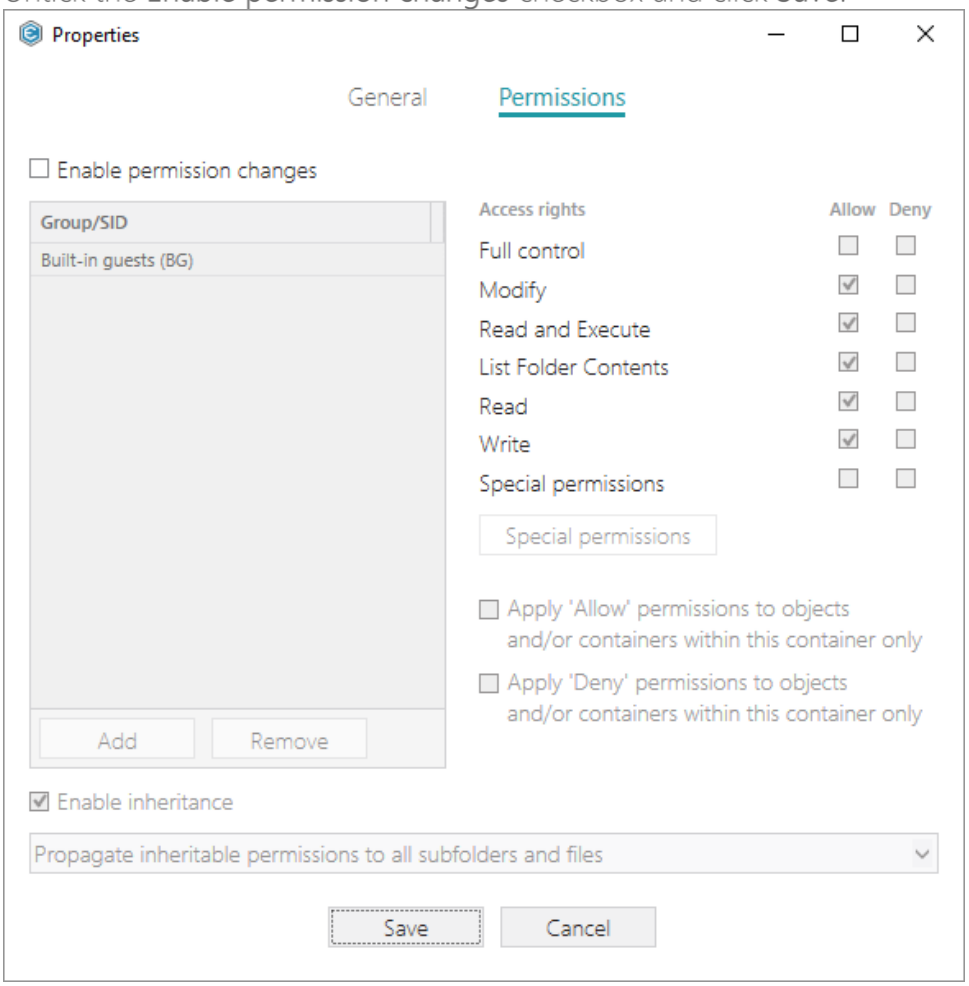
- Select **Properties** from the context menu of a folder or a file, for which you want to remove permissions.



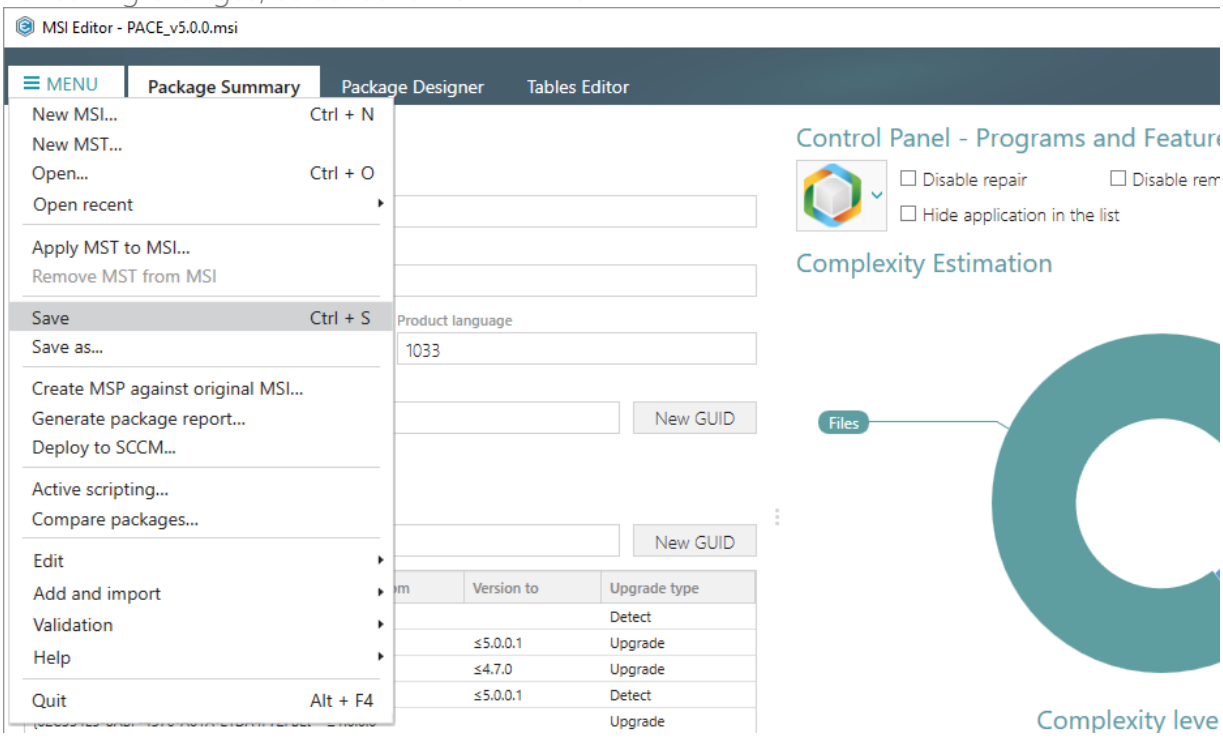

- Go to the **Permissions** tab of the opened **Properties** window.



- Untick the **Enable permission changes** checkbox and click **Save**.



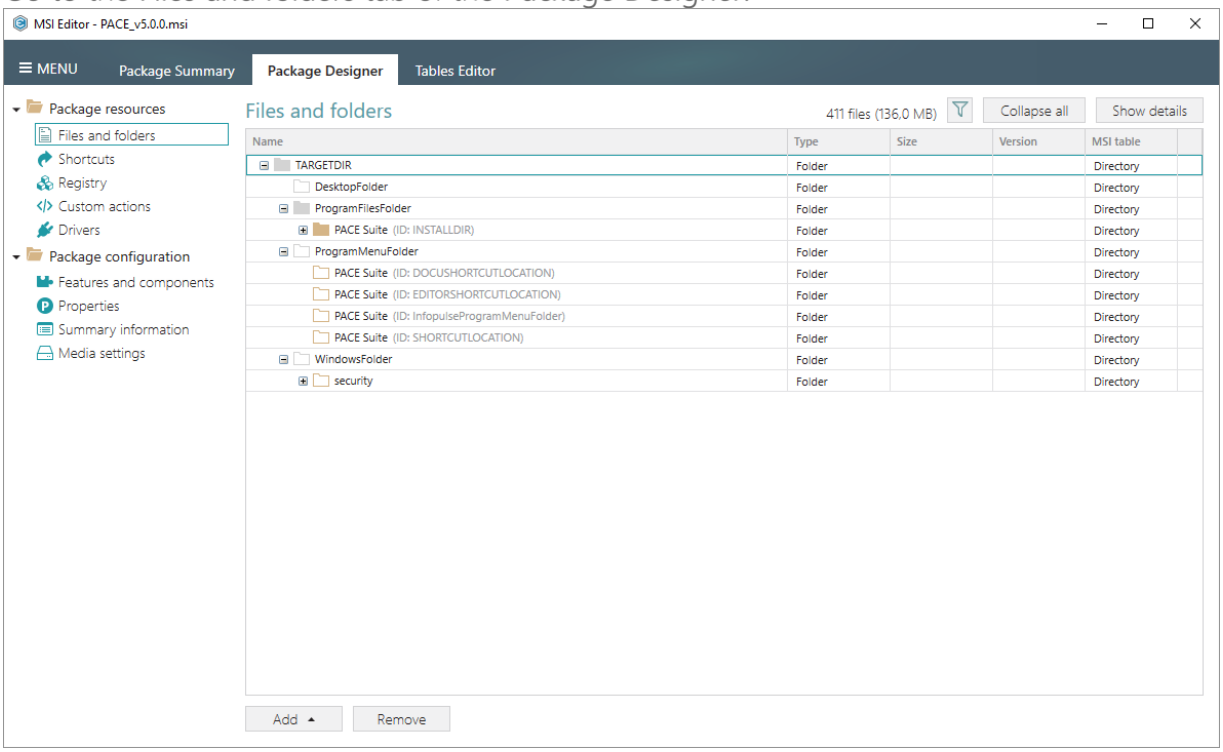

- For saving changes, select **Save** from **MENU**.



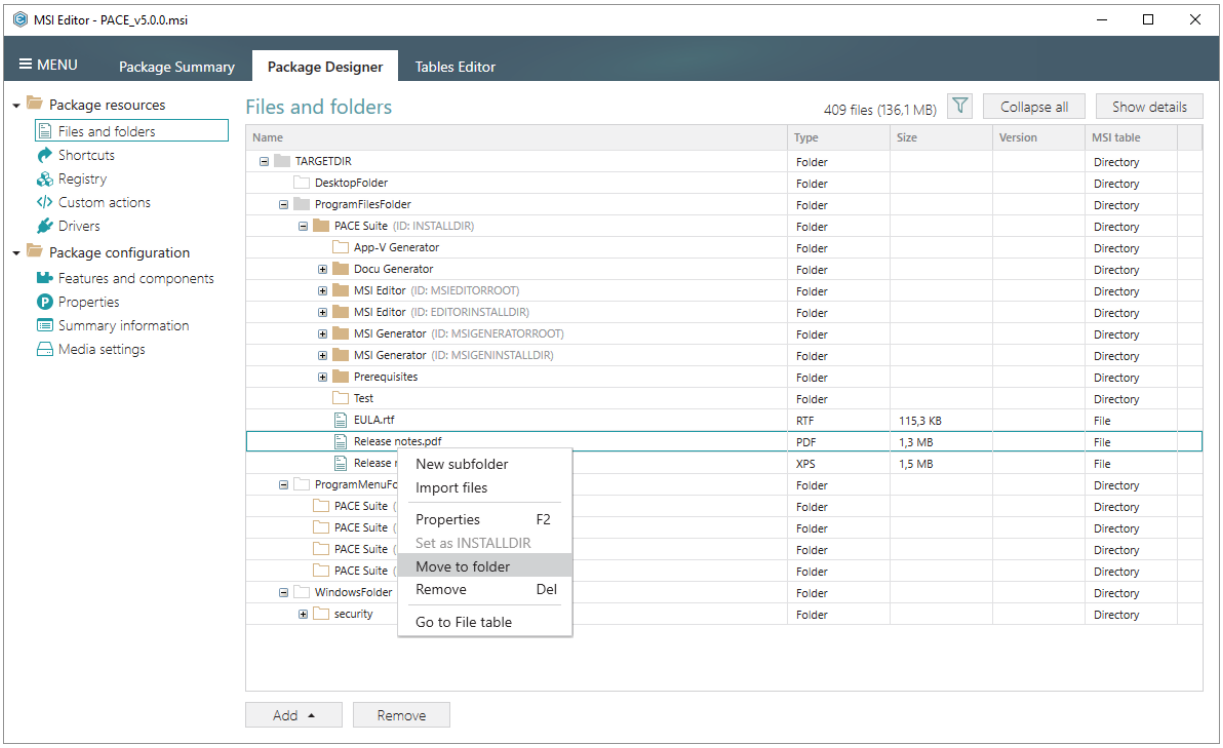
### 3.7.2.7 Move Files, Folders

Move a file or a folder from one folder to another. Note that moving few files of the multi-file component will result in adding new component and/or changing component details.

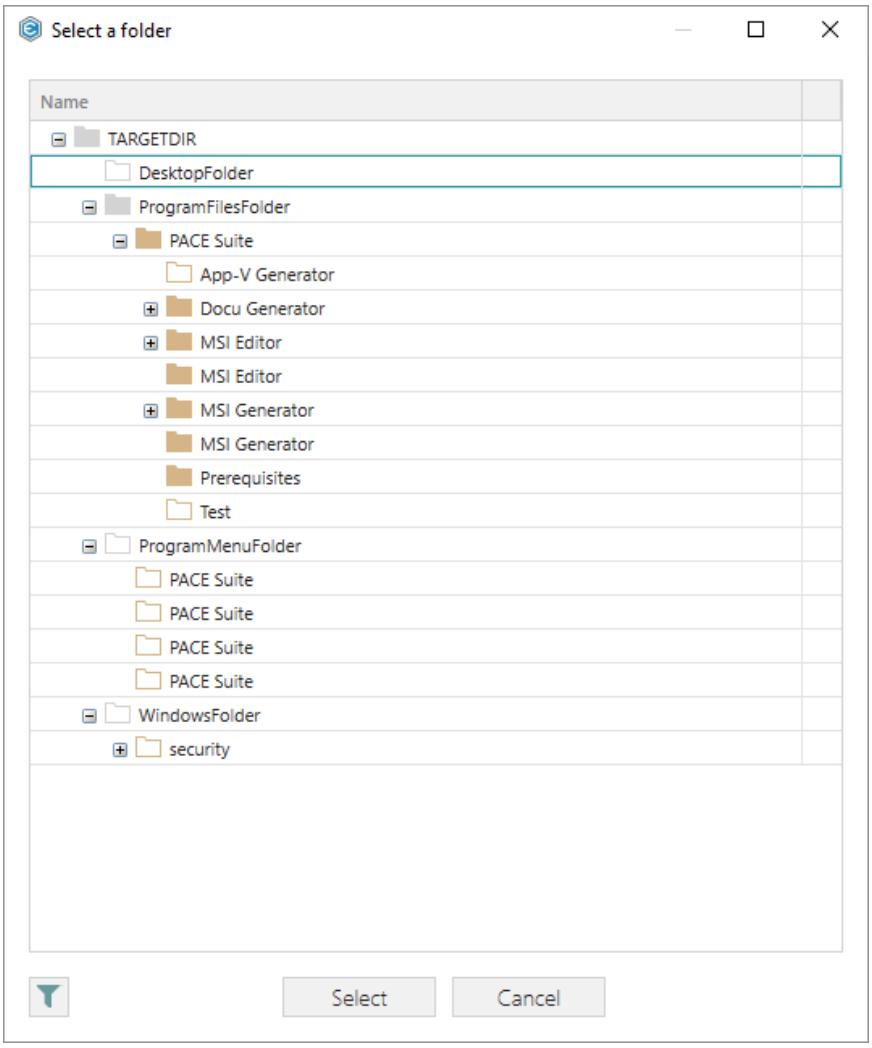
- Go to the **Files and folders** tab of the **Package Designer**.



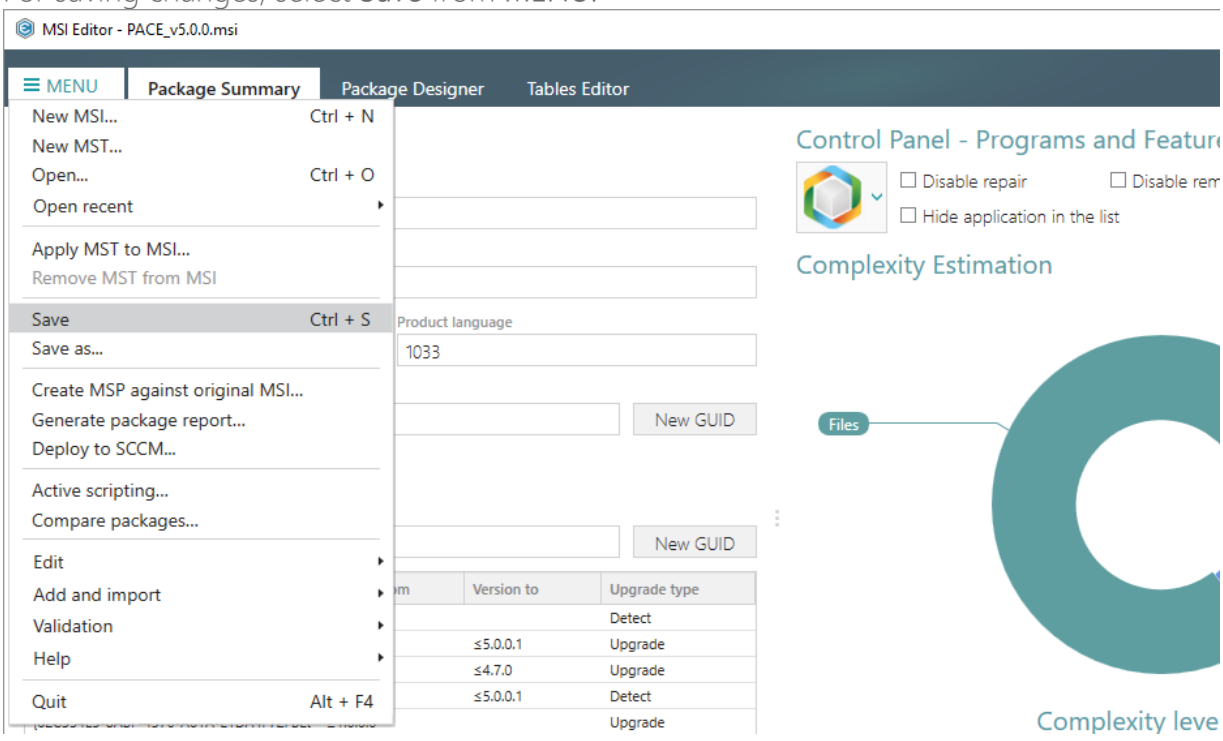

- Select **Move to folder** from the context menu of a file or a folder, which you want to move to another folder.



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



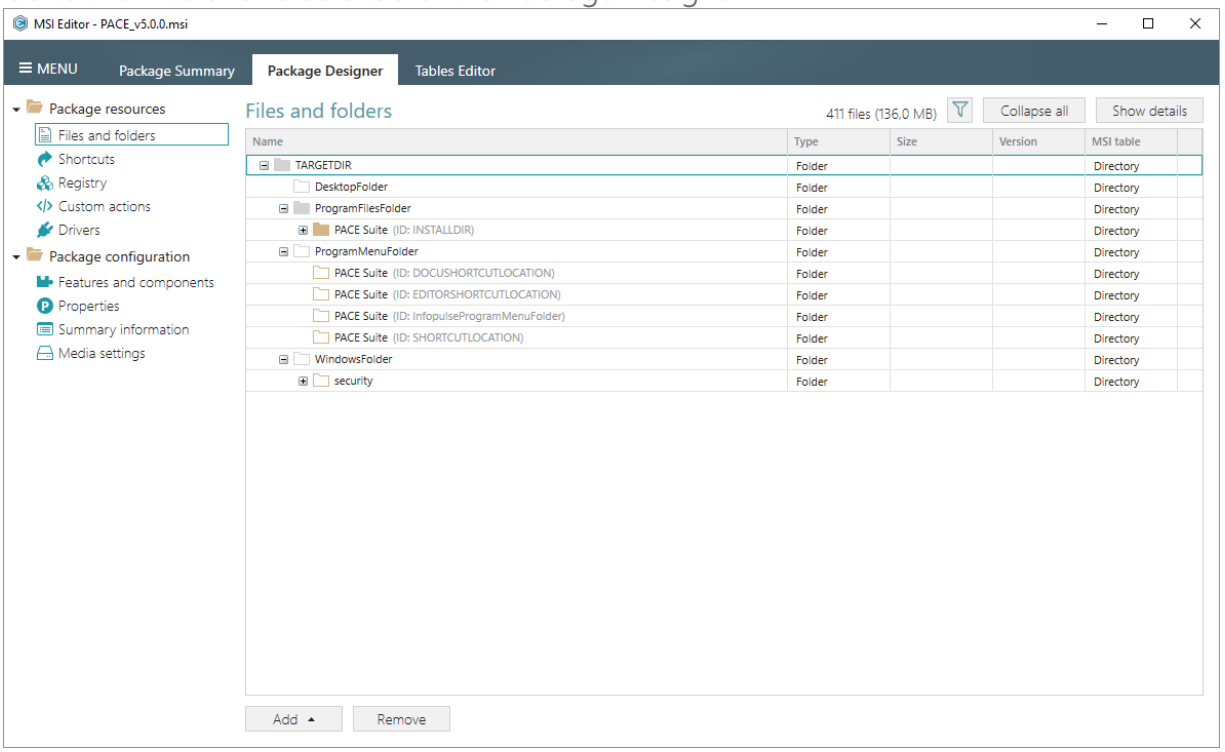

- For saving changes, select **Save** from **MENU**.



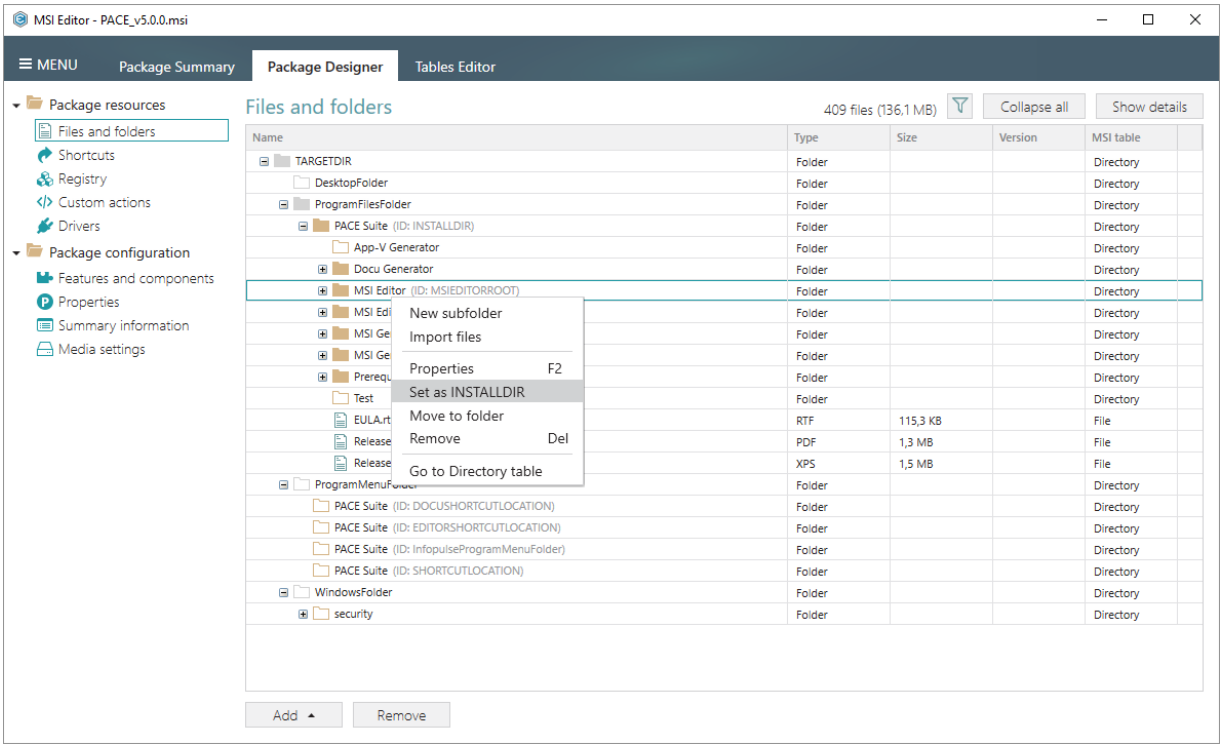
### 3.7.2.8 Set INSTALLDIR

Set a subfolder (user-defined folder) of the package as the INSTALLDIR. This means that an identifier of the selected folder will be changed to the 'INSTALLDIR' value. Note that identifier of a folder is used as the folder property. 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 Files and folders tab of the Package Designer.

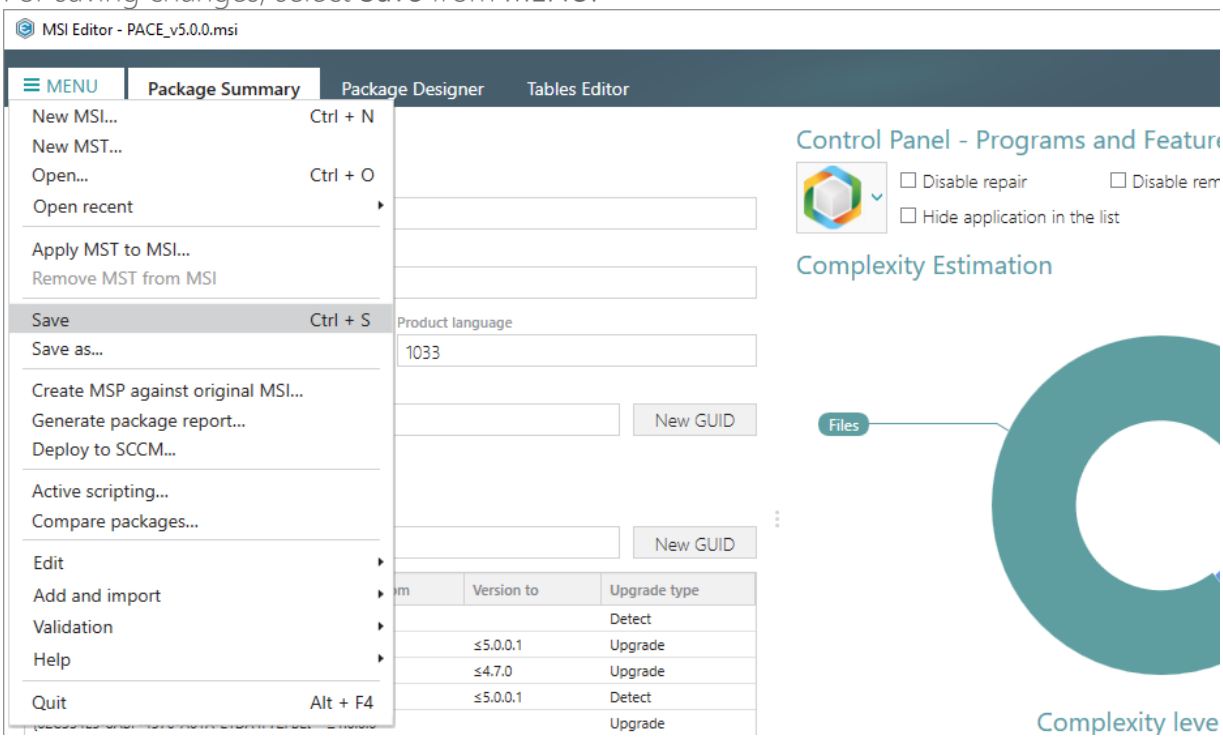



- Select Set as INSTALLDIR from the context menu of a subfolder (user-defined folder), which you want to set as the INSTALLDIR property. Note that system folders, displayed using the grey folder icon, could not be set as INSTALLDIR.




The INSTALLDIR identifier is displayed in the parentheses.

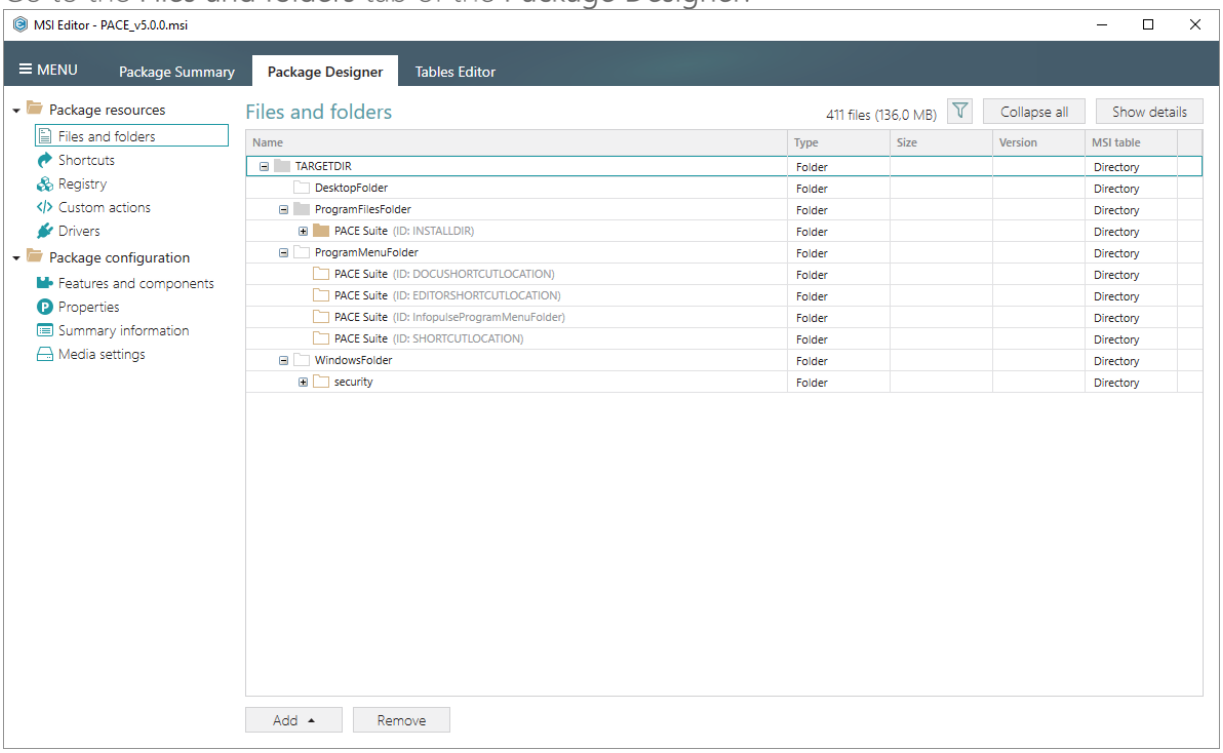
- For saving changes, select Save from MENU.



### 3.7.2.9 Edit INI File

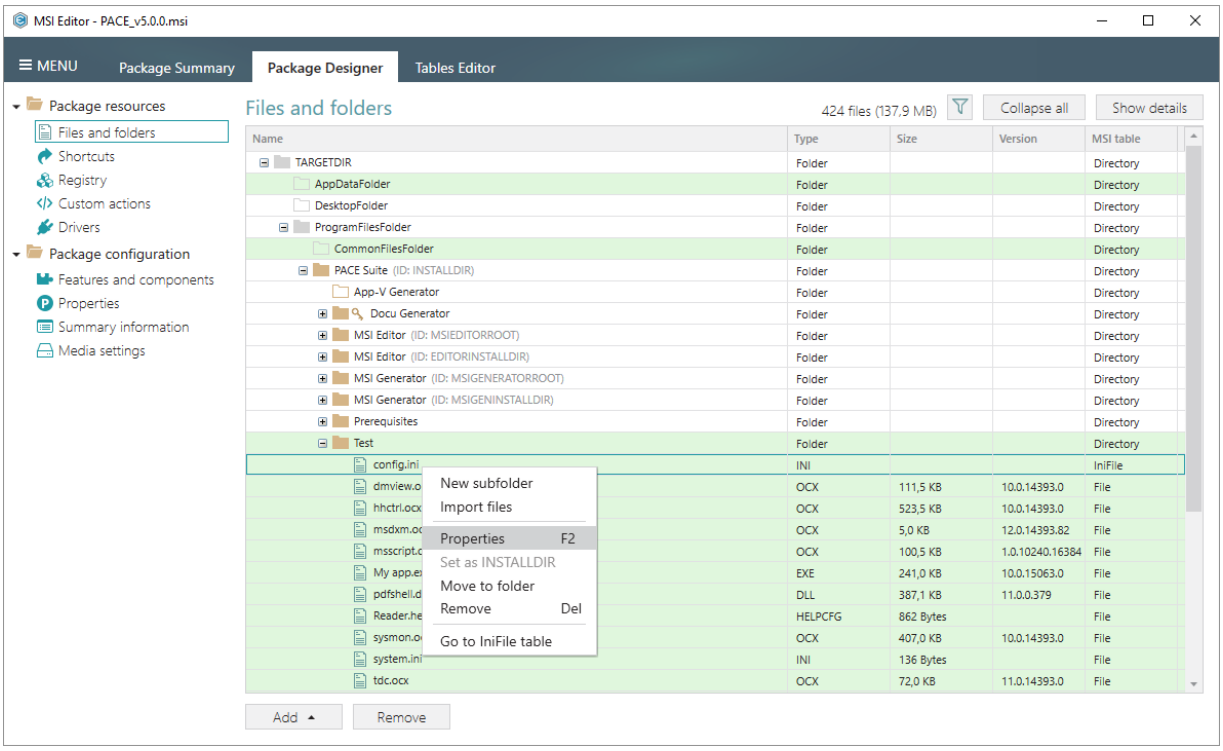
Edit a name, a destination location and a content of the INI file, which has been read from the IniFile table. Note that content of the binary files, which has been read from the File table, could not be modified using the approach, described below.

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

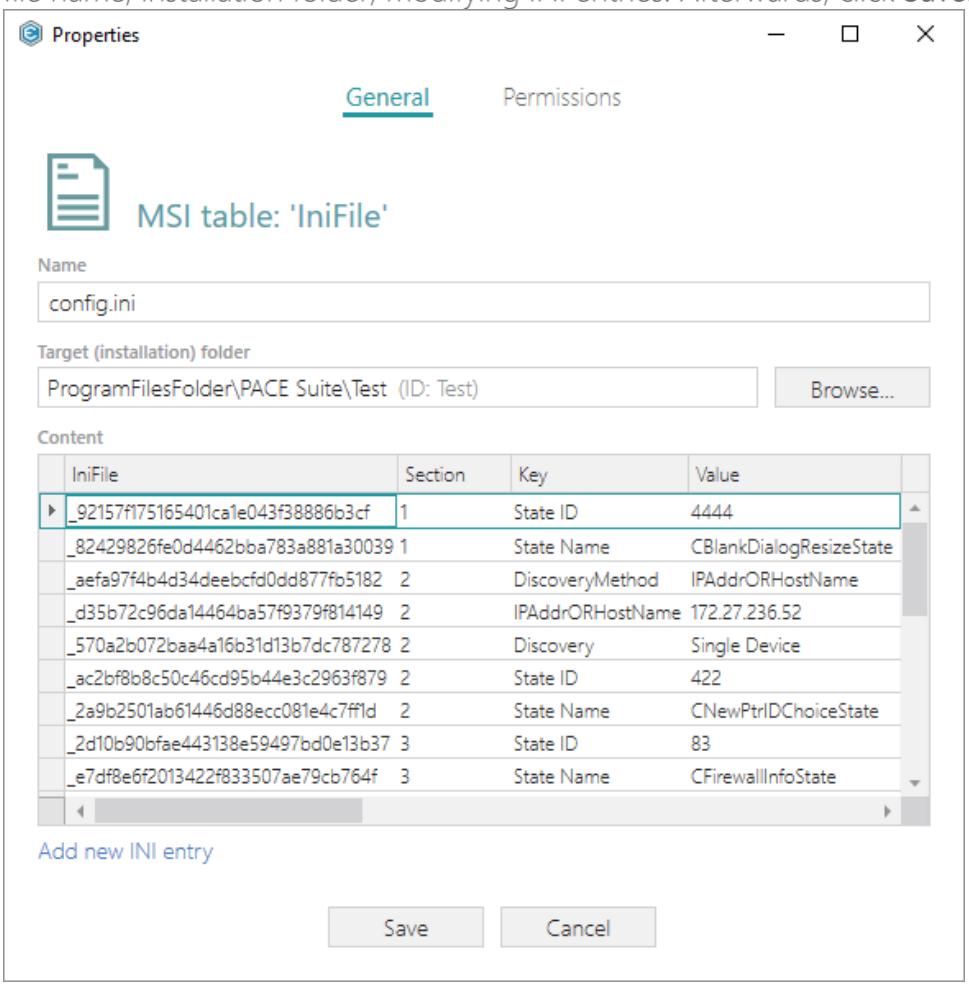



- Locate INI file, which you want to edit, and ensure that this file is located in the IniFile table. Note that only files from IniFile table could be reviewed and modifier because their content is stored in MSI database. Files from the File table are binary files and their content could not be reviewed on the fly. Select Properties from the context menu of the INI file you want

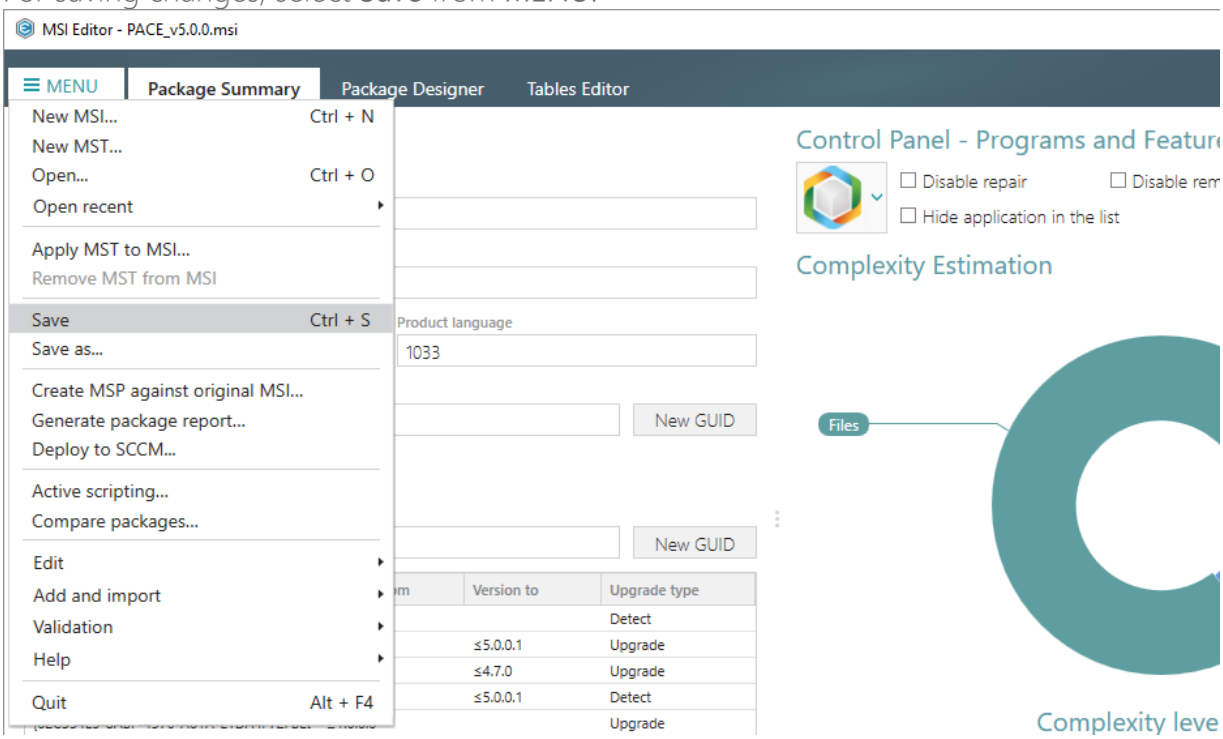
to edit.




- Using the **Properties** window make the necessary changes to the INI file, such as changing file name, installation folder, modifying INI entries. Afterwards, click **Save**.



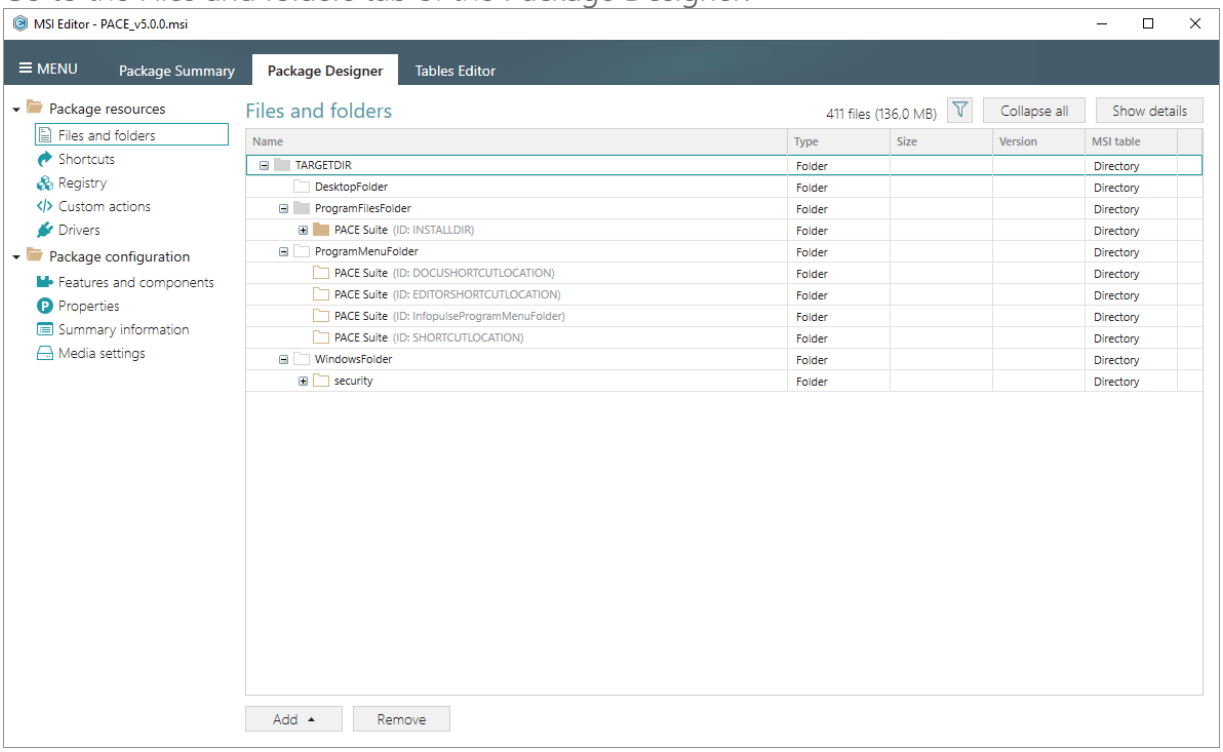
- For saving changes, select Save from MENU.



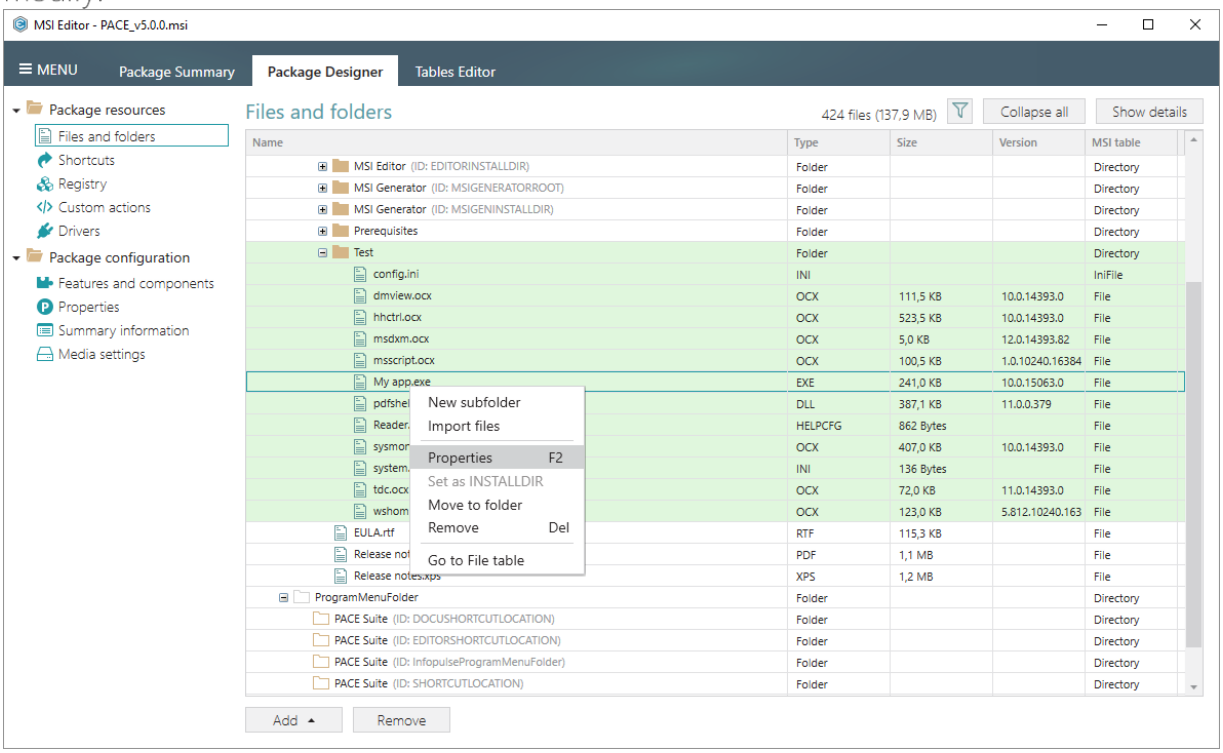

### 3.7.2.10 Edit File Properties

Edit a file name, an identifier, a destination location, a component and many other attributes of the binary file, which has been read from the File table.

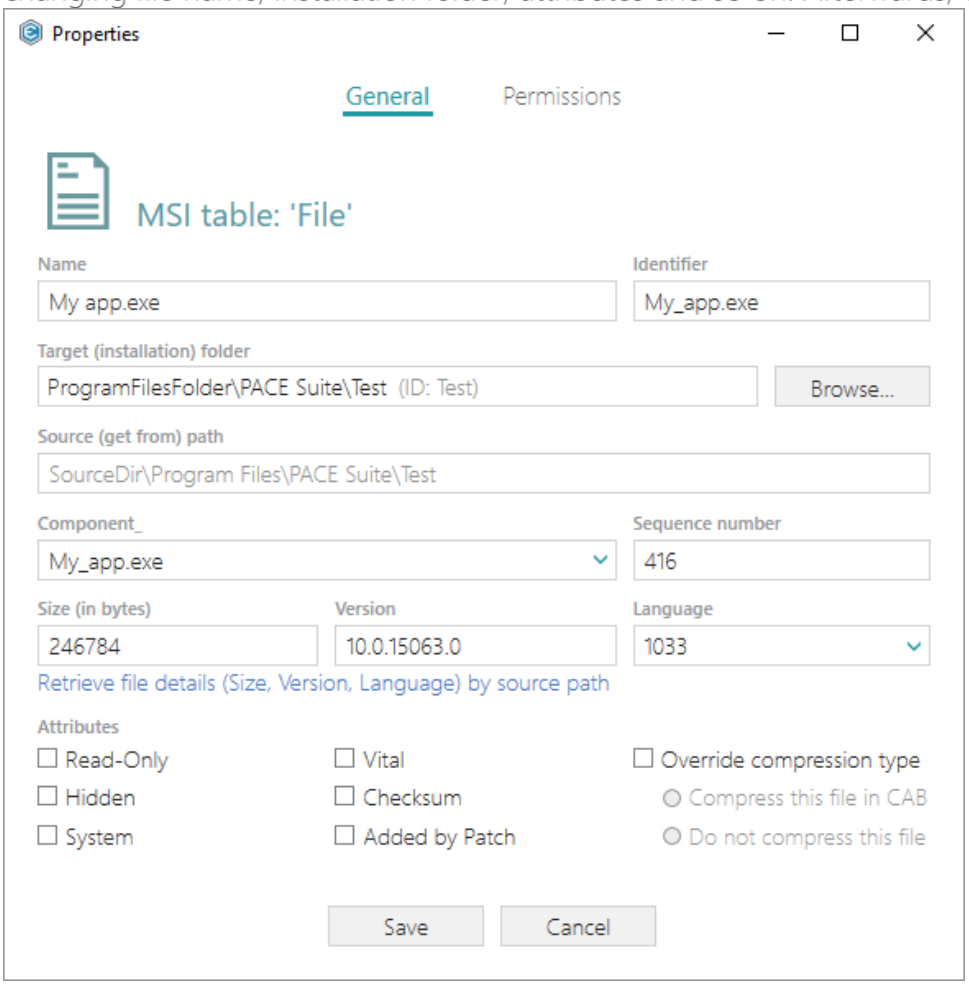
- Go to the Files and folders tab of the Package Designer.



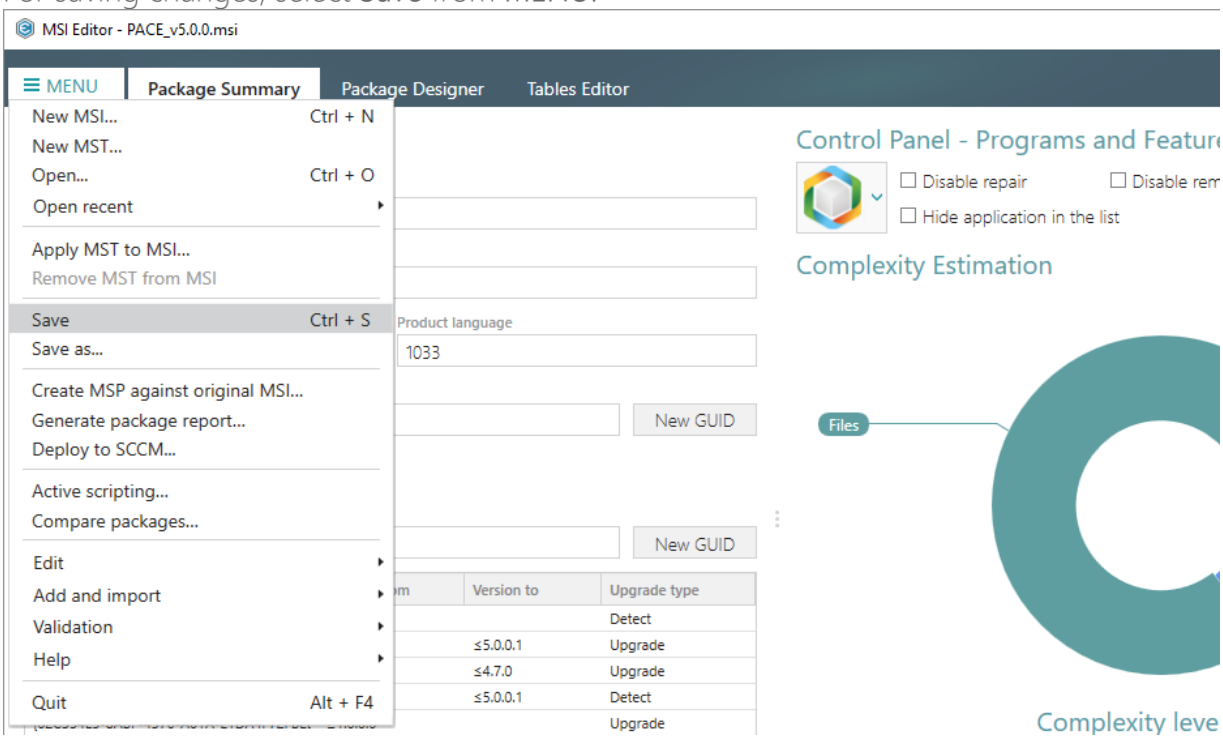
- Select **Properties** from the context menu of a file, details of which you want to review and modify.




- Using the **Properties** window make the necessary changes to the file details, such as changing file name, installation folder, attributes and so on. Afterwards, click **Save**.



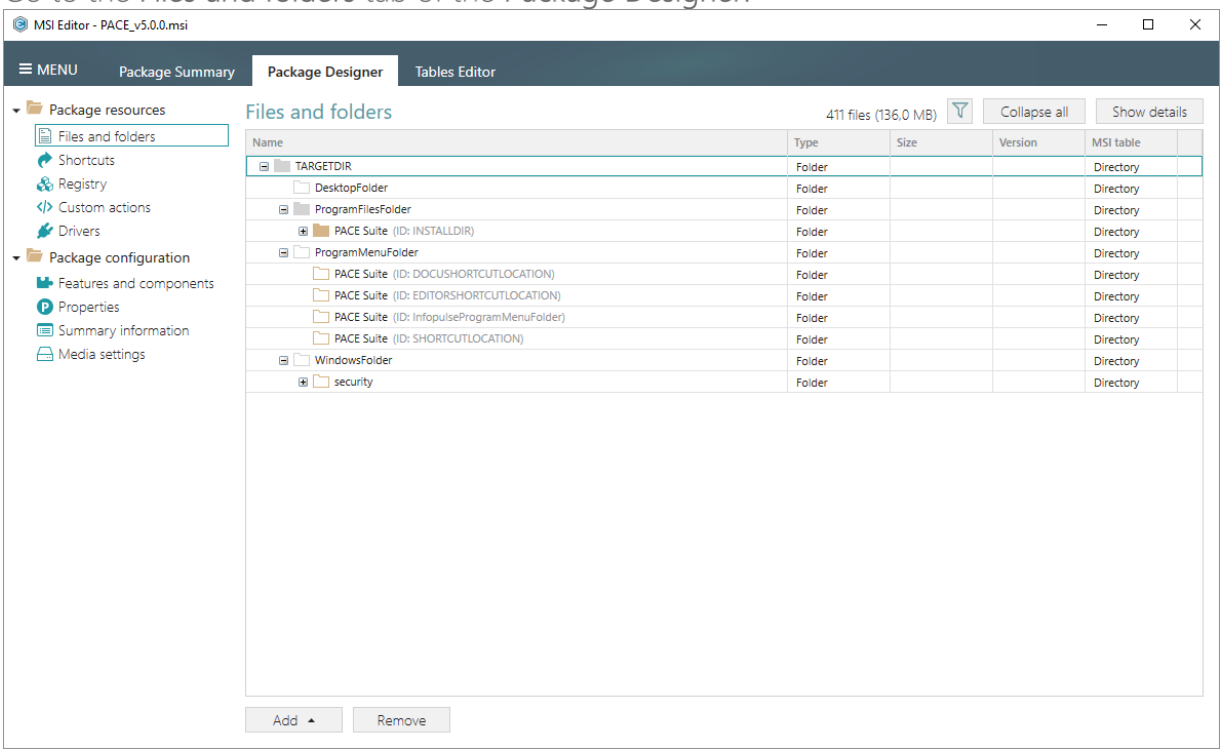
- For saving changes, select Save from MENU.



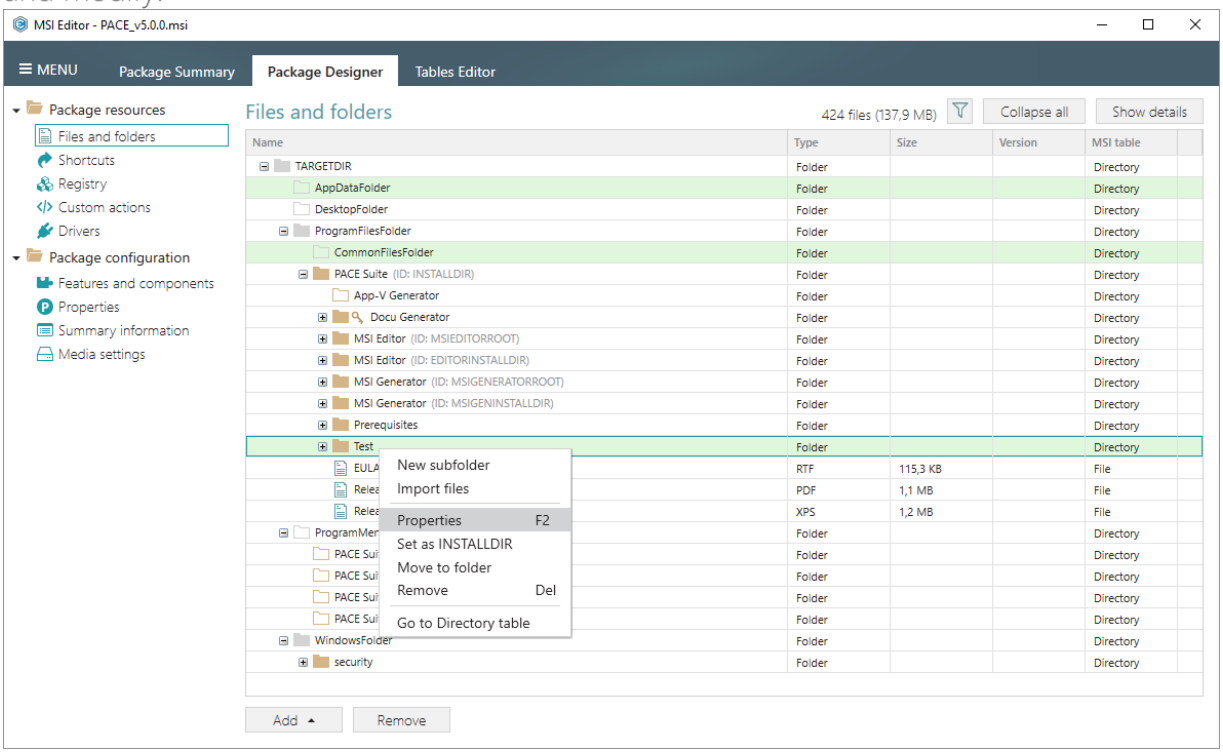

### 3.7.2.11 Edit Folder Properties

Edit a folder target and source name, an identifier, a parent folder.

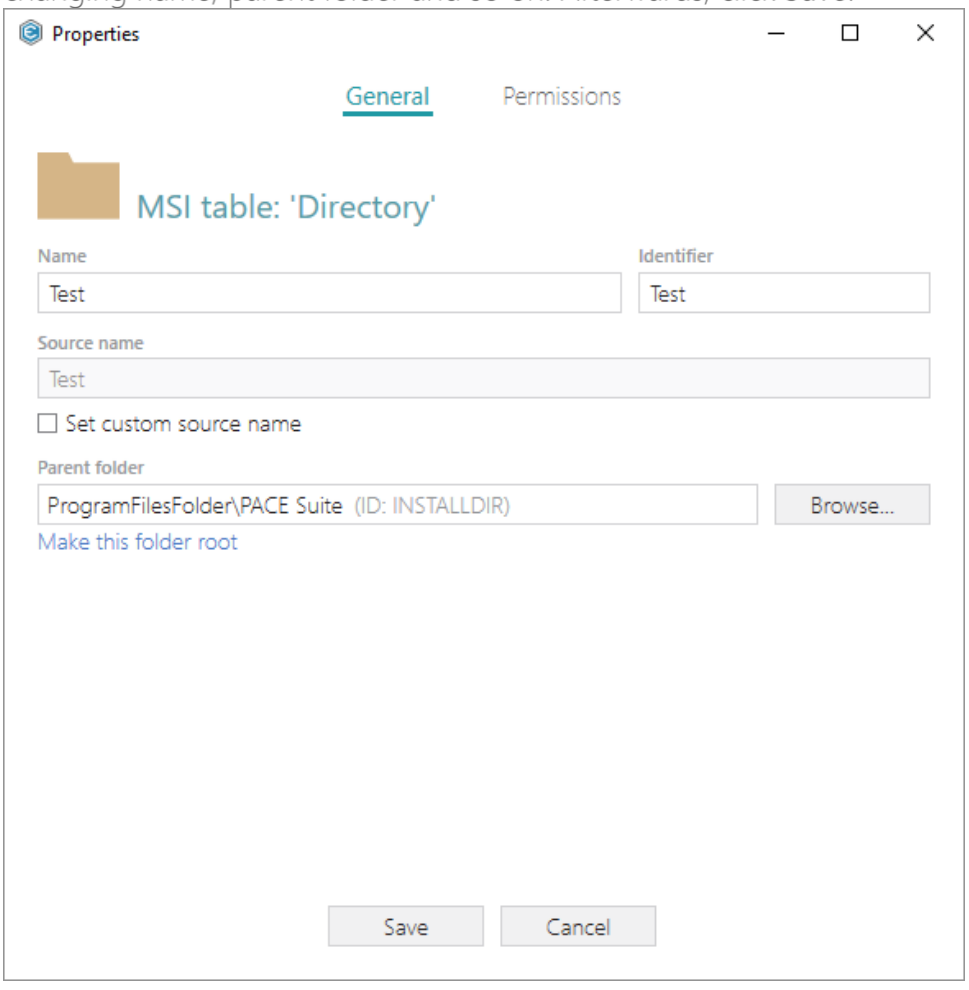
- Go to the Files and folders tab of the Package Designer.



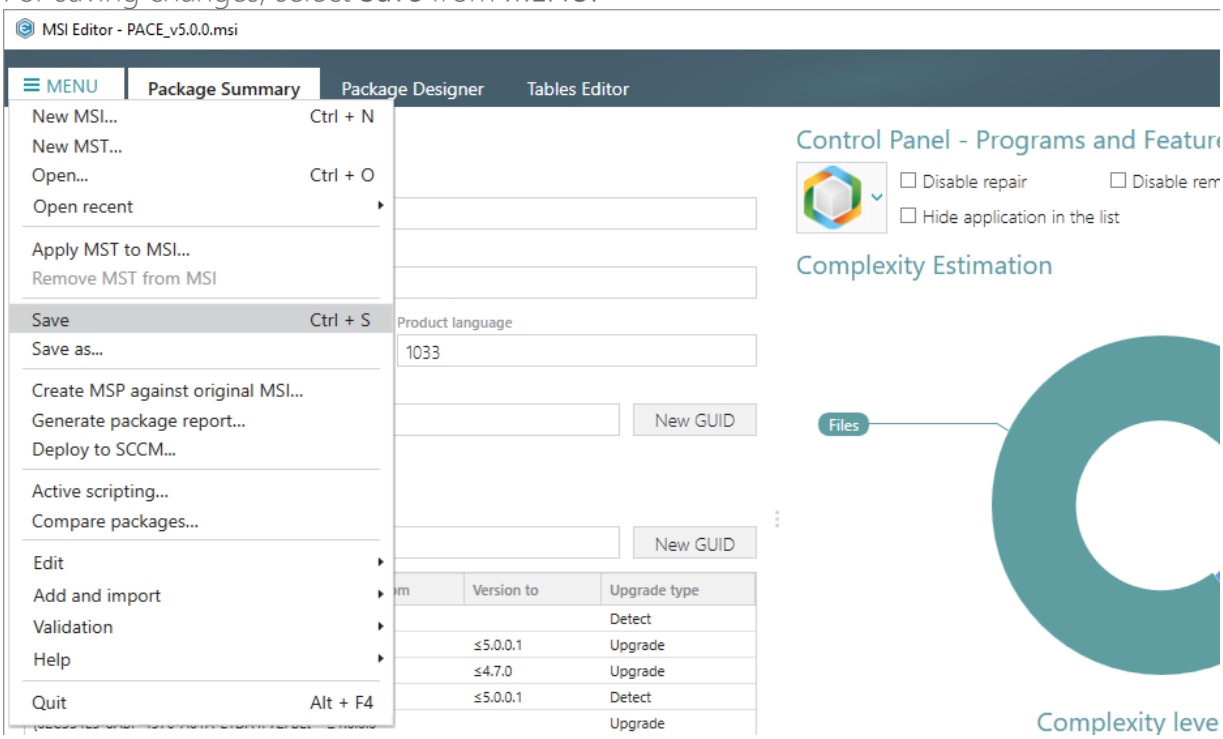
- Select **Properties** from the context menu of a folder, details of which you want to review and modify.




- Using the **Properties** window, make the necessary changes to the folder details, such as changing name, parent folder and so on. Afterwards, click **Save**.



- For saving changes, select Save from MENU.




## 3.7.3 Shortcuts

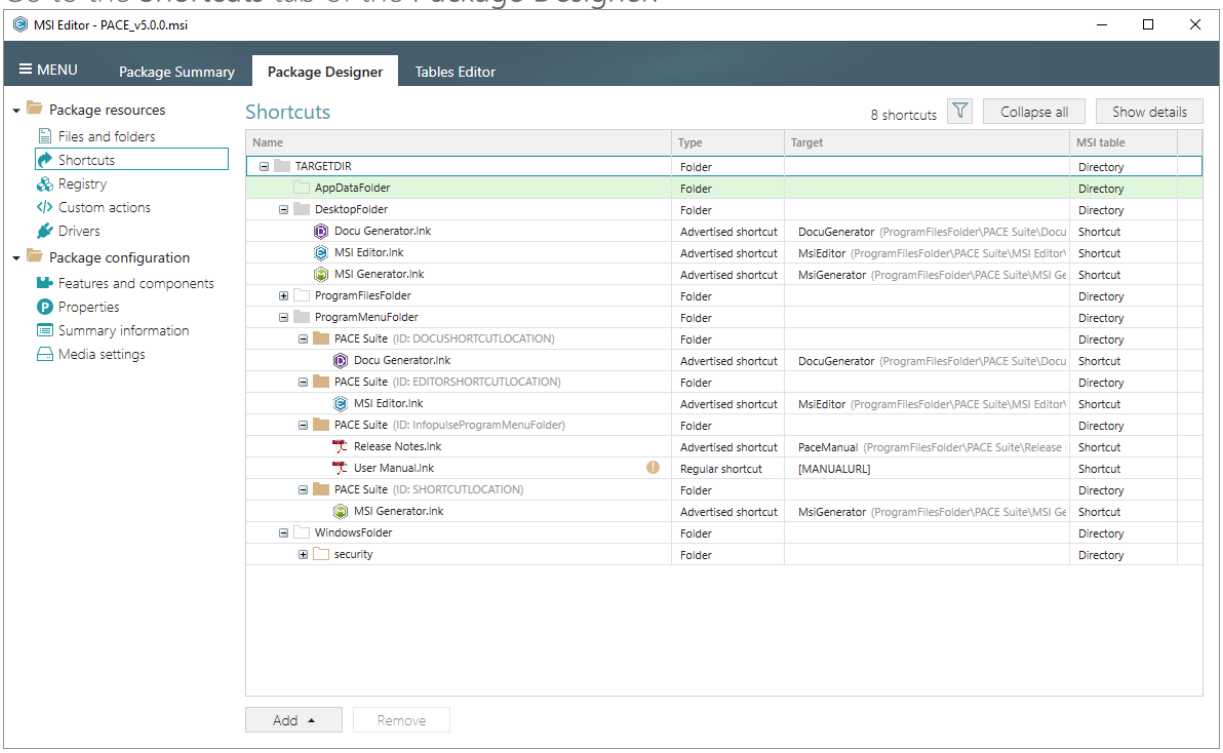
Choose a scenario that better suits your needs:

- Add Shortcut, described in section 3.7.3.1
  Add a regular or an advertised shortcut that points to a file, folder or URL.
- Import Shortcuts, described in section 3.7.3.2
  Import shortcuts from the file system to the package.
- Move Shortcut, described in section 3.7.3.3
  Move a shortcut to another folder.
- Edit Shortcut Properties, described in section 3.7.3.4
  Edit shortcut details such as icon, name, installation folder, shortcut type, arguments, description and so on.

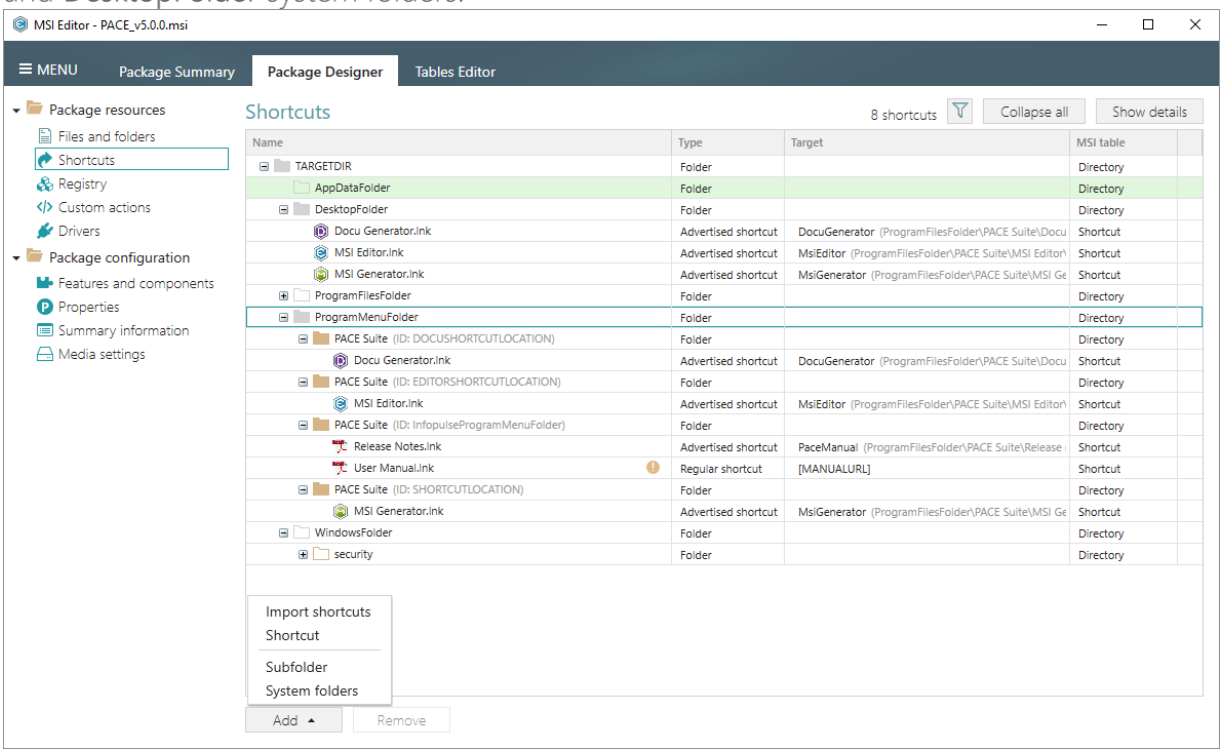
### 3.7.3.1 Add Shortcut

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

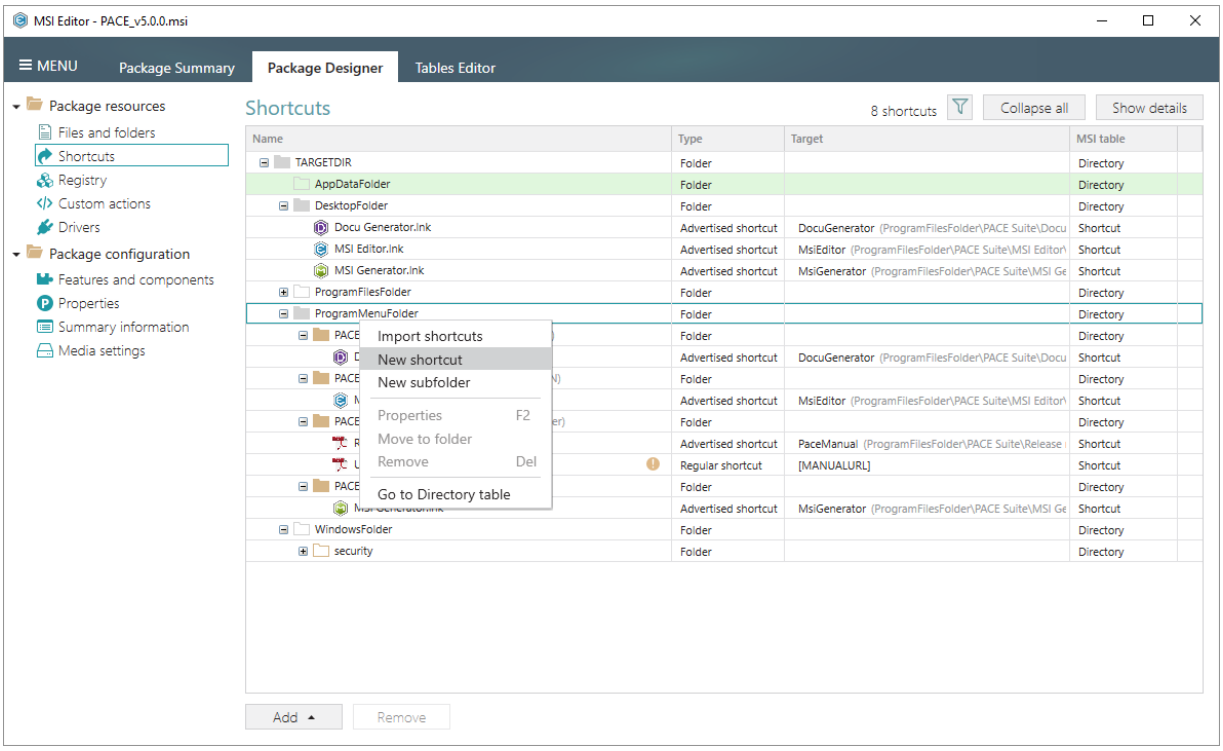
- Go to the **Shortcuts** tab of the **Package Designer**.




- Check that the folders tree contains a folder, where you want to create this shortcut. If needed, add the missing system folders or subfolders clicking **Add -> System folders** or **Add -> Subfolder**. Usually, application shortcuts are installed to the **ProgramMenuFolder** and **DesktopFolder** system folders.

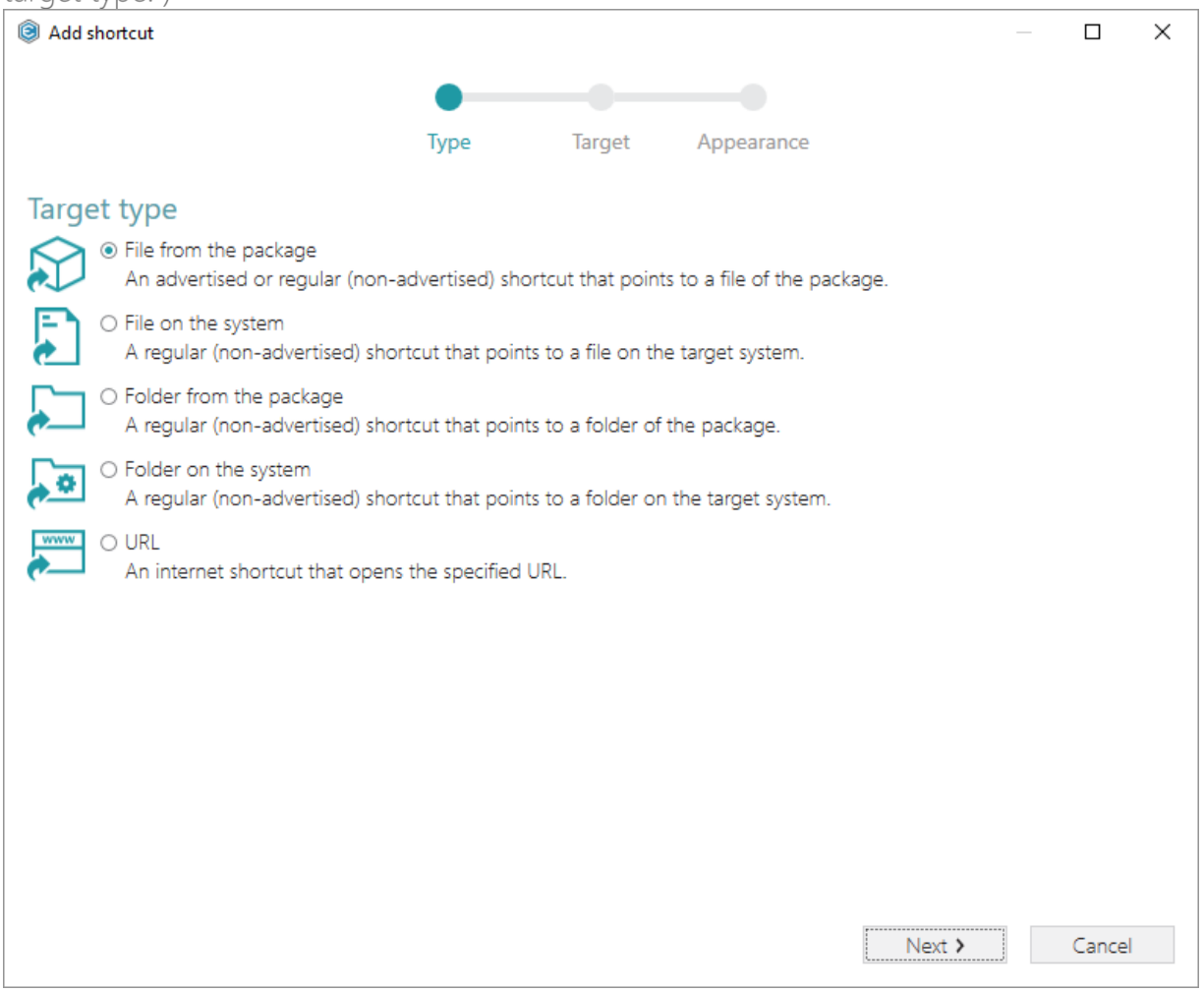


- Select **New shortcut** from the context menu of a folder, where you want to create a shortcut.

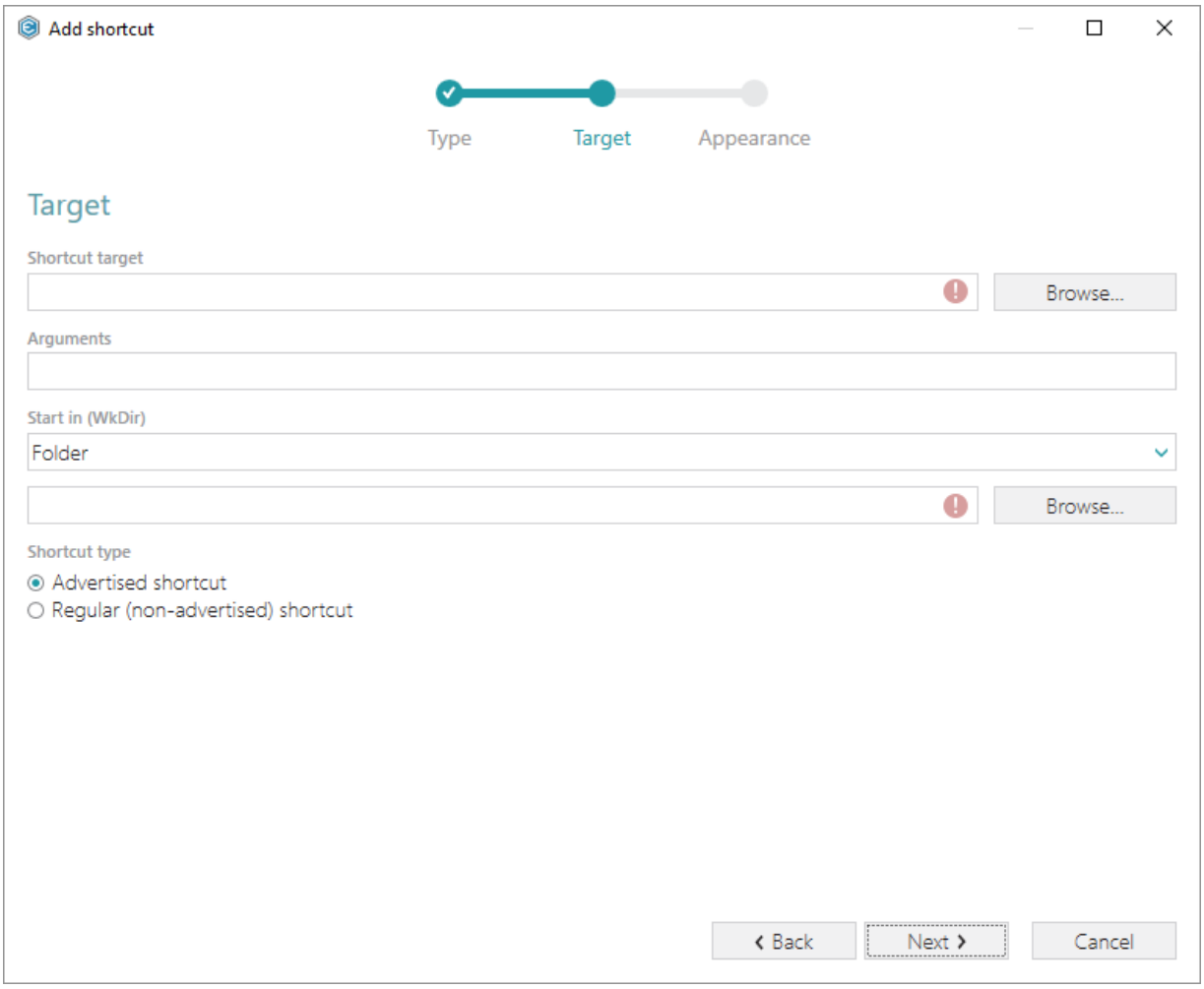



- Select a type of the shortcut target and click **Next**.
(The following instructions describe creation of the shortcut of the '**File from the package**'

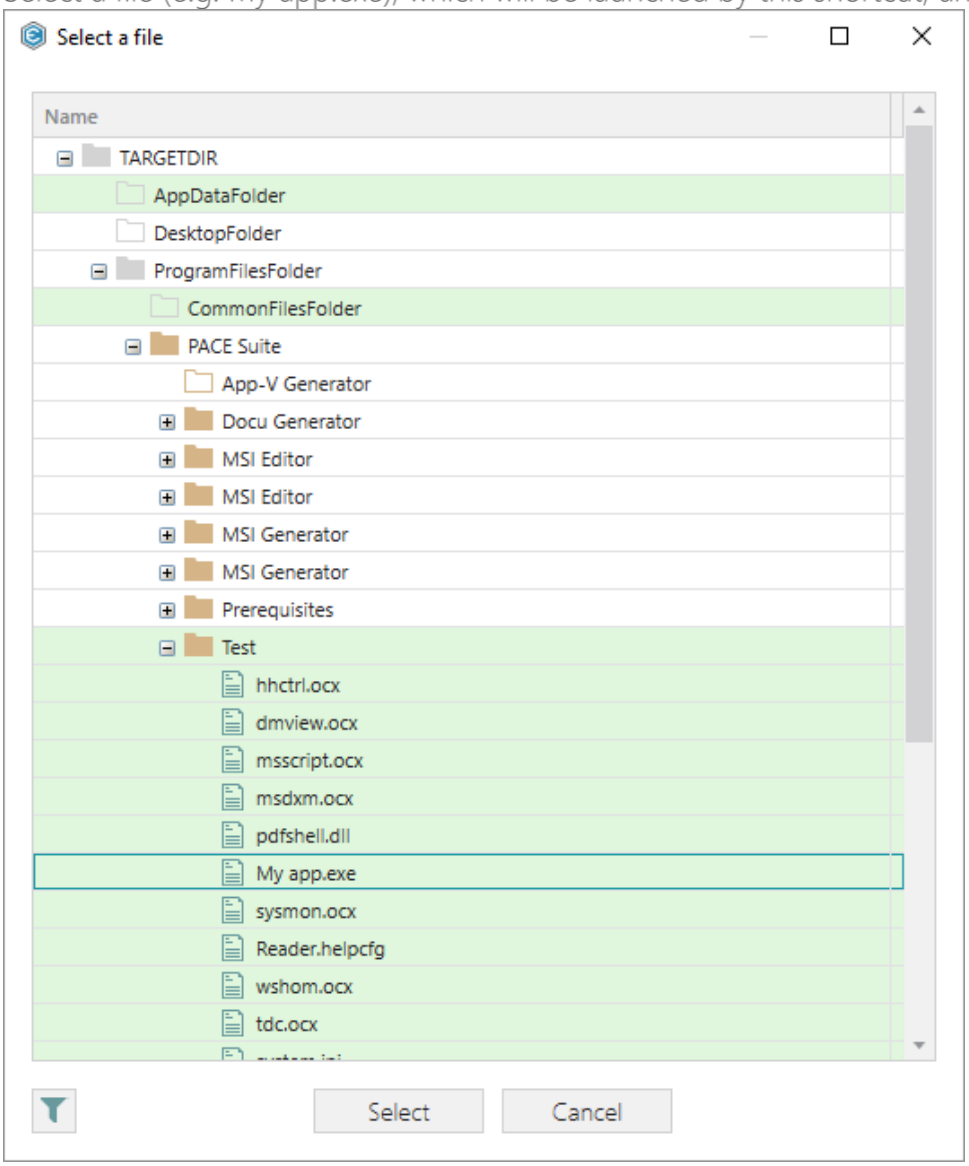
target type. )


Add shortcut
Type
Target
Appearance
Target type
File from the package
An advertised or regular (non-advertised) shortcut that points to a file of the package.
File on the system
A regular (non-advertised) shortcut that points to a file on the target system.
Folder from the package
A regular (non-advertised) shortcut that points to a folder of the package.
Folder on the system
A regular (non-advertised) shortcut that points to a folder on the target system.
URL
An internet shortcut that opens the specified URL.
Next ›
Cancel

- On the second step, click **Browse**, located next to the **Shorcut target** field, to set a target of the shortcut.

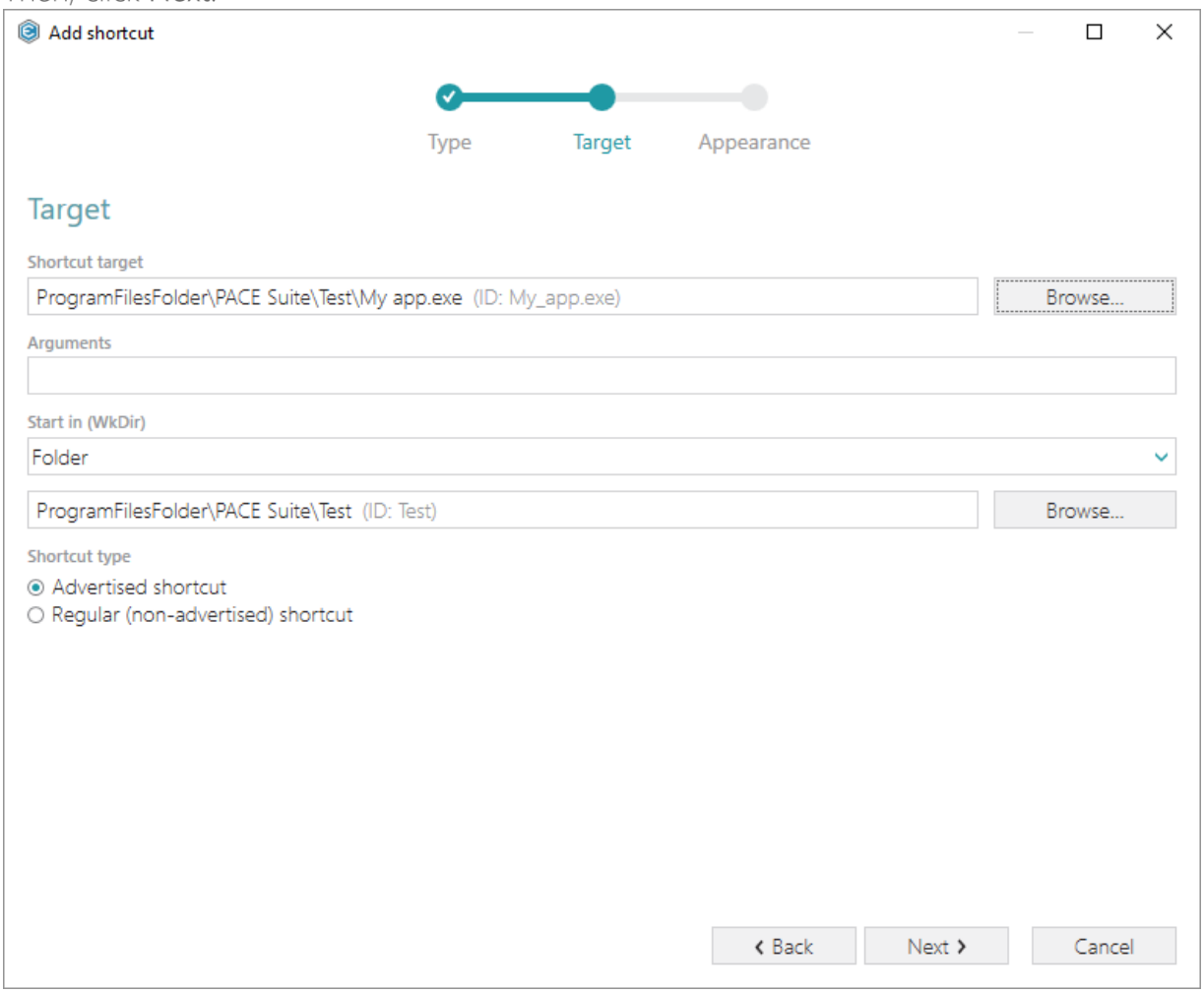


- Select a file (e.g. My app.exe), which will be launched by this shortcut, and click **Select**.

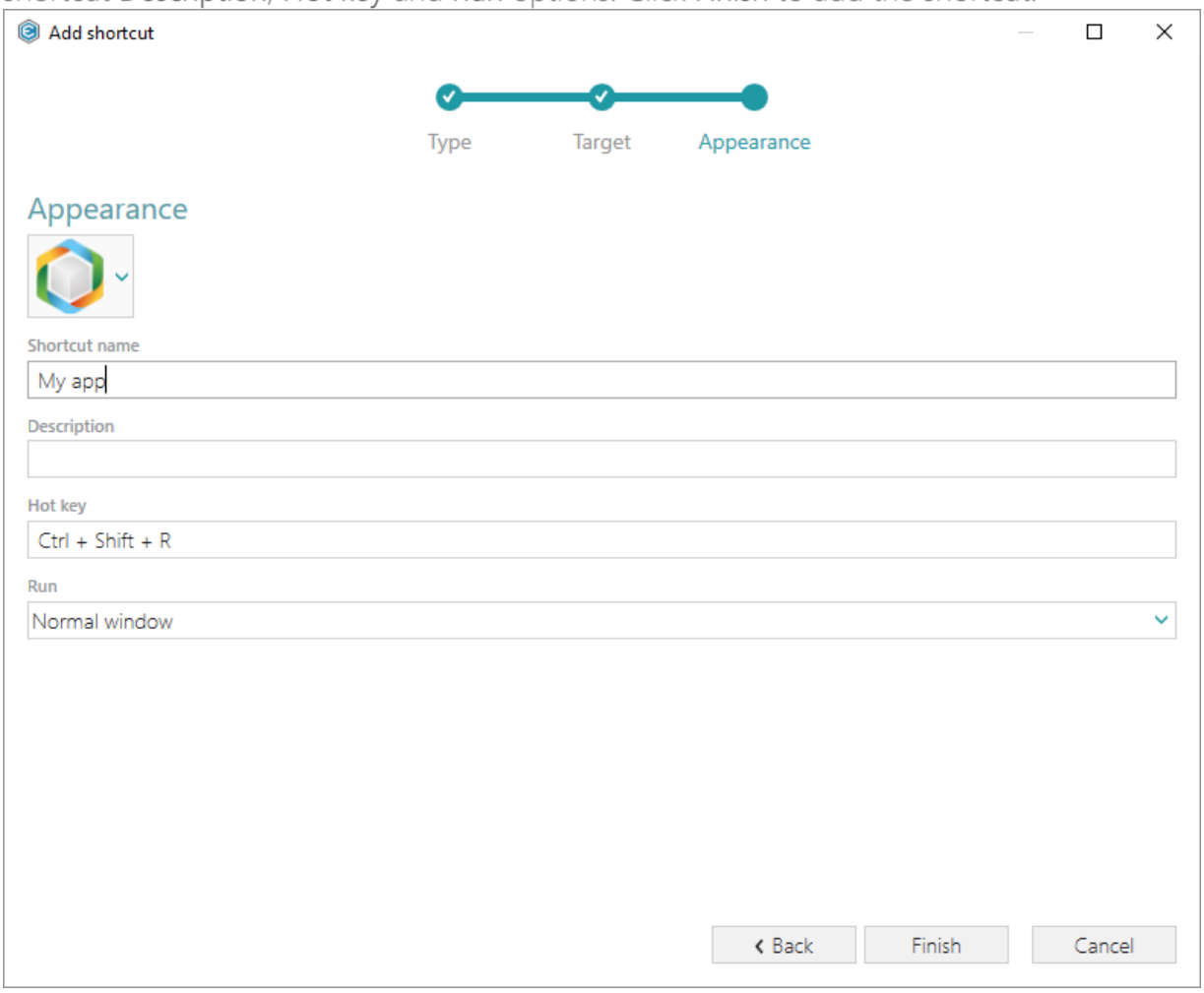



- If needed, enter a shortcut **Arguments**, change the **Start in** path (select a folder or a property), and select a **Shortcut type**. Note that only the Advertised shortcut supports advertisement and repair. This means that if the executable to which the shortcut is pointing is not there then windows installer will repair the application and replace the missing files.

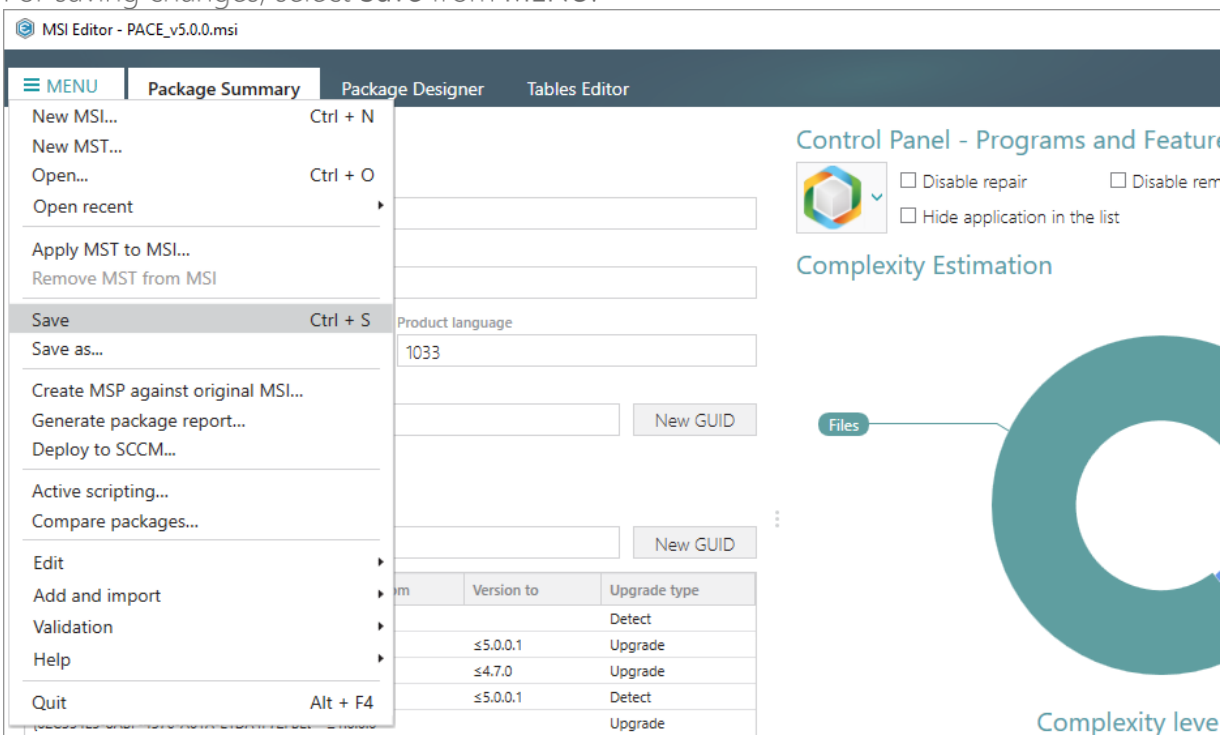
Then, click **Next**.



- 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.



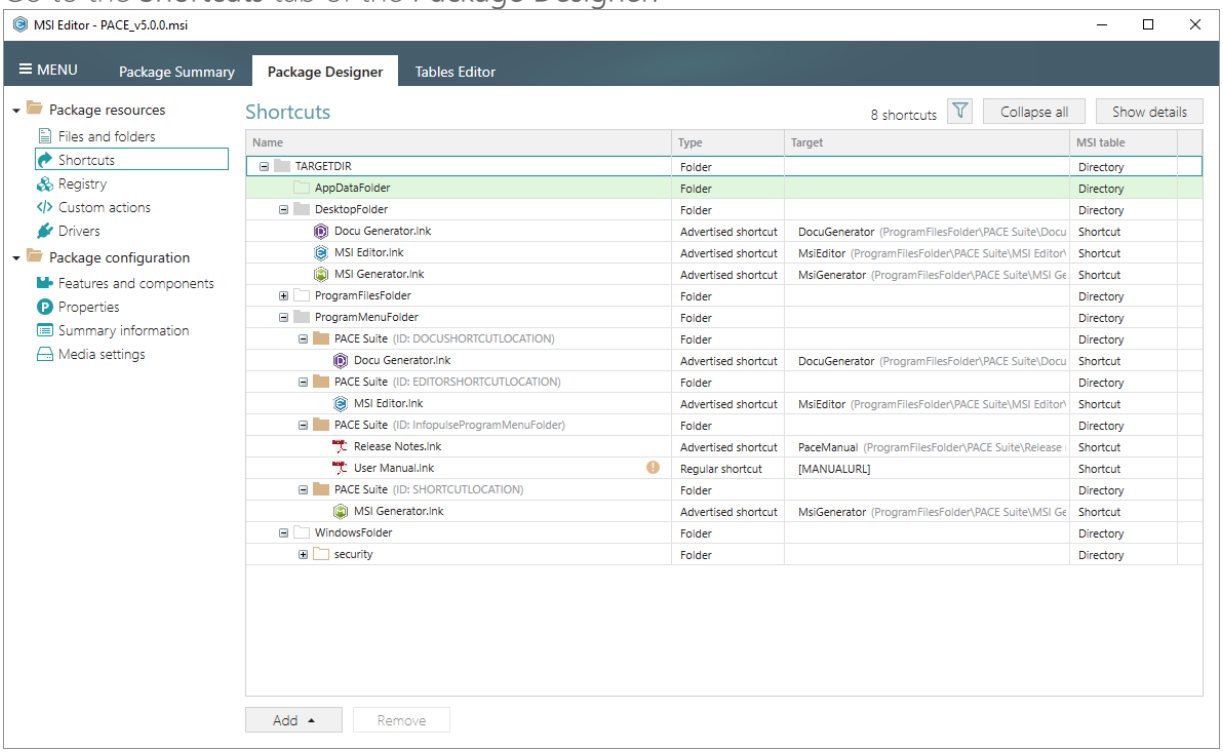

- For saving changes, select **Save** from **MENU**.



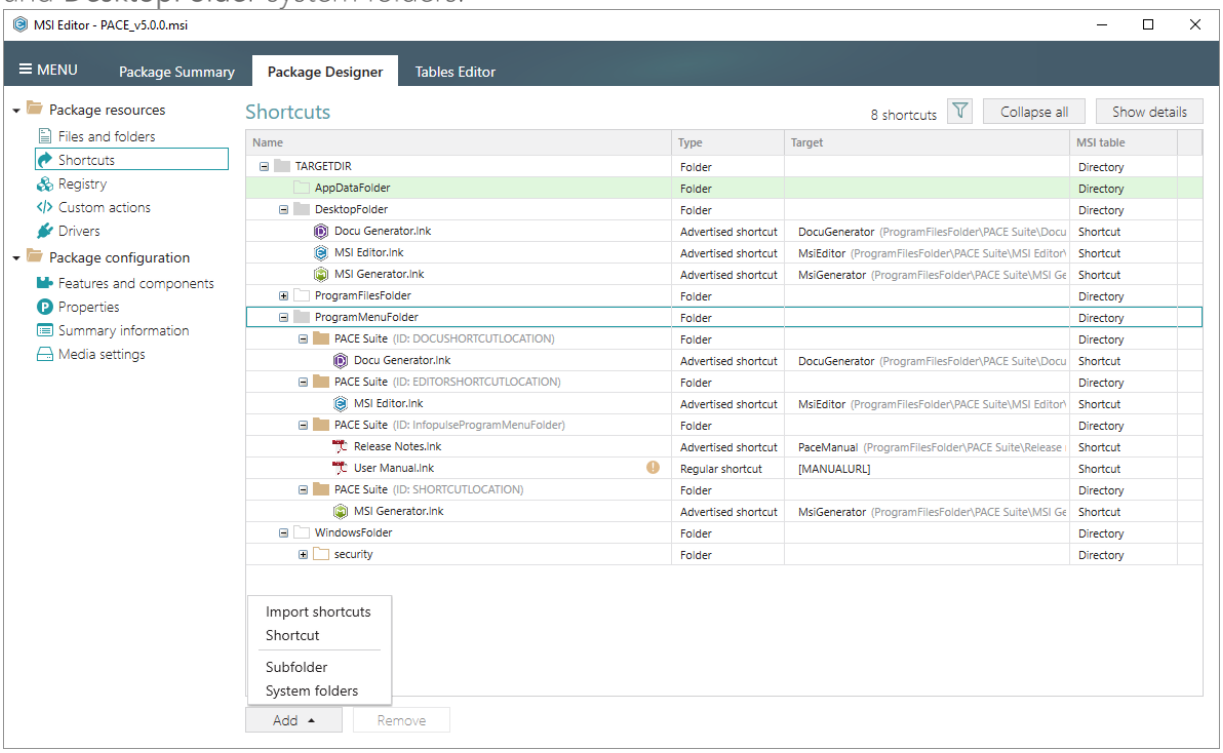
### 3.7.3.2 Import Shortcuts

Import shortcuts from the file system to the package.

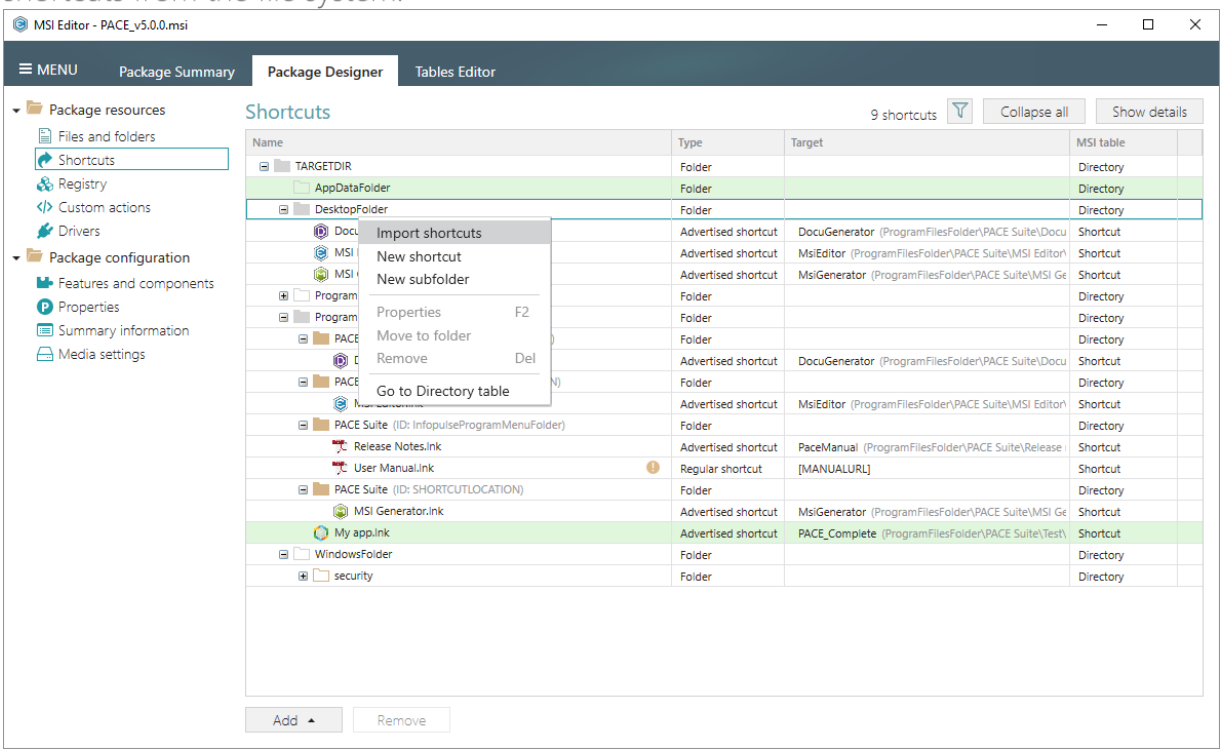
- Go to the **Shortcuts** tab of the **Package Designer**.



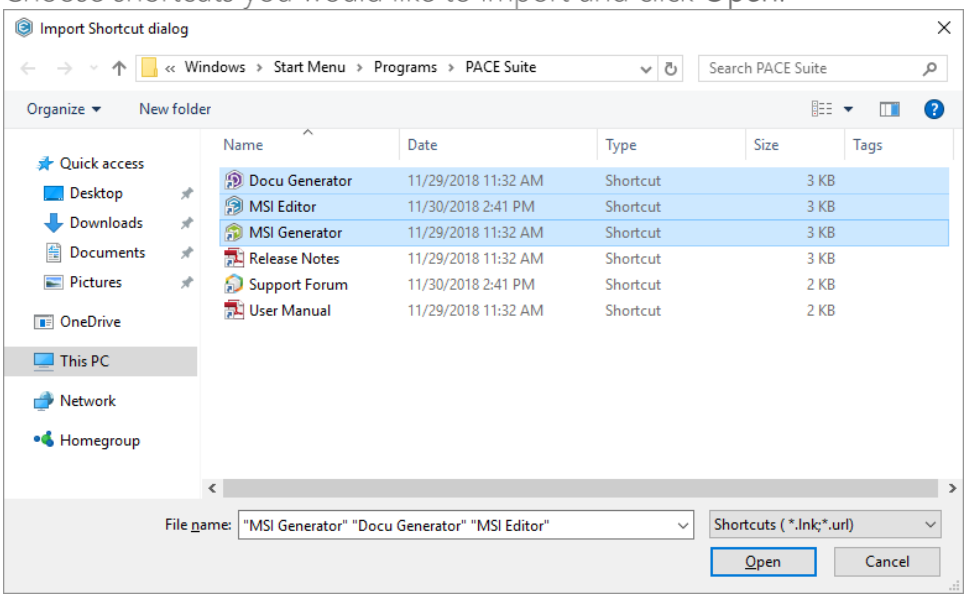

- Check that the folders tree contains a folder, to which you want to import shortcuts. If needed, add the missing system folders or subfolders clicking **Add -> System folders** or **Add -> Subfolder**. Usually, application shortcuts are installed to the **ProgramMenuFolder** and **DesktopFolder** system folders.



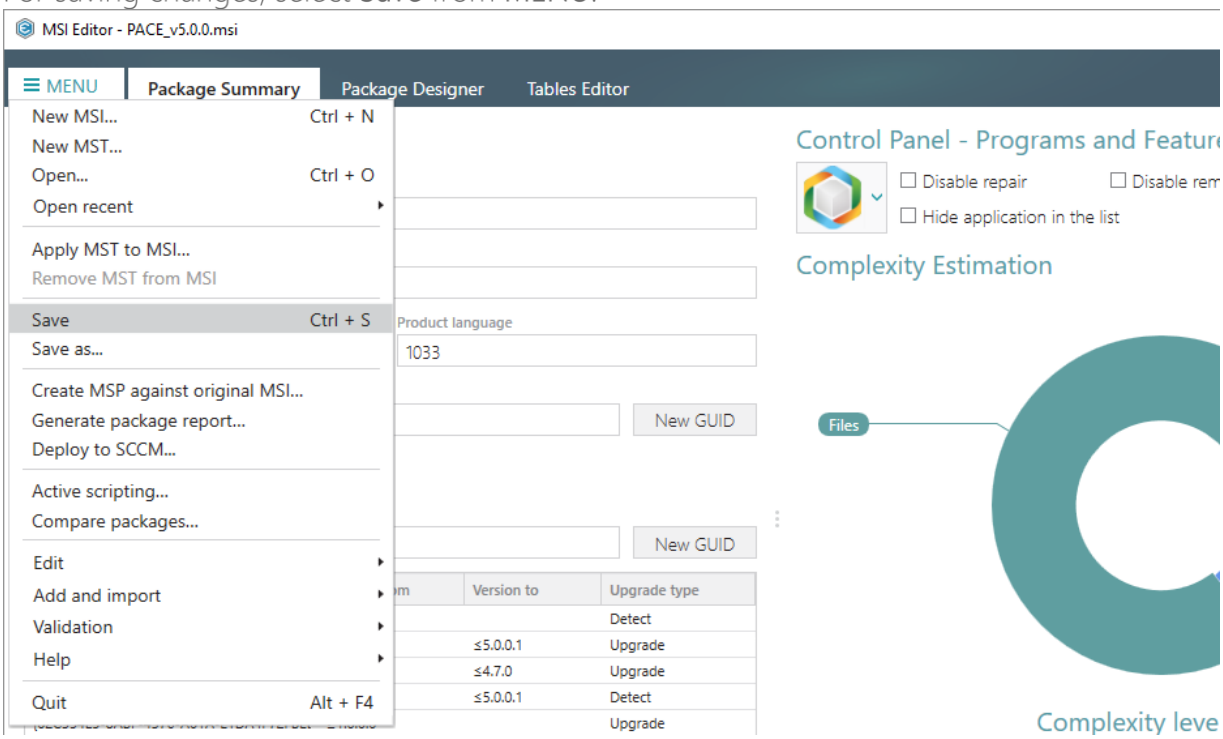
- Select **Import shortcuts** from the context menu of a folder, to which you want to import shortcuts from the file system.




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



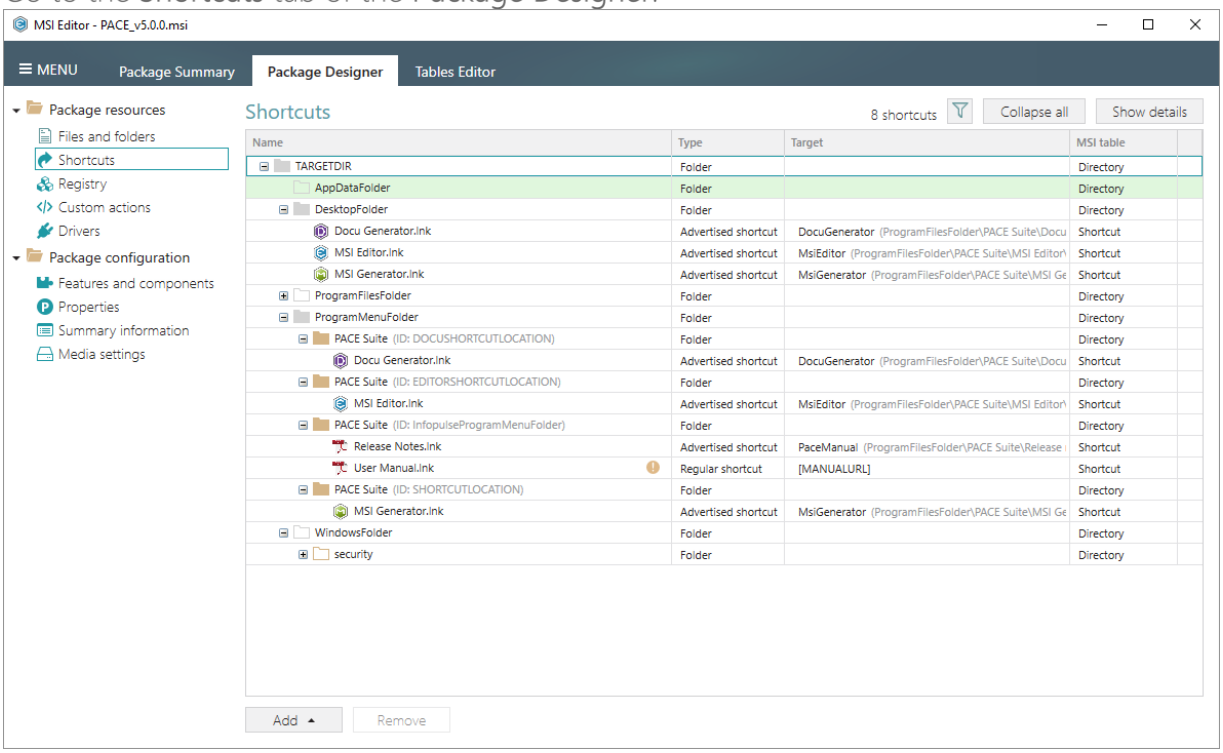
- For saving changes, select Save from MENU.



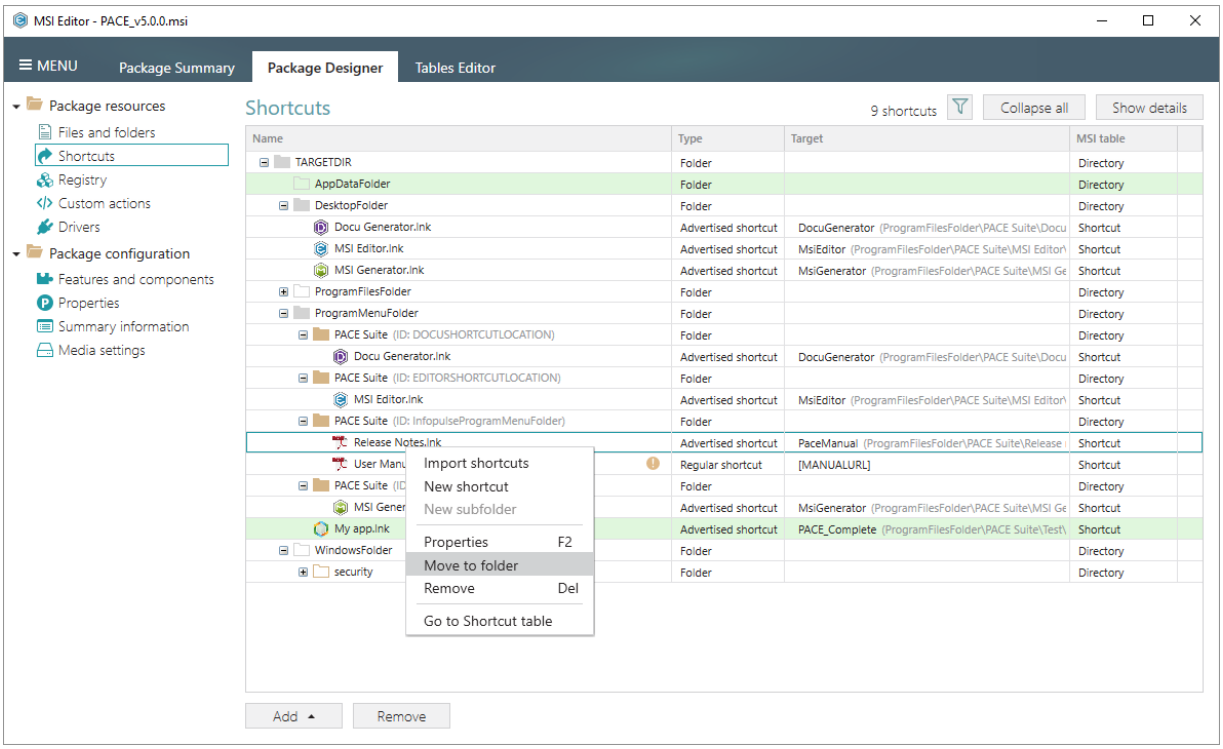

### 3.7.3.3 Move Shortcut

Move a shortcut to another folder.

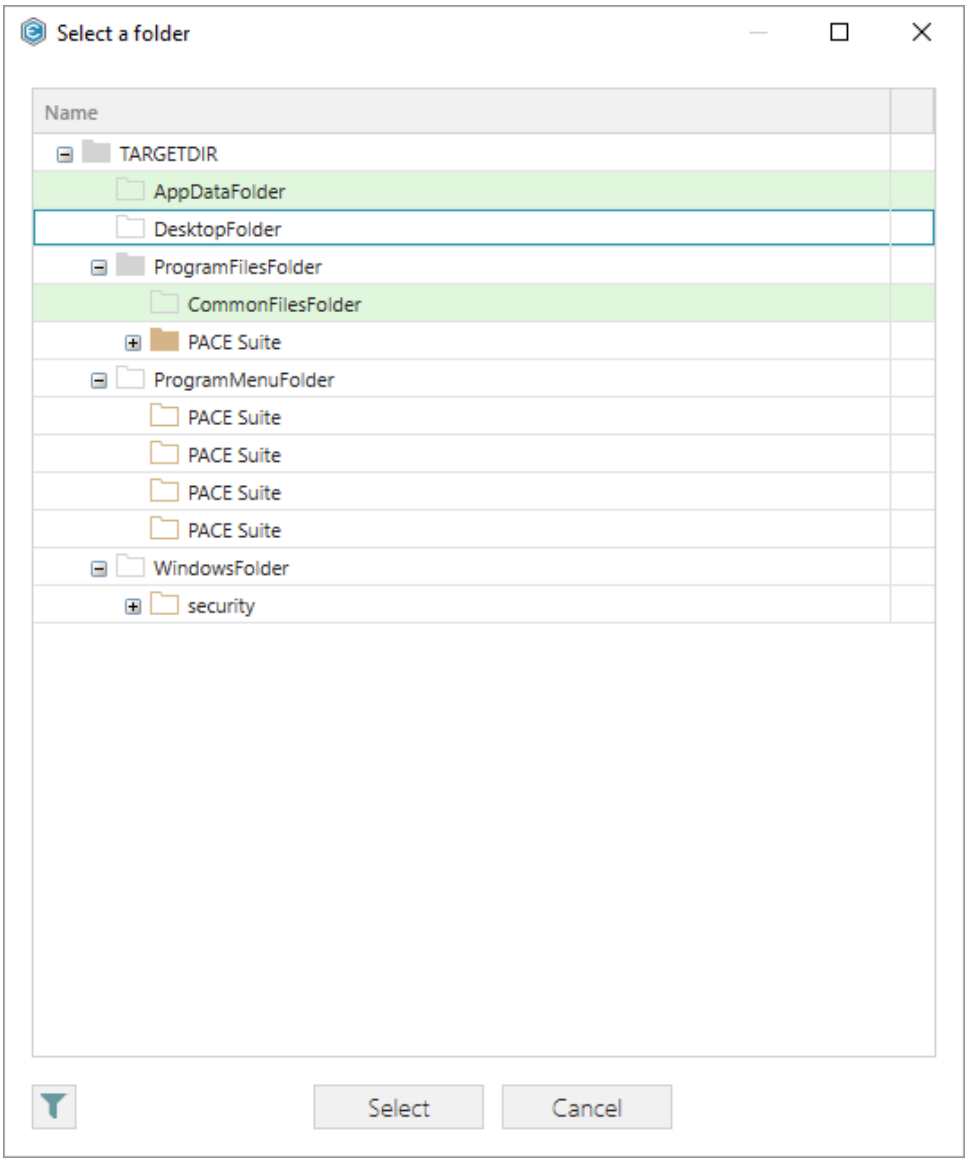
- Go to the Shortcuts tab of the Package Designer.



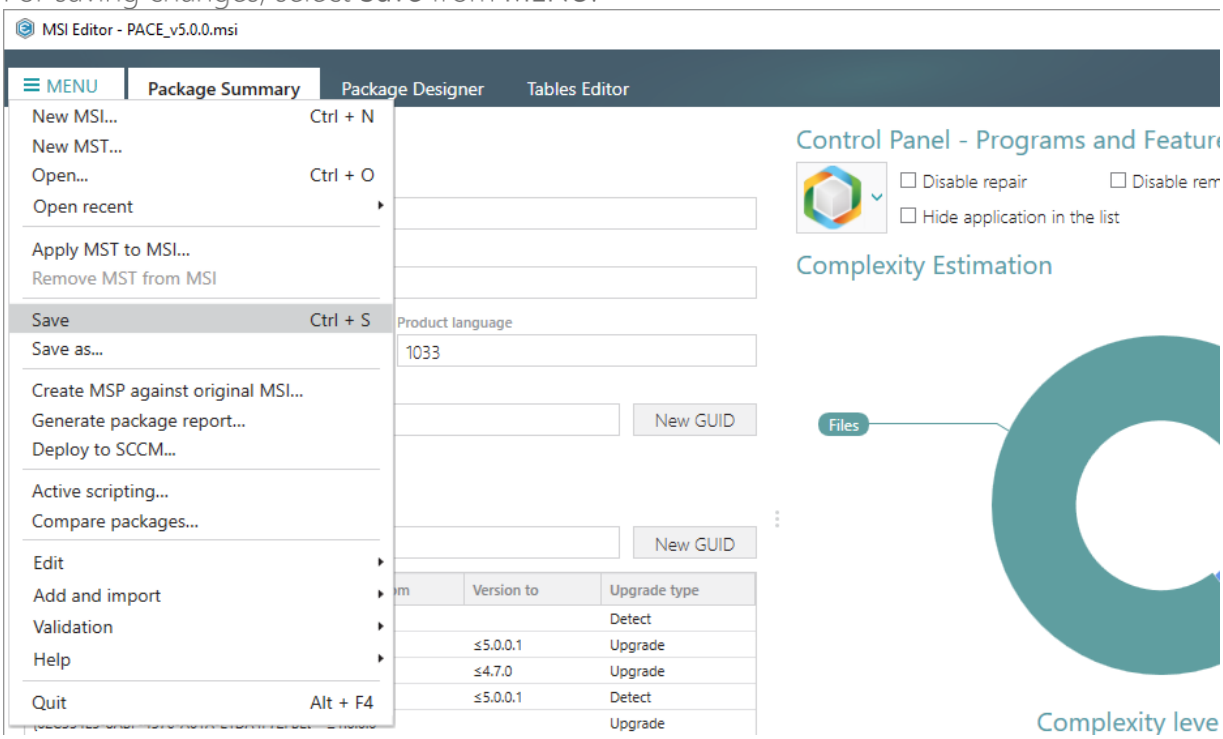
- Select **Move to folder** from the context menu of a shortcut, which you want to move to another folder.



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



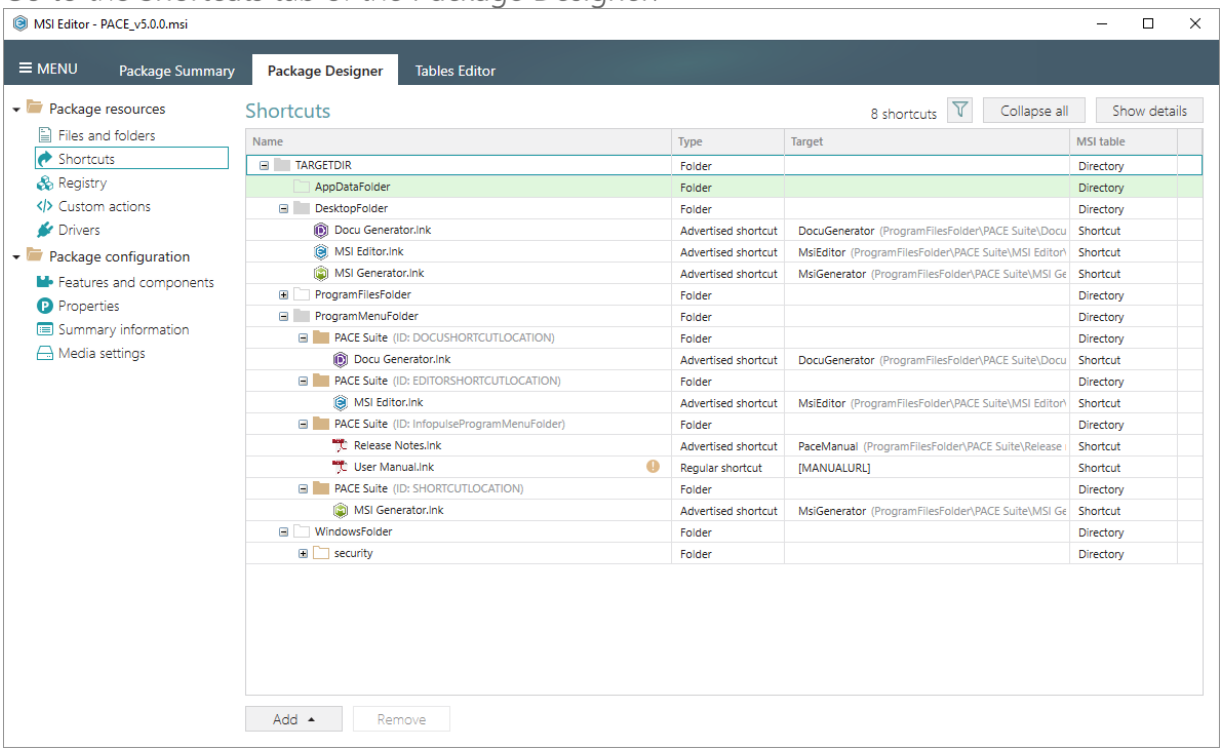
- For saving changes, select Save from MENU.



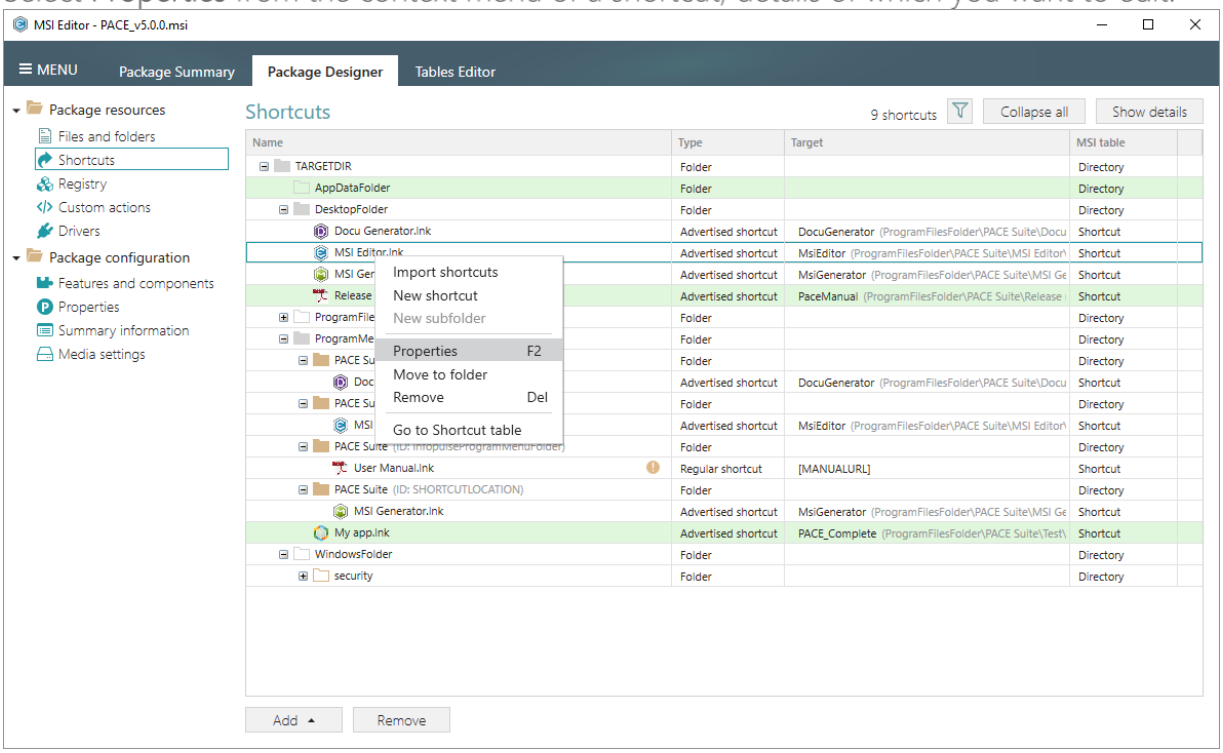

### 3.7.3.4 Edit Shortcut Properties

Edit shortcut details such as icon, name, installation folder, shortcut type, arguments, description and so on.

- Go to the Shortcuts tab of the Package Designer.

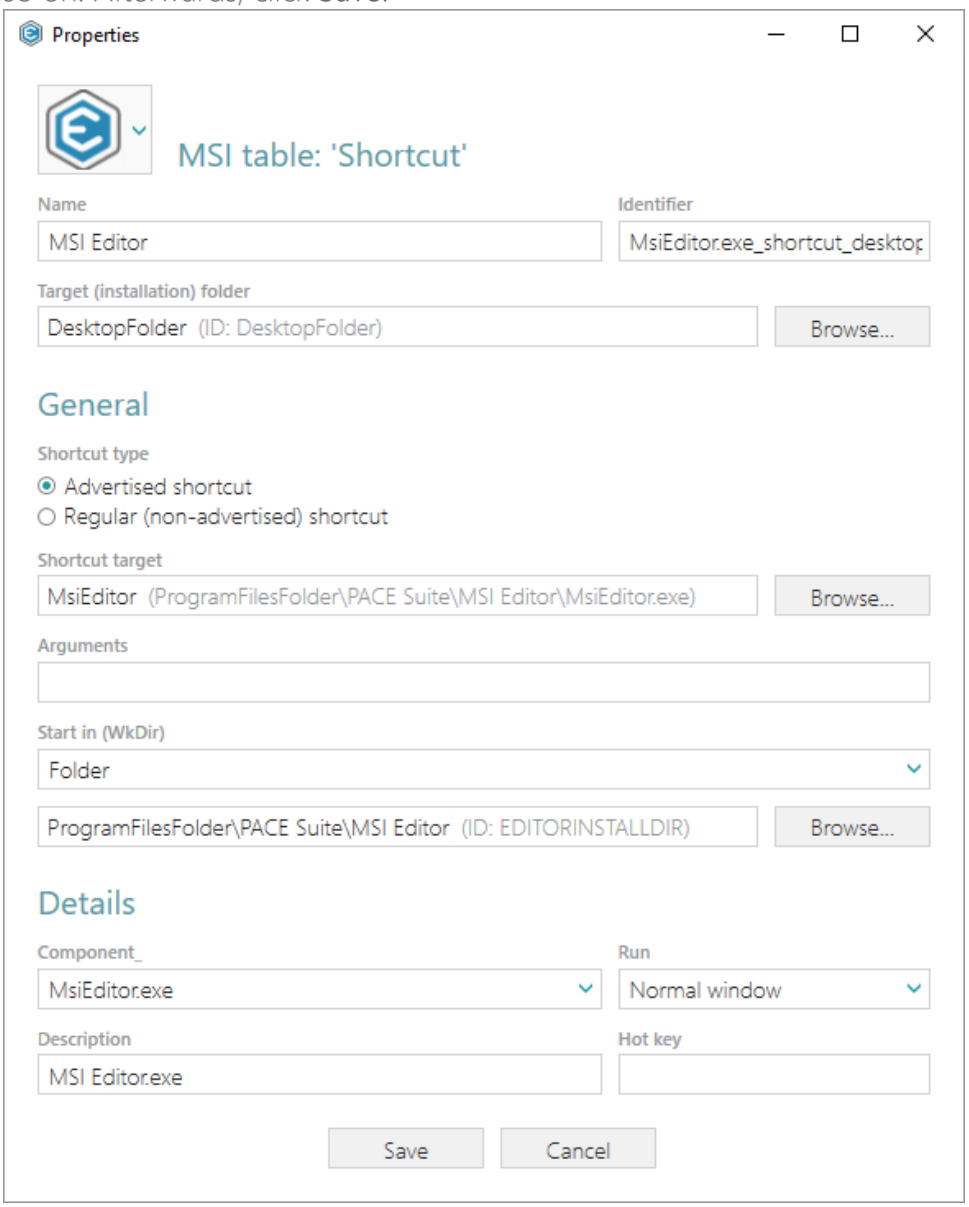


- Select **Properties** from the context menu of a shortcut, details of which you want to edit.

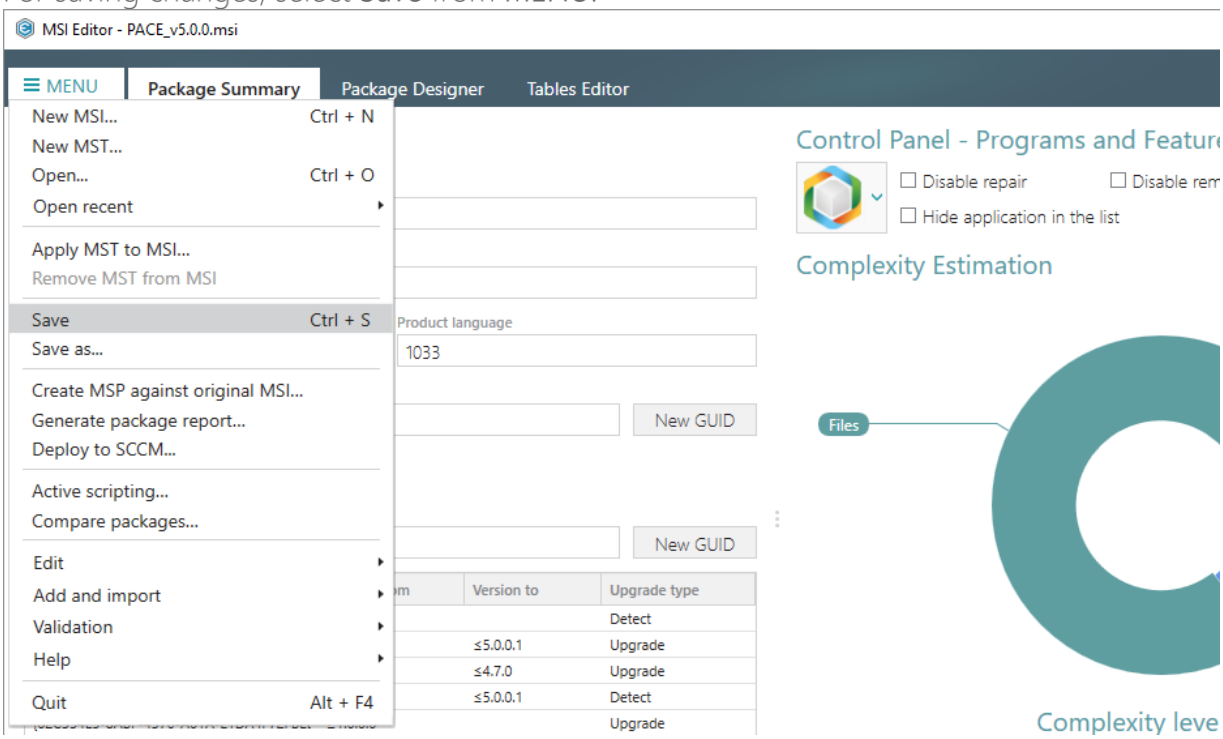



- Using the **Properties** window make the necessary changes to the shortcut details, such as changing icon, name, installation folder, shortcut type, arguments, description, hot key and

so on. Afterwards, click Save.



- For saving changes, select Save from MENU.

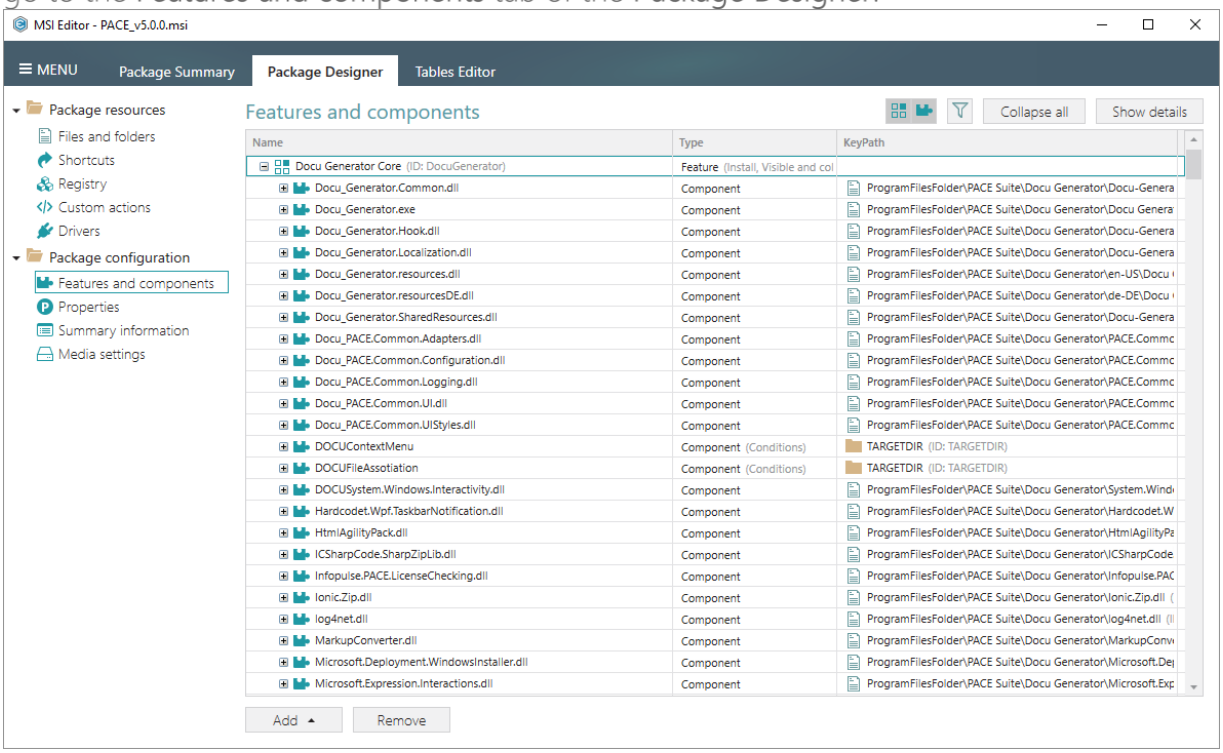



## 3.7.4 Registry

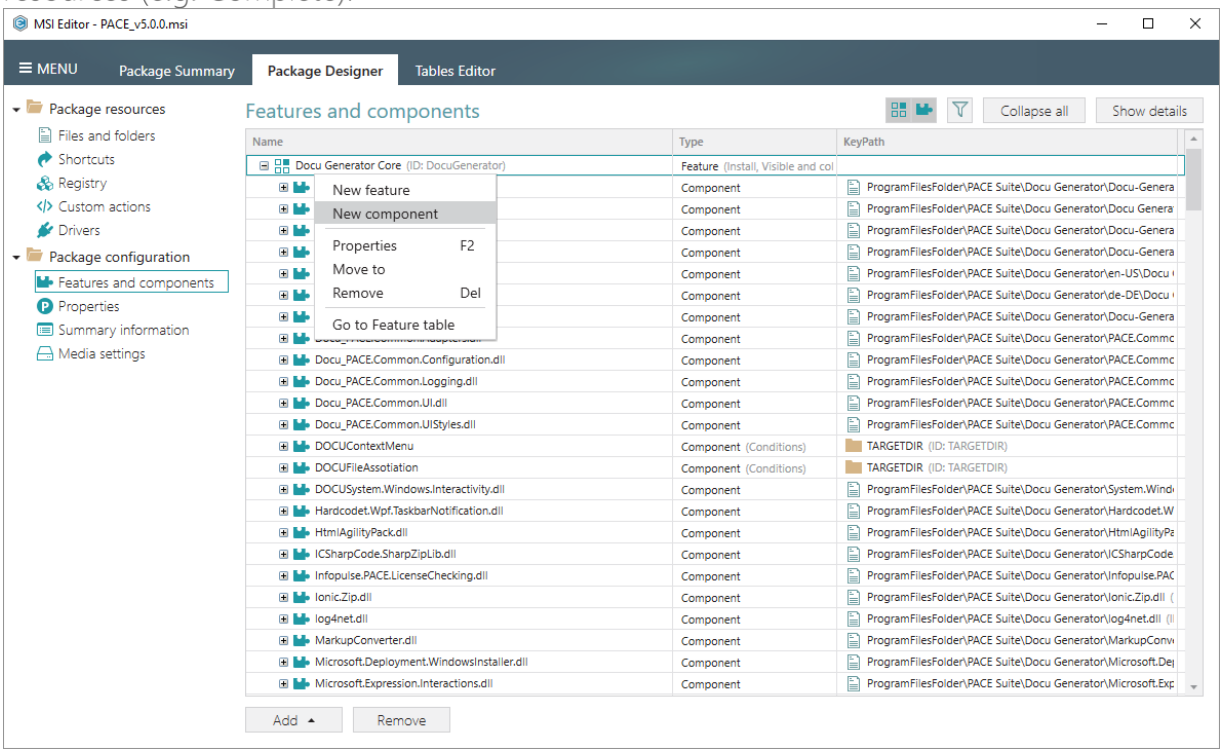
Choose a scenario that better suits your needs:

- Add Registry Key, described in section 3.7.4.1
  Add a new registry key or sub-key manually.
- Add Registry Value, described in section 3.7.4.2
  Add a new registry value manually.
    - Add ActiveSetup, described in section 3.7.4.2.1
      Intergrate ActiveSetup into your package to propagate the per-user resources for every logged-in user of the system.
- 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,

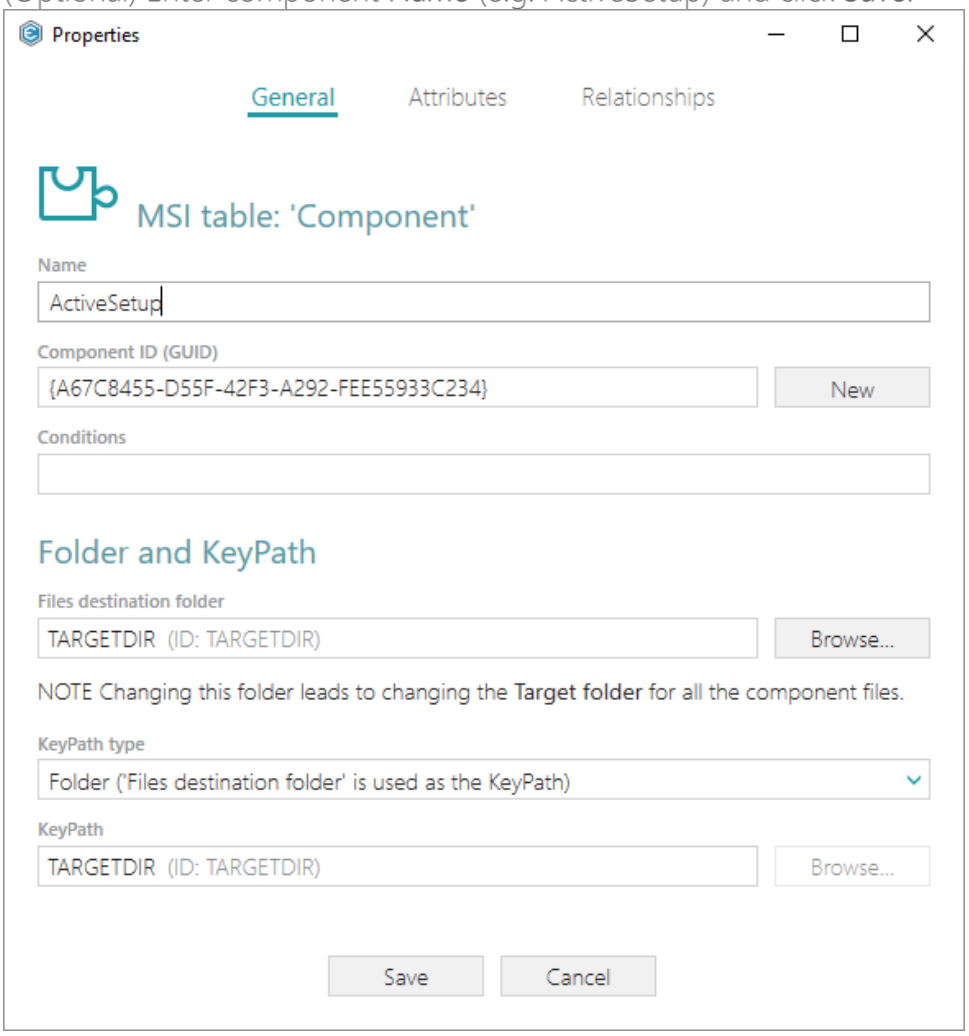
go to the Features and components tab of the Package Designer.



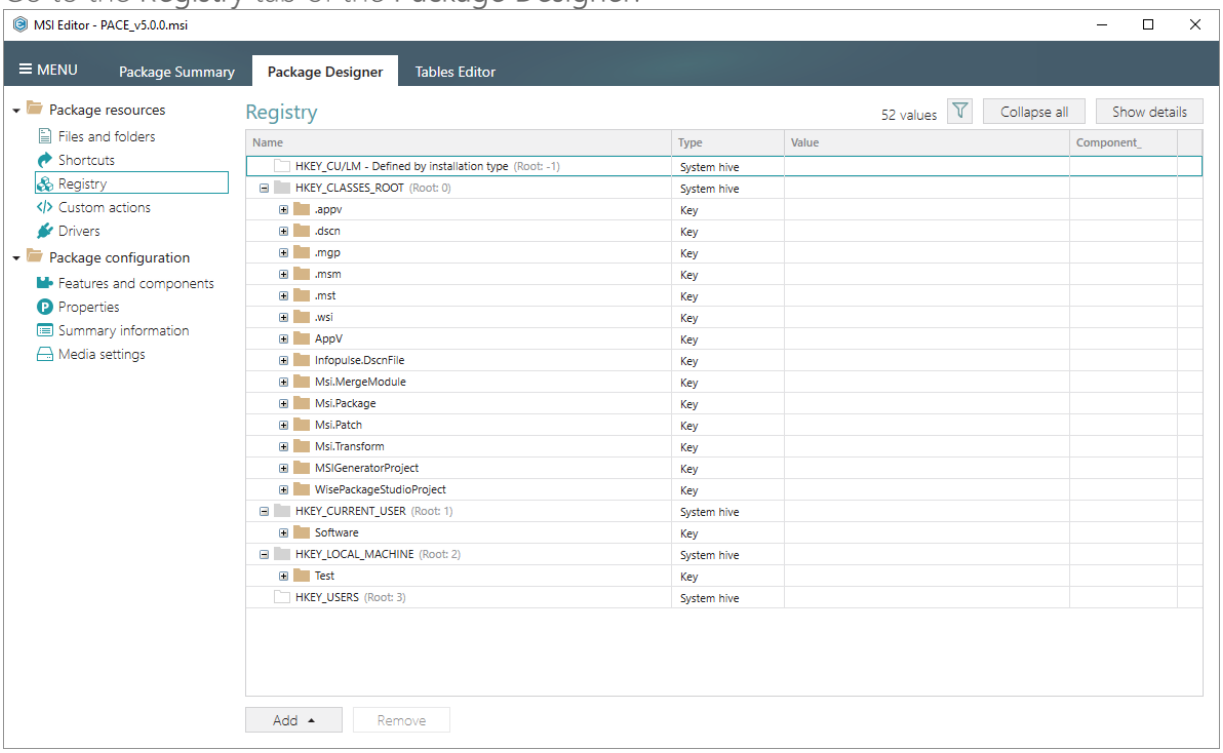

- (Optional) Select New component from the context menu of any feature. It is recommended to use feature that always will be installed and contains per-machine resources (e.g. Complete).



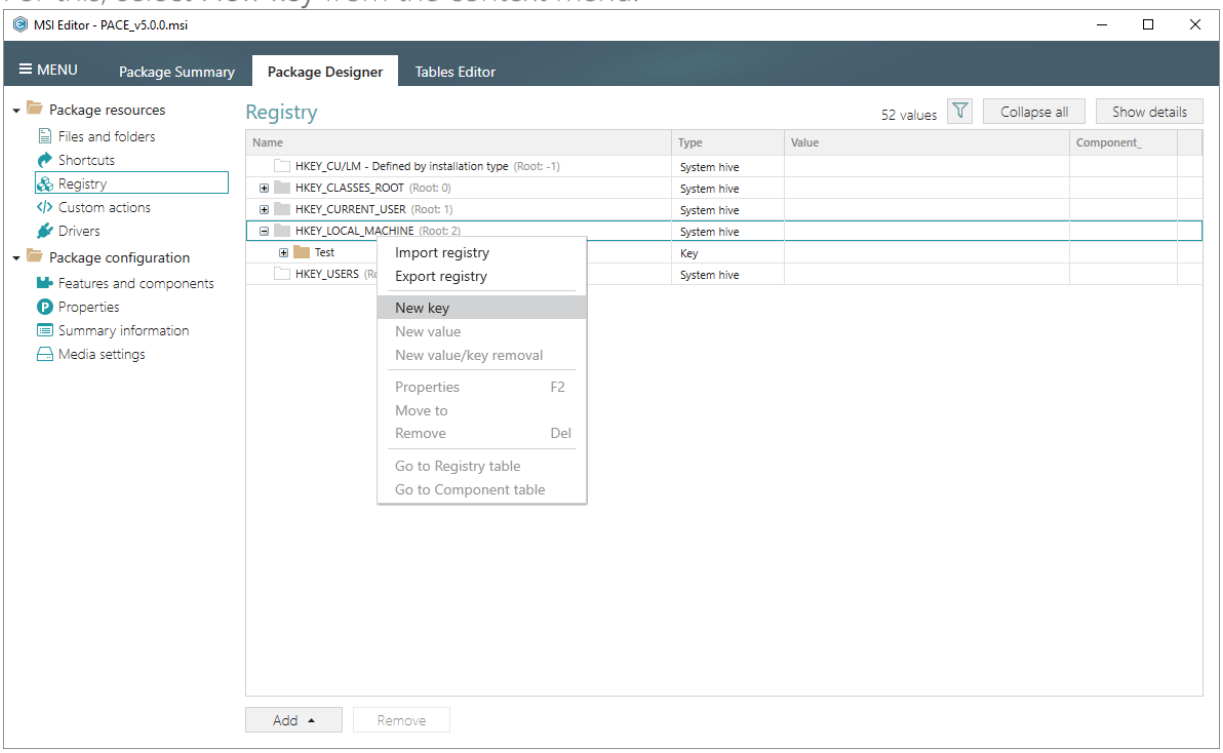
- (Optional) Enter component **Name** (e.g. ActiveSetup) and click **Save**.



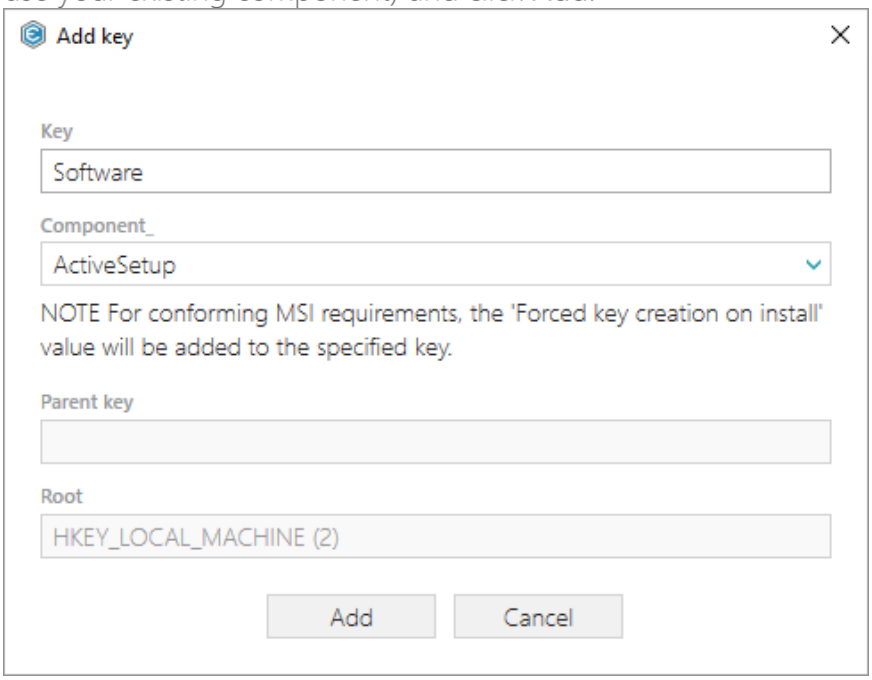

- Go to the **Registry** tab of the **Package Designer**.



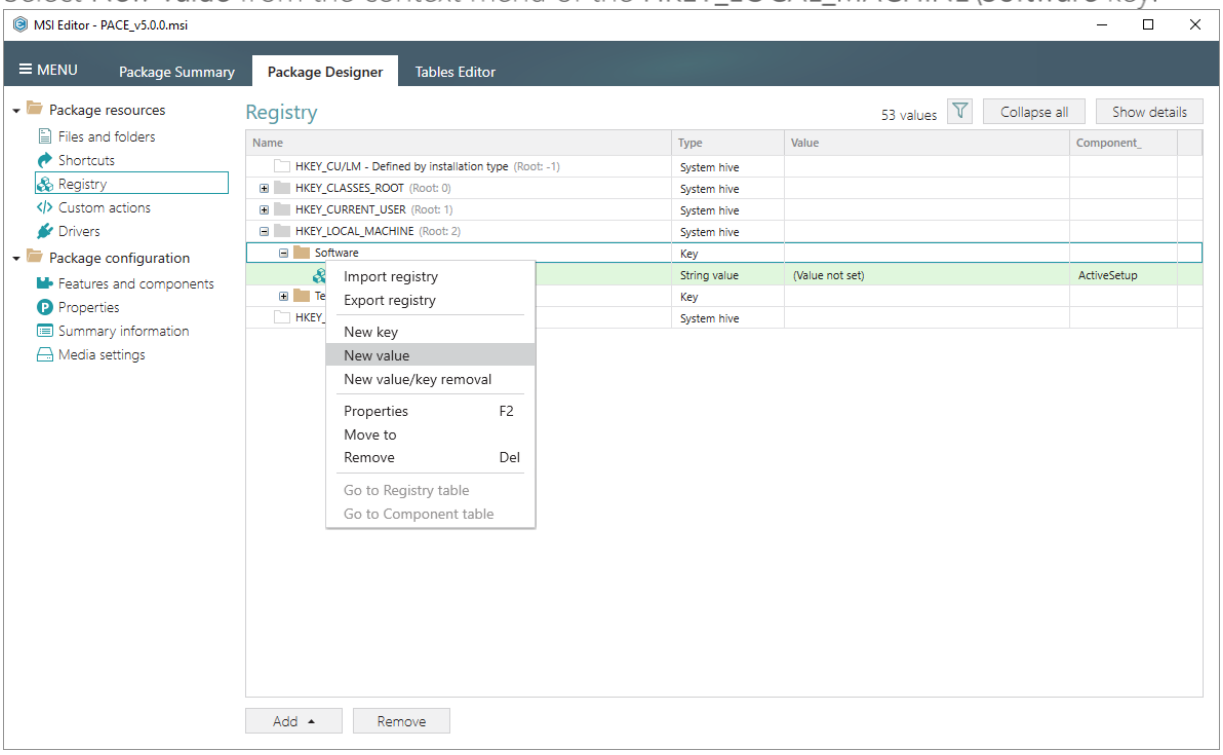
- Create the "Software" key under the HKEY_LOCAL_MACHINE hive, if the key does not exist. For this, select New key from the context menu.




- Enter the key name "Software", select the previously created ActiveSetup component (or use your existing component) and click Add.

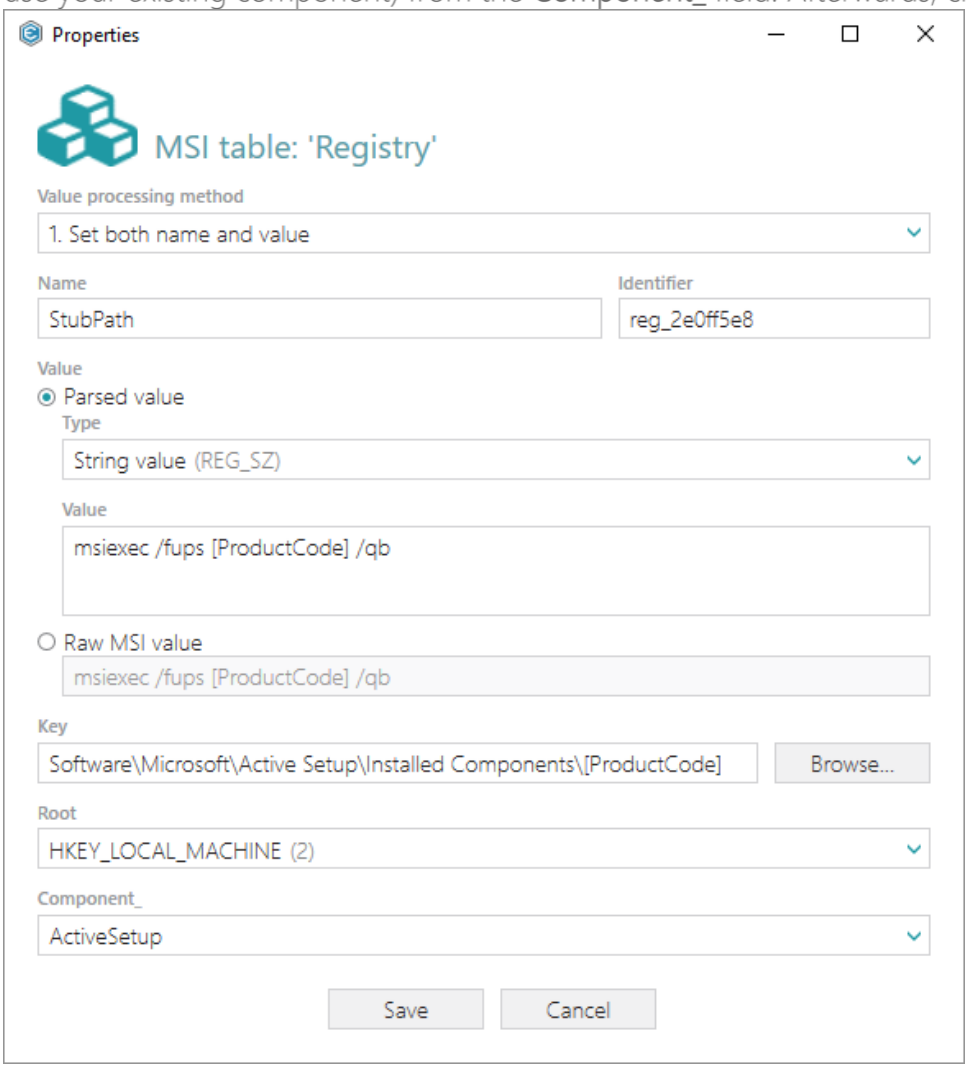


- Select **New value** from the context menu of the **HKEY_LOCAL_MACHINE\Software** key.

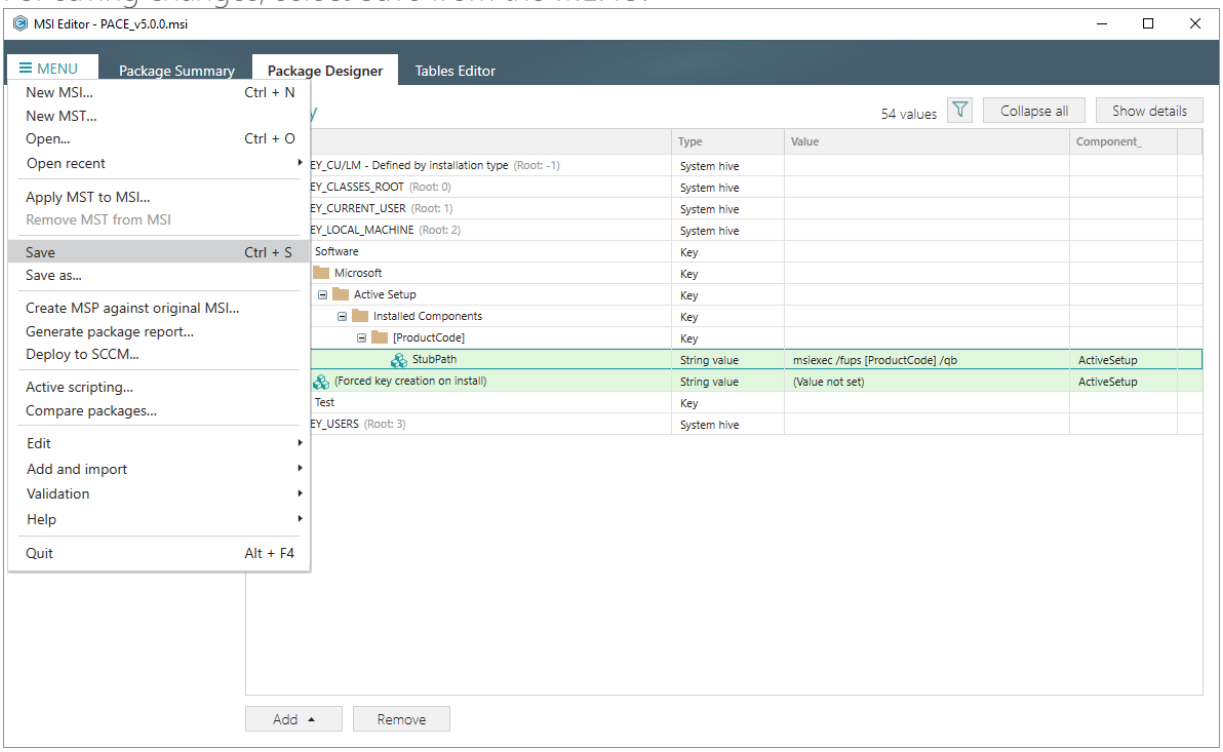



- Enter the "*StubPath*" to **Name**, the "*msiexec /fups [ProductCode] /qb*" to **Value**, replace value of the **Key** field with the "*Software\Microsoft\Active Setup\Installed Components\[ProductCode]*" and select the previously created **ActiveSetup** component (or

use your existing component) from the Component_ field. Afterwards, click Save.




- For saving changes, select Save from the MENU.

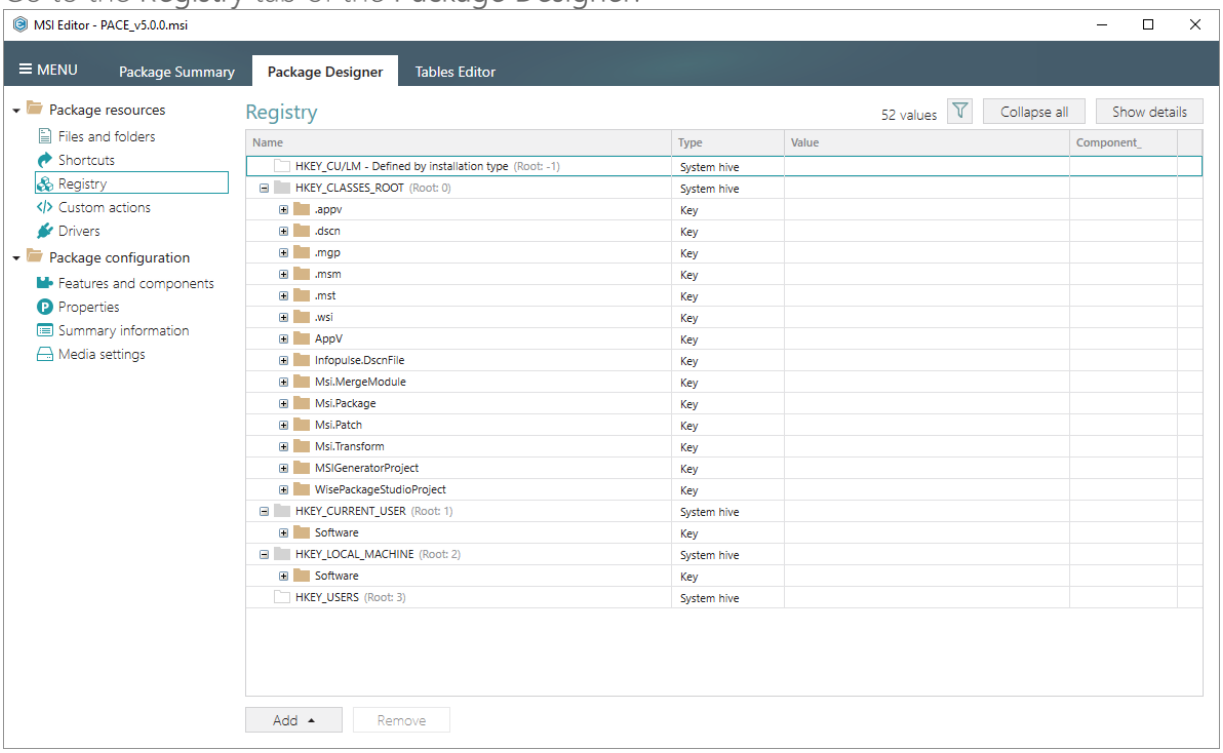


- Add Value/Key Removal, described in section 3.7.4.2.1
  Add a new RemoveRegistry value manually for sucuring a value or a key removal during the package installation.
- Import Registry, described in section 3.7.4.4
  Import registry from the REG file.
- Export Registry, described in section 3.7.4.5
  Export the selected registry to the REG file.
- Set Permissions, described in section 3.7.4.6
  Set custom permissions for the registry key.
- Remove Permissions, described in section 3.7.4.7
  Locate registry key with custom permissions and remove the unnecessary permissions.
- Move Value, Key, described in section 3.7.4.8
  Move a registry value or a key from one parent key to another one.
- Edit Registry Key, described in section 3.7.4.9
  Modify details and options of the selected registry key.
- Edit Registry Value, described in section 3.7.4.10
  Modify details of the selected registry value.
- Edit Value/Key Removal, described in section 3.7.4.11
  Modify details of the selected value/key removal item (RemoveRegistry) to secure a value or a key removal during the package installation.

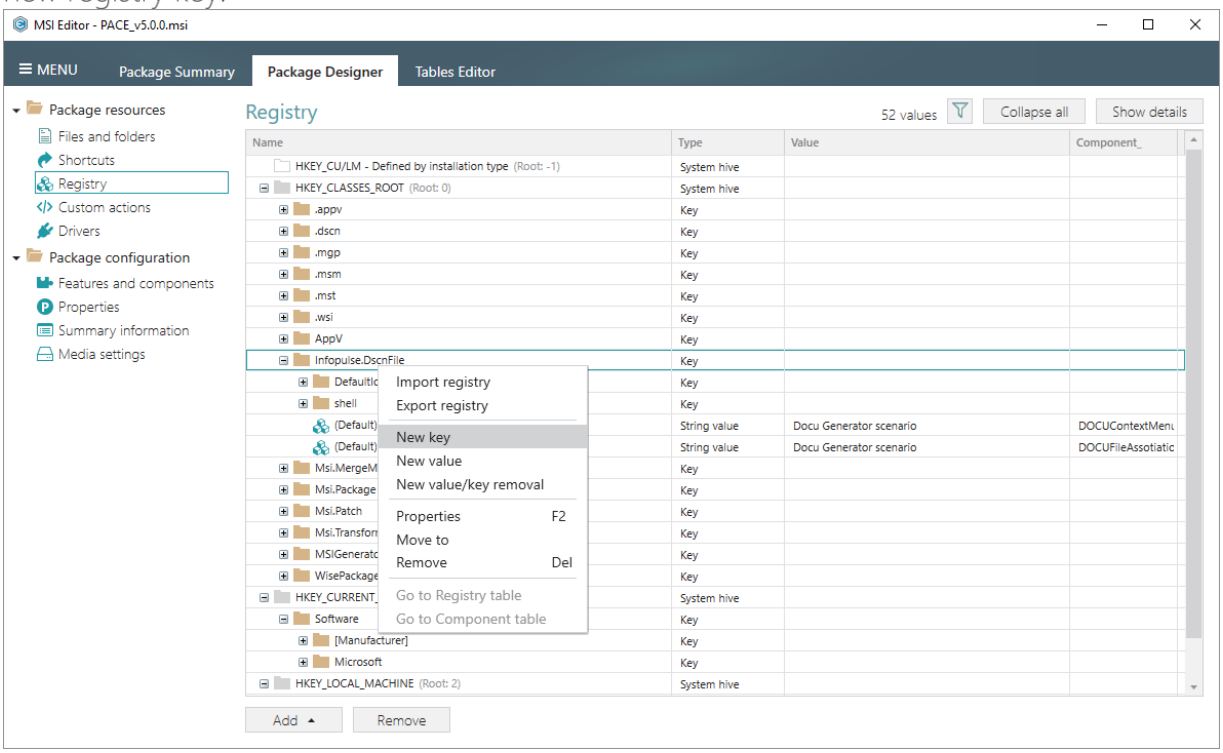
### 3.7.4.1 Add Registry Key

Add a new registry key or sub-key manually.

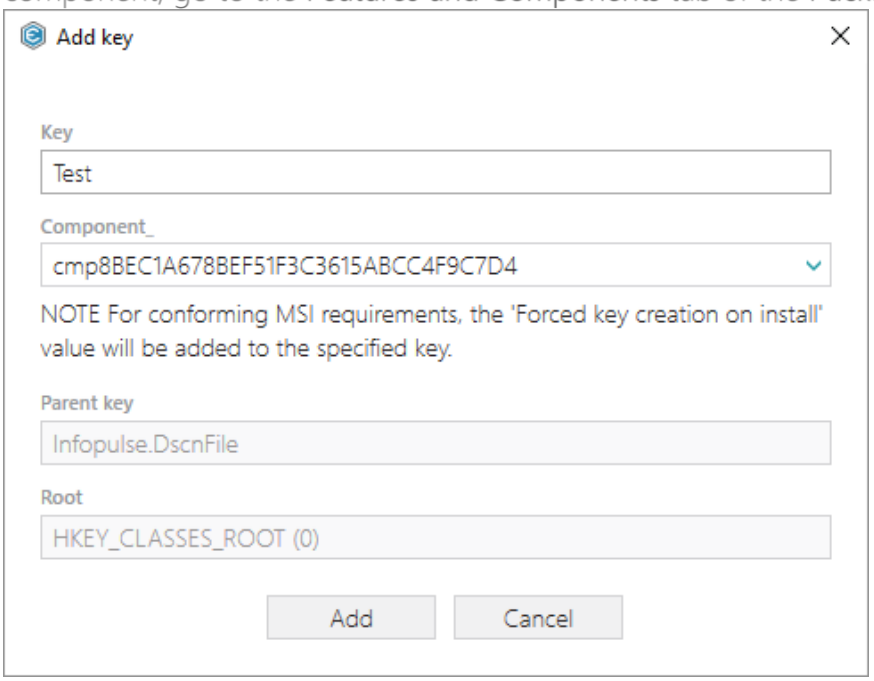
- Go to the **Registry** tab of the **Package Designer**.



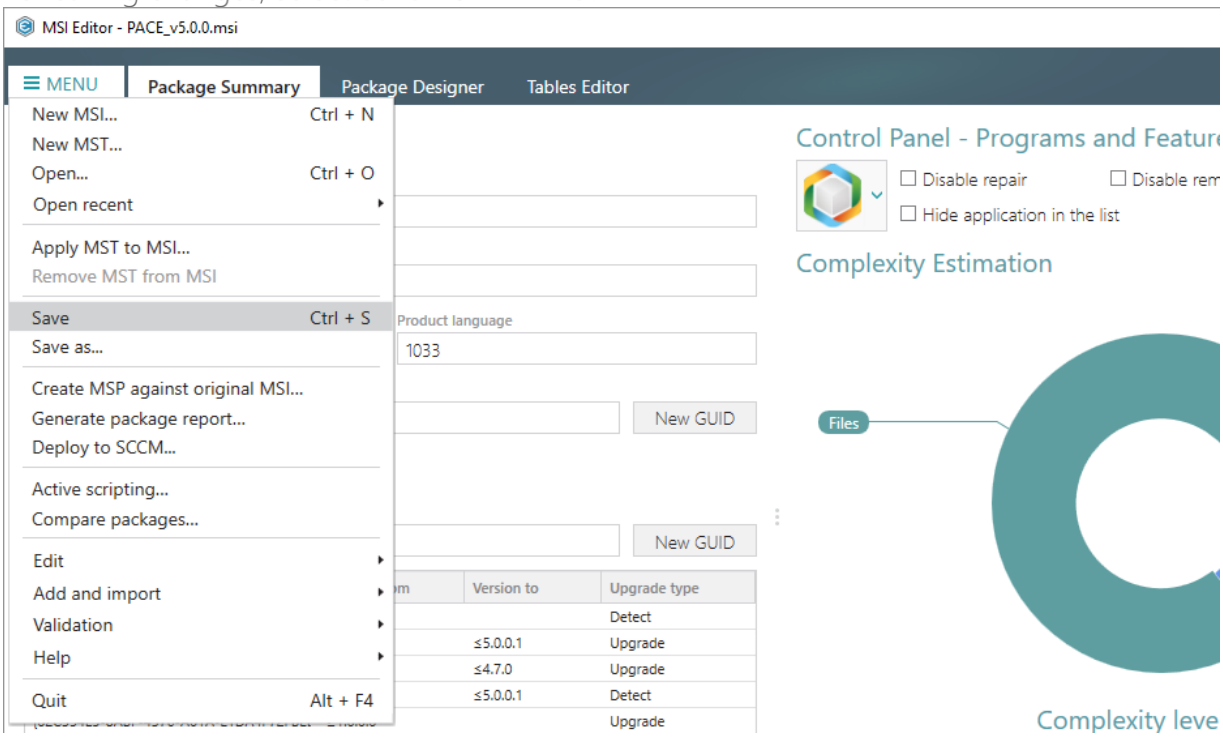

- Select **New key** from the context menu of a root hive or a key, where you want to create a new registry key.



- 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**.



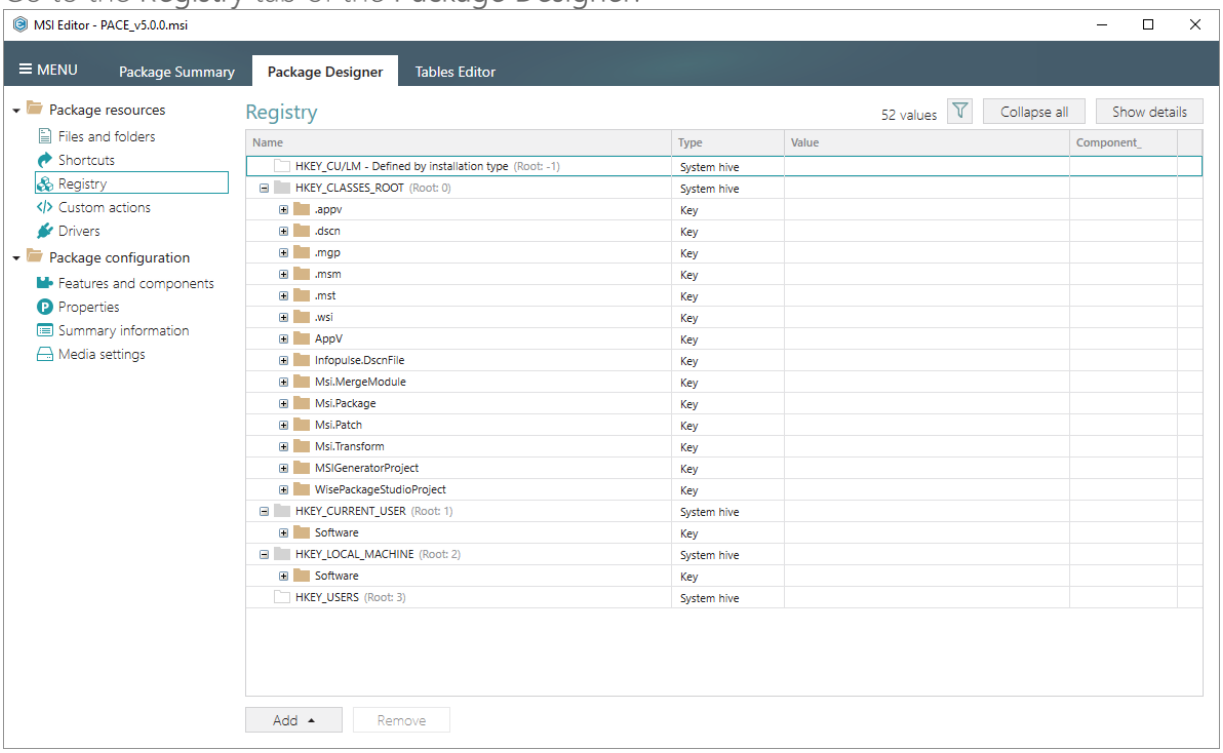

- For saving changes, select **Save** from **MENU**.



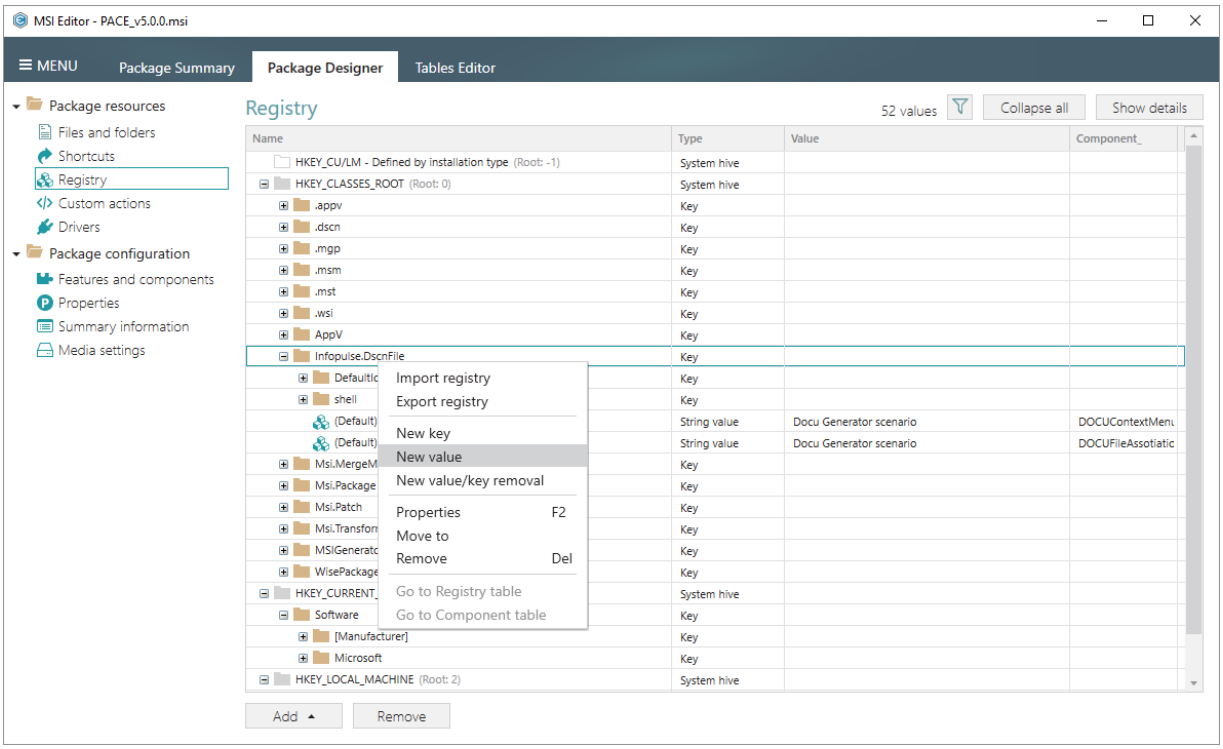

### 3.7.4.2 Add Registry Value

Add a new registry value manually.

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



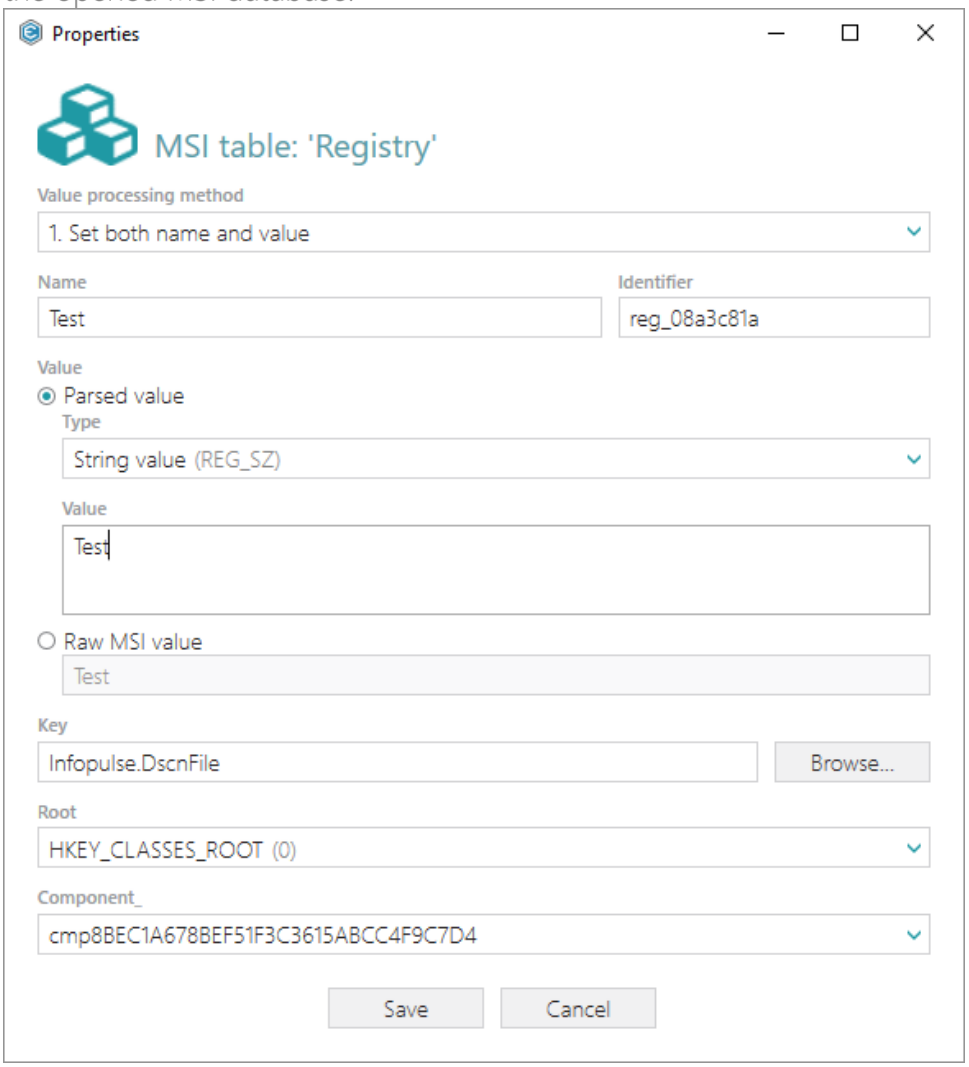

- Select **New value** from the context menu of a key, where you want to create a new registry value.



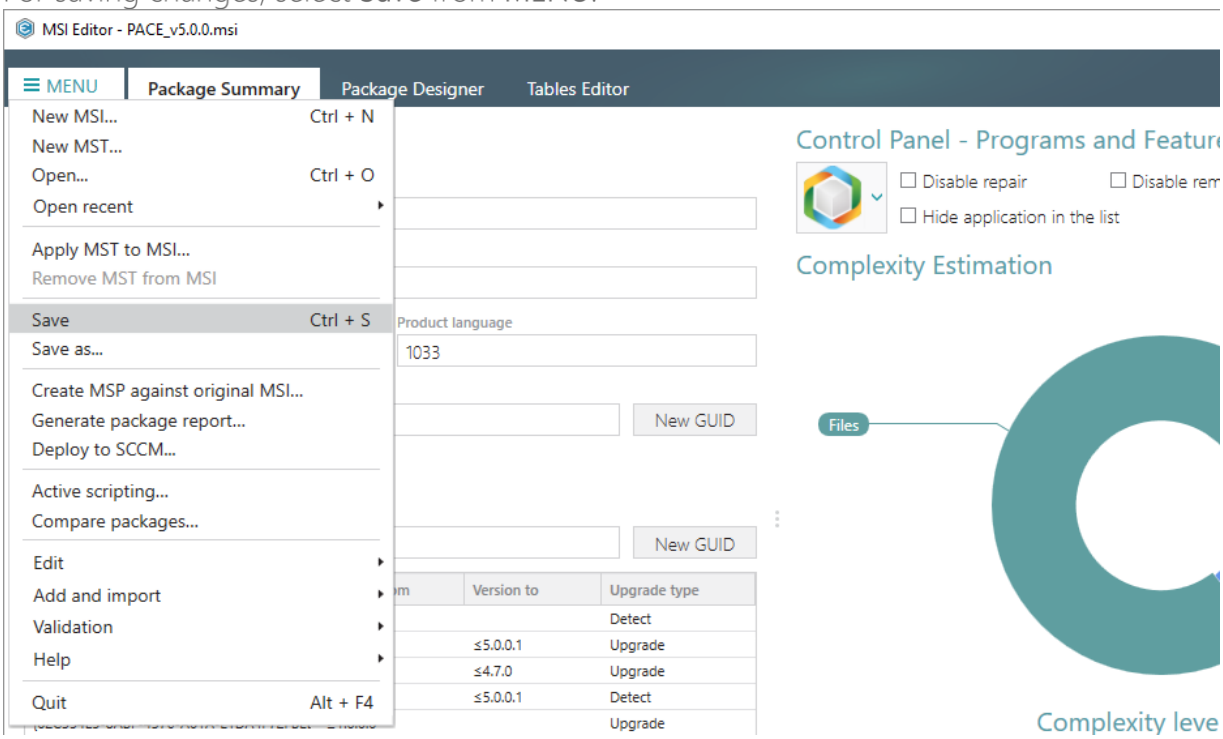

- For creating an ordinary registry value, just specify a **Name**, **Value** itself, select a **Component** from the list, to which this value will be assigned, and then click **Save**.

  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 changes, select Save from MENU.

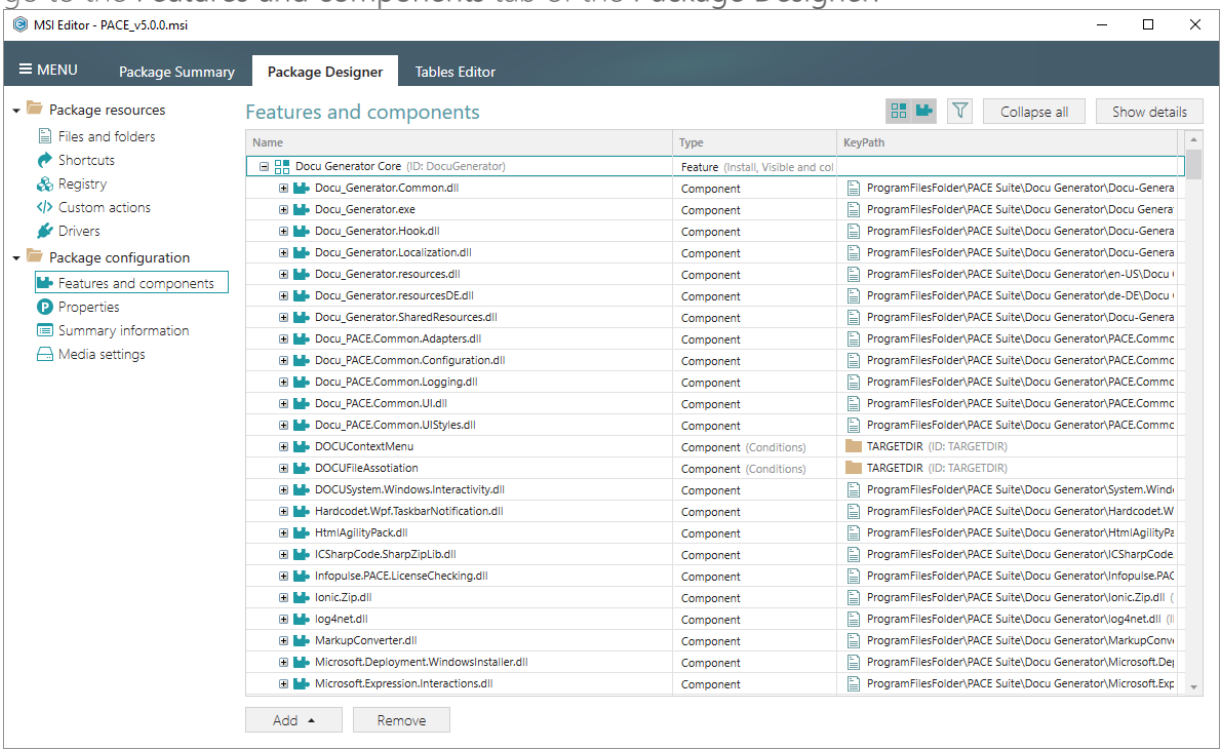



### 3.7.4.2.1 Add ActiveSetup

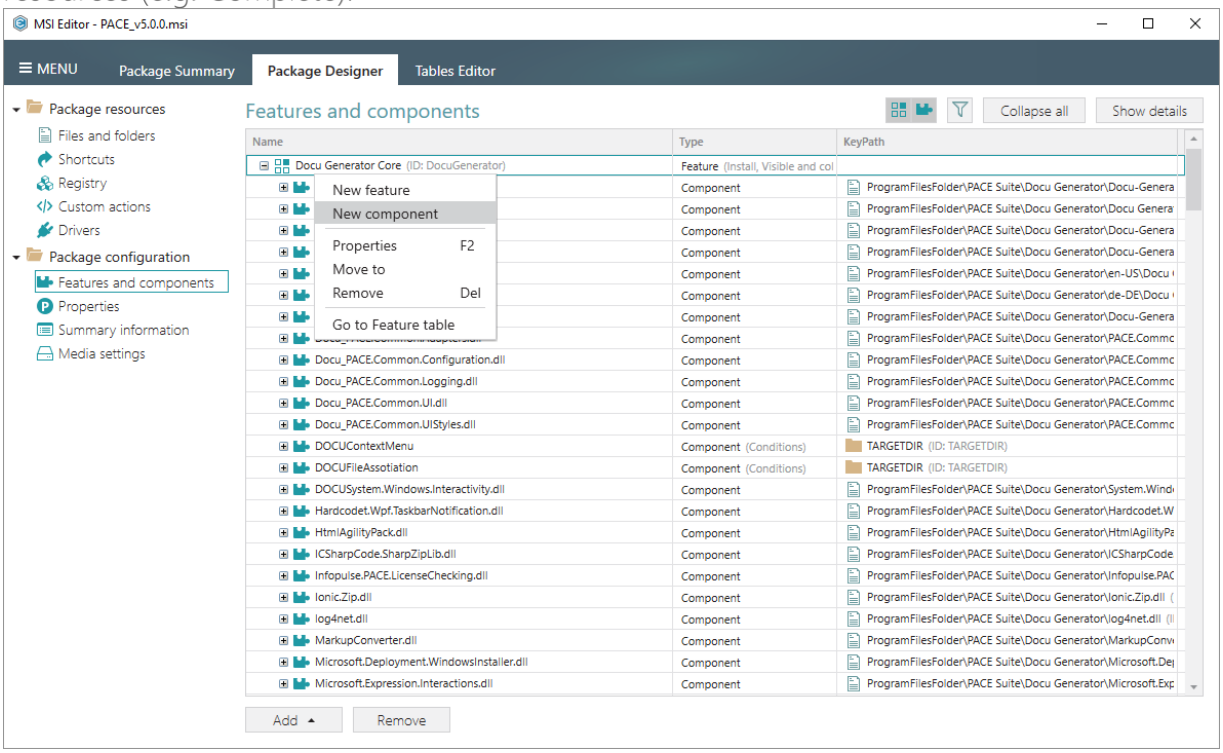
Intergrate ActiveSetup into your package to propagate the per-user resources for every logged-in user of the system. The ActiveSetup is represented as the registry value that triggers the repair function of the Windows Installer during the next user log-in in order to restore the missing resources.

- (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,

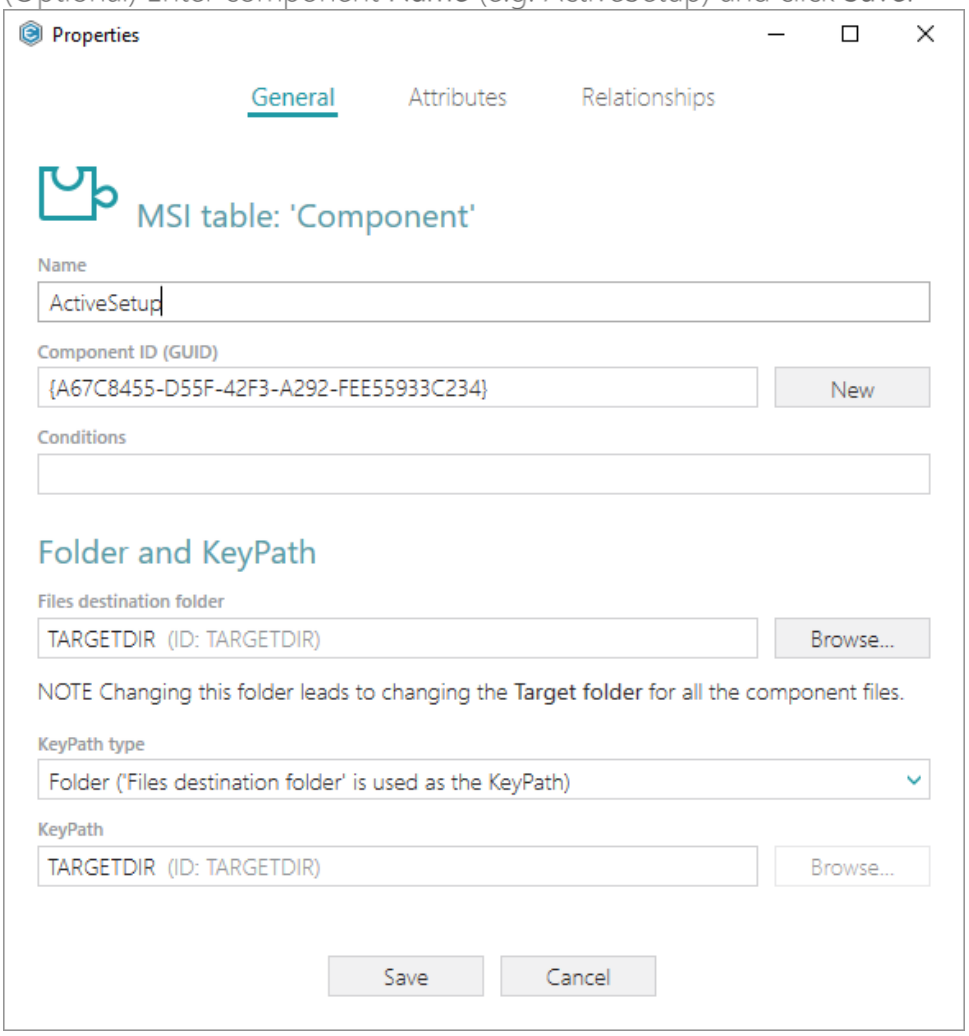
go to the Features and components tab of the Package Designer.



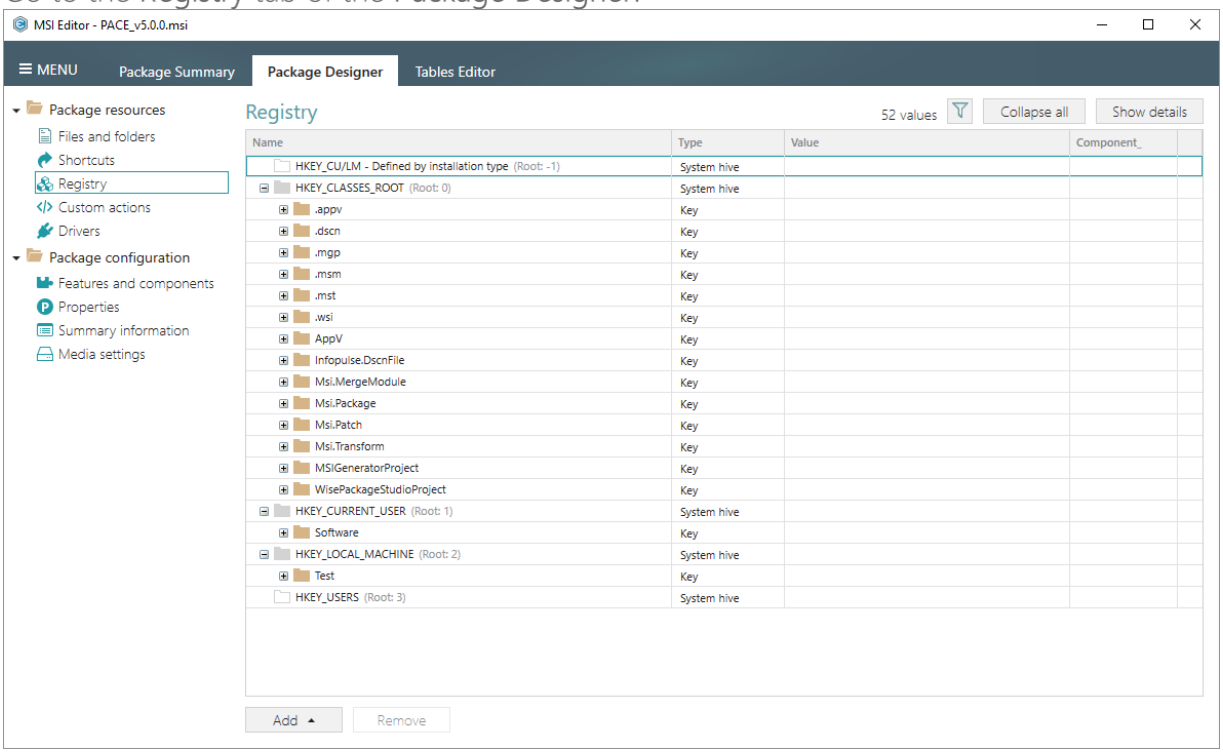

- (Optional) Select New component from the context menu of any feature. It is recommended to use feature that always will be installed and contains per-machine resources (e.g. Complete).



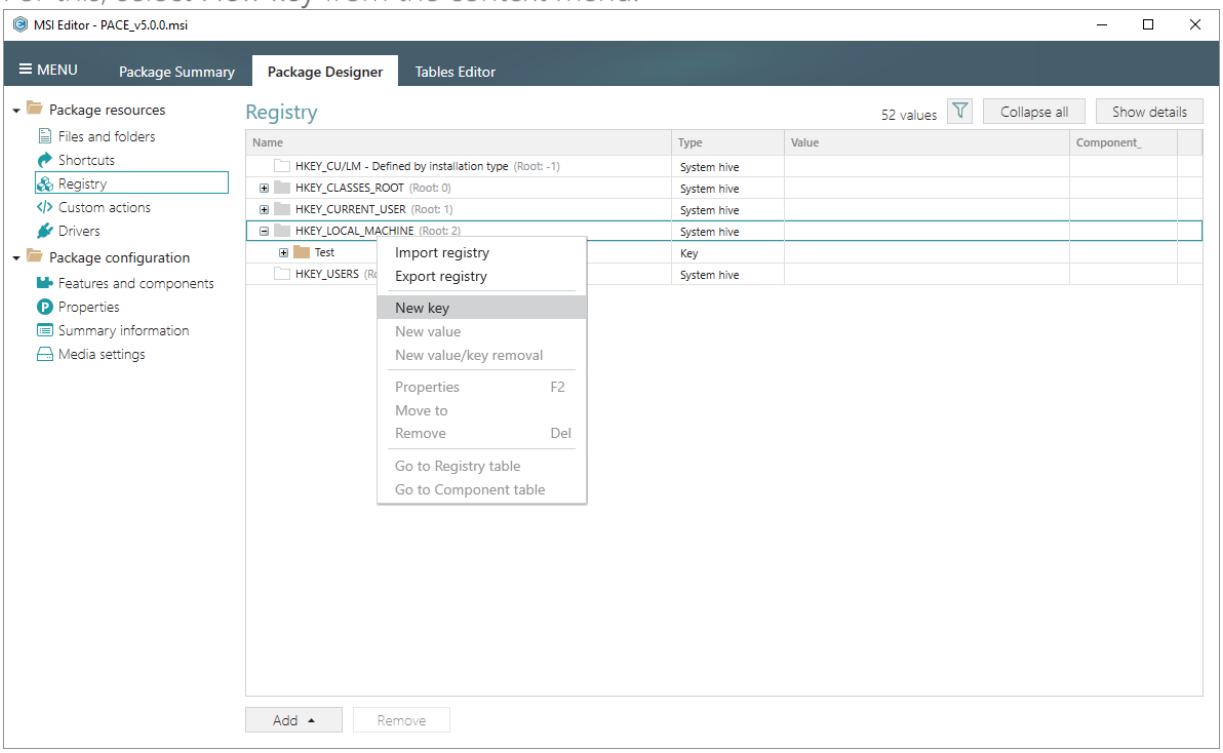
- (Optional) Enter component **Name** (e.g. ActiveSetup) and click **Save**.



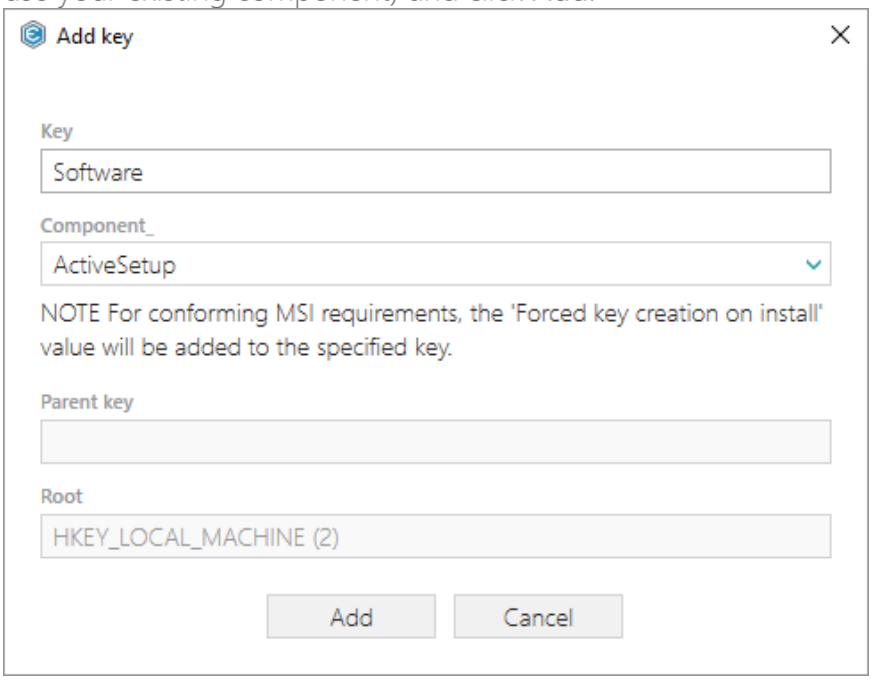

- Go to the **Registry** tab of the **Package Designer**.



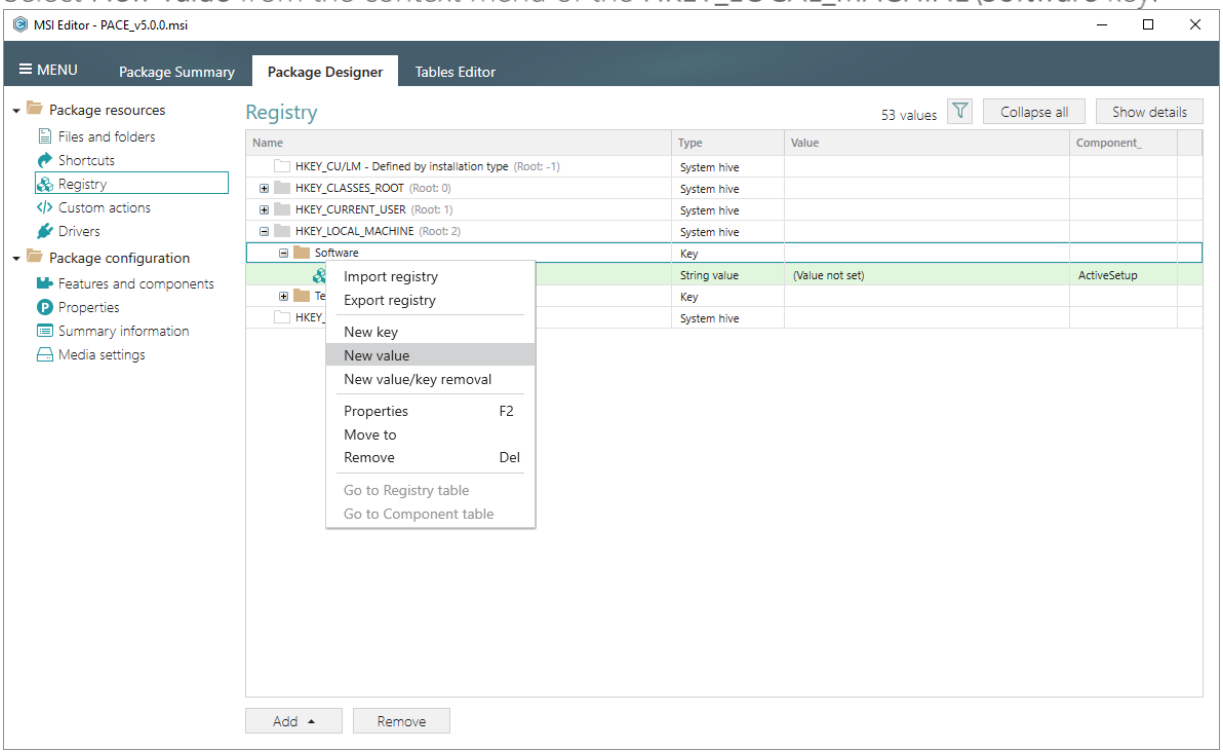
- Create the "Software" key under the HKEY_LOCAL_MACHINE hive, if the key does not exist. For this, select New key from the context menu.




- Enter the key name "Software", select the previously created ActiveSetup component (or use your existing component) and click Add.

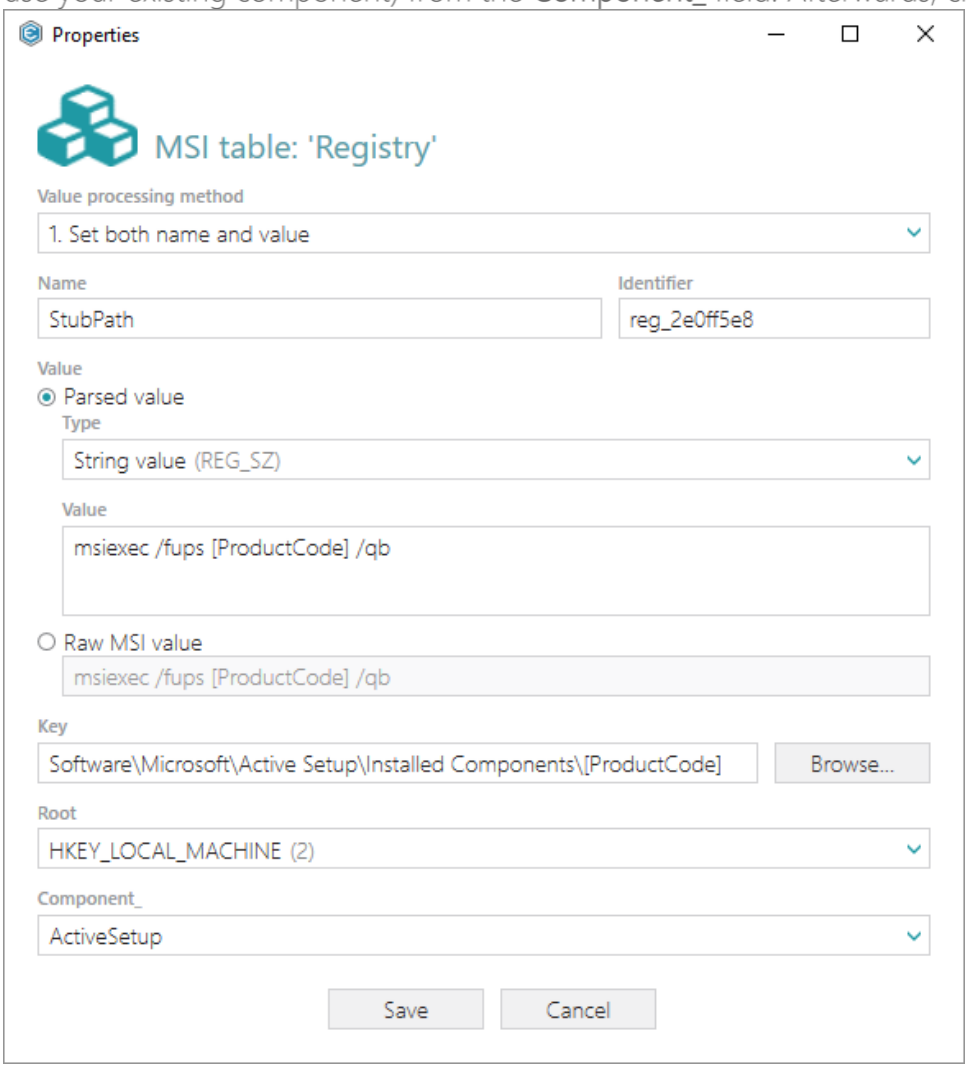


- Select **New value** from the context menu of the **HKEY_LOCAL_MACHINE\Software** key.

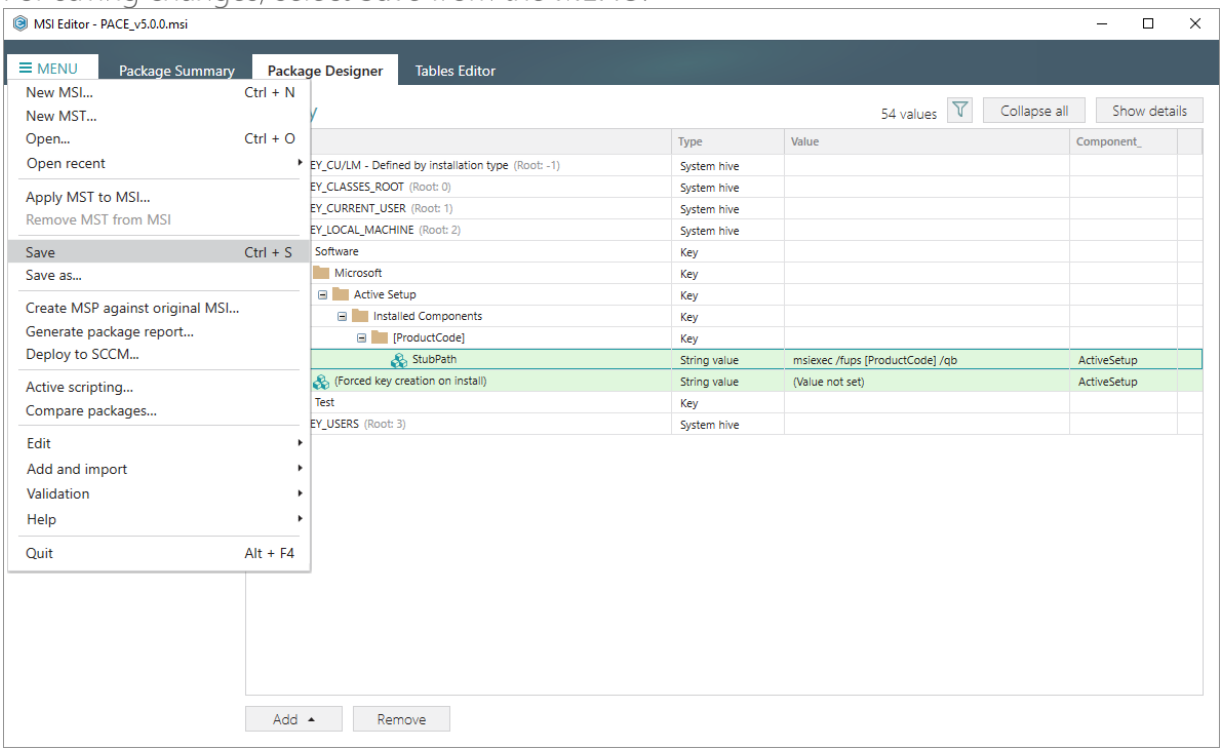



- Enter the "*StubPath*" to **Name**, the "*msiexec /fups [ProductCode] /qb*" to **Value**, replace value of the **Key** field with the "*Software\Microsoft\Active Setup\Installed Components\[ProductCode]*" and select the previously created **ActiveSetup** component (or

use your existing component) from the Component_ field. Afterwards, click Save.



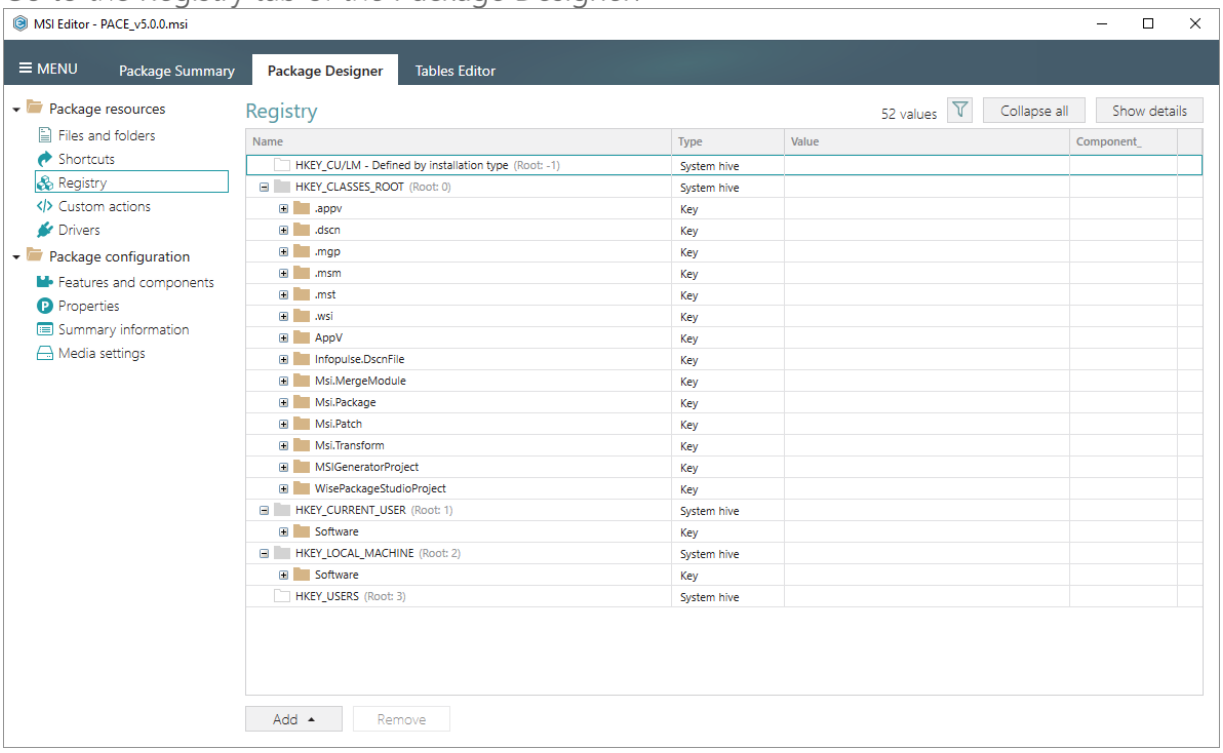

- For saving changes, select Save from the MENU.



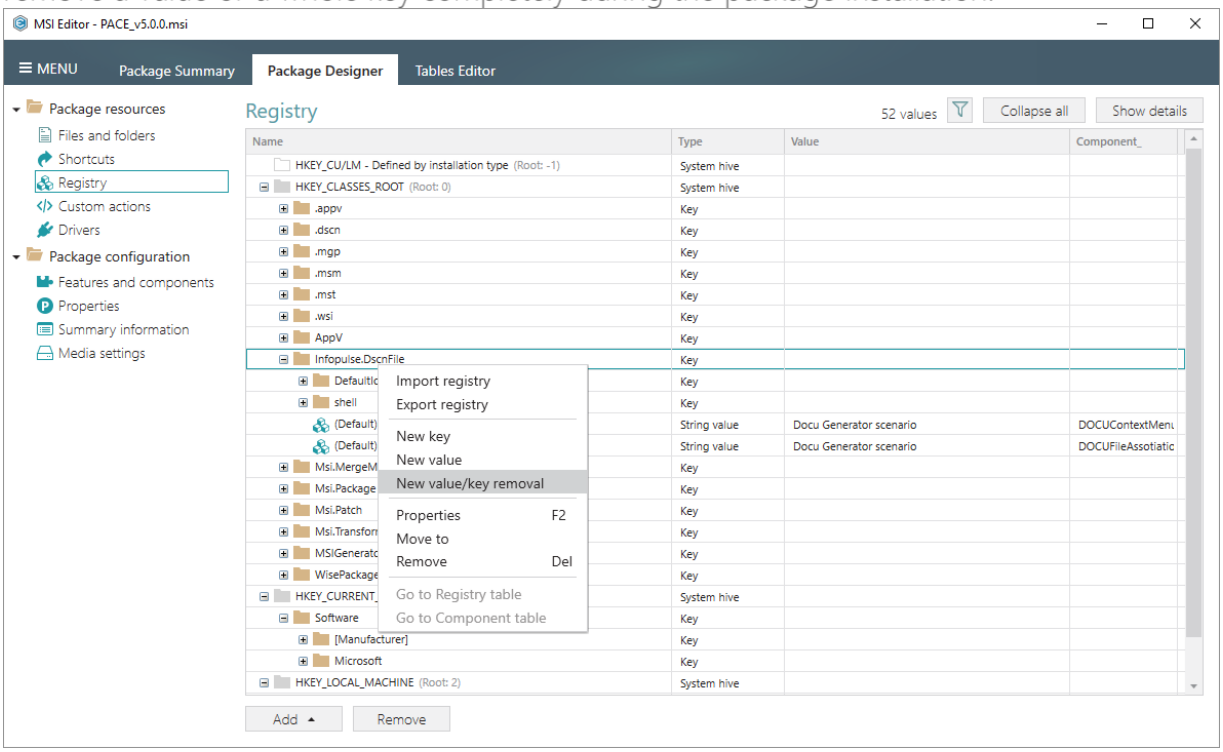
### 3.7.4.3 Add Value/Key Removal

Add a new RemoveRegistry value manually for sucuring a value or a key removal during the package installation.

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

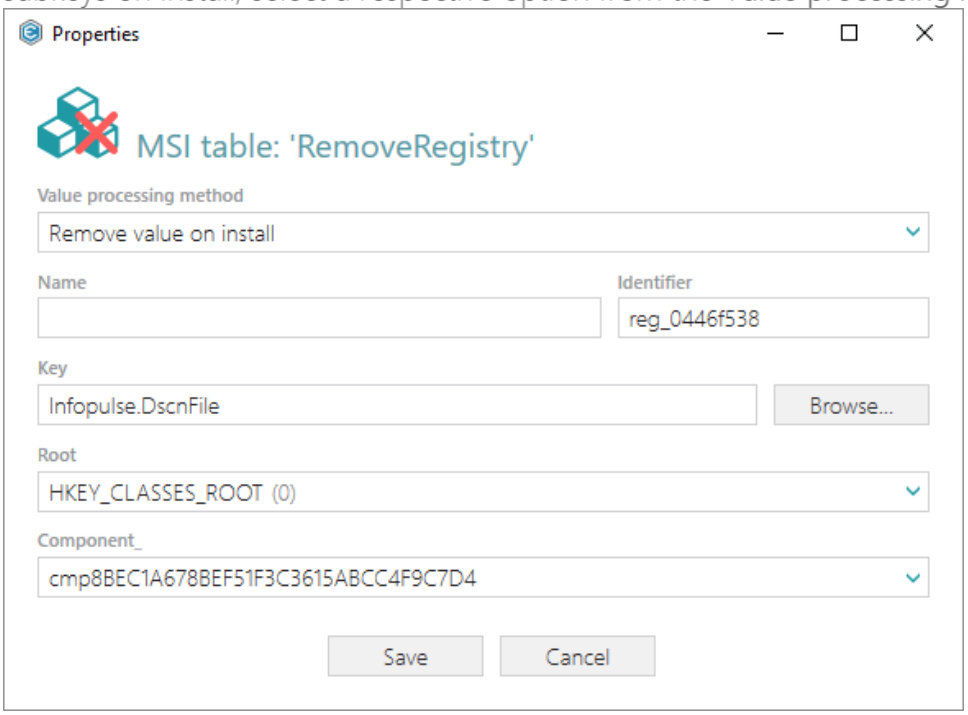



- Select **New value/key removal** from the context menu of a key, from which you want to remove a value or a whole key completely during the package installation.

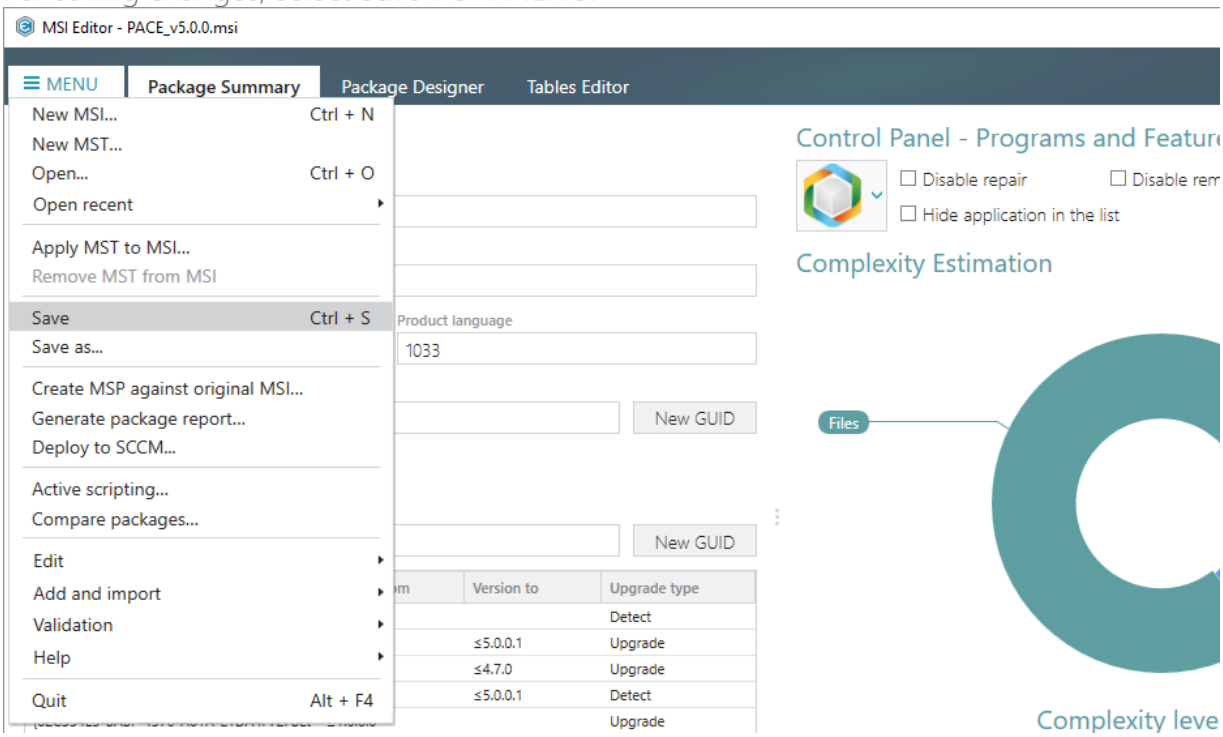



- For removing a registry value, specify its **Name**, select a **Component** from the list, which will controls the deletion of thee specified registry value, and then click **Save**.

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.



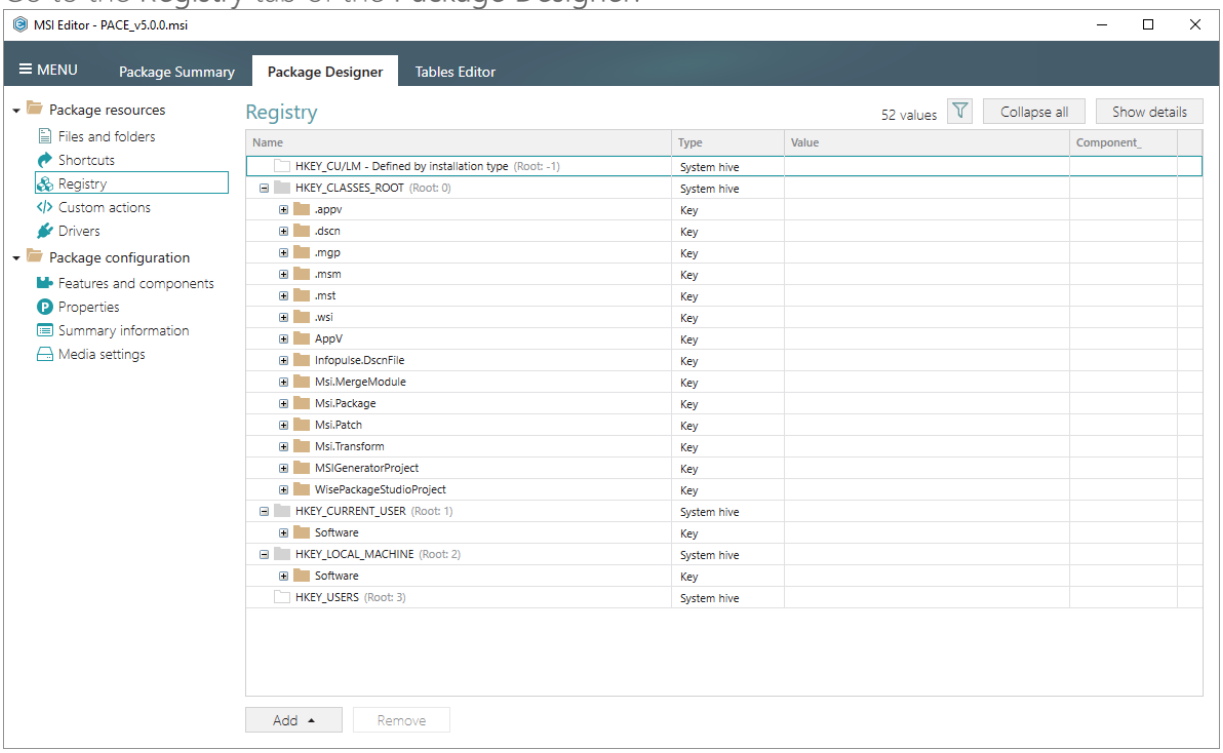

- For saving changes, select Save from MENU.



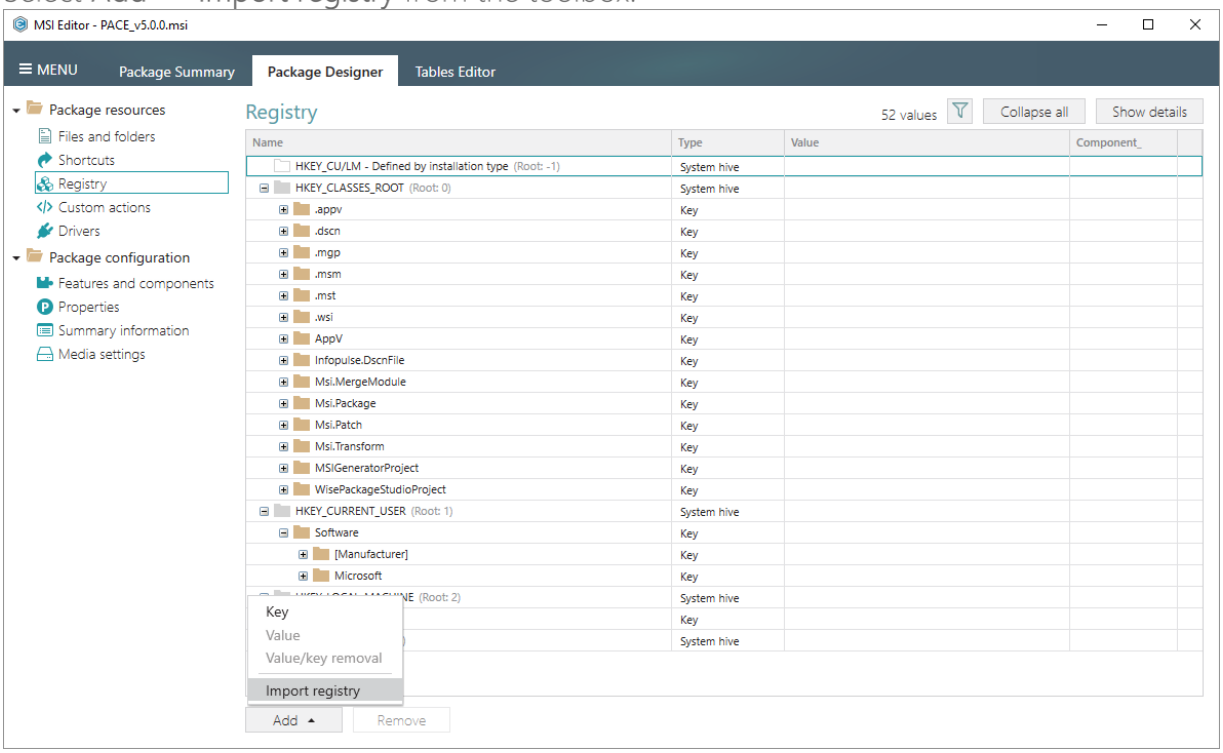
### 3.7.4.4 Import Registry

Import registry from the REG file.

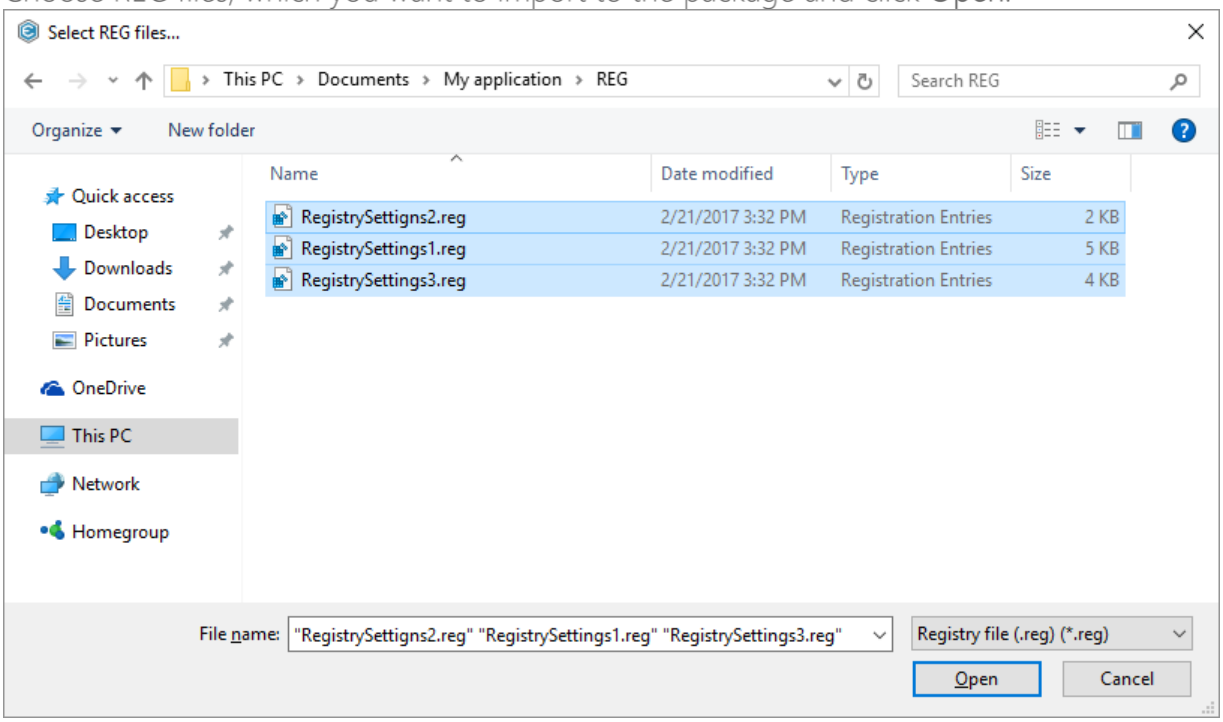
- Go to the Registry tab of the Package Designer.



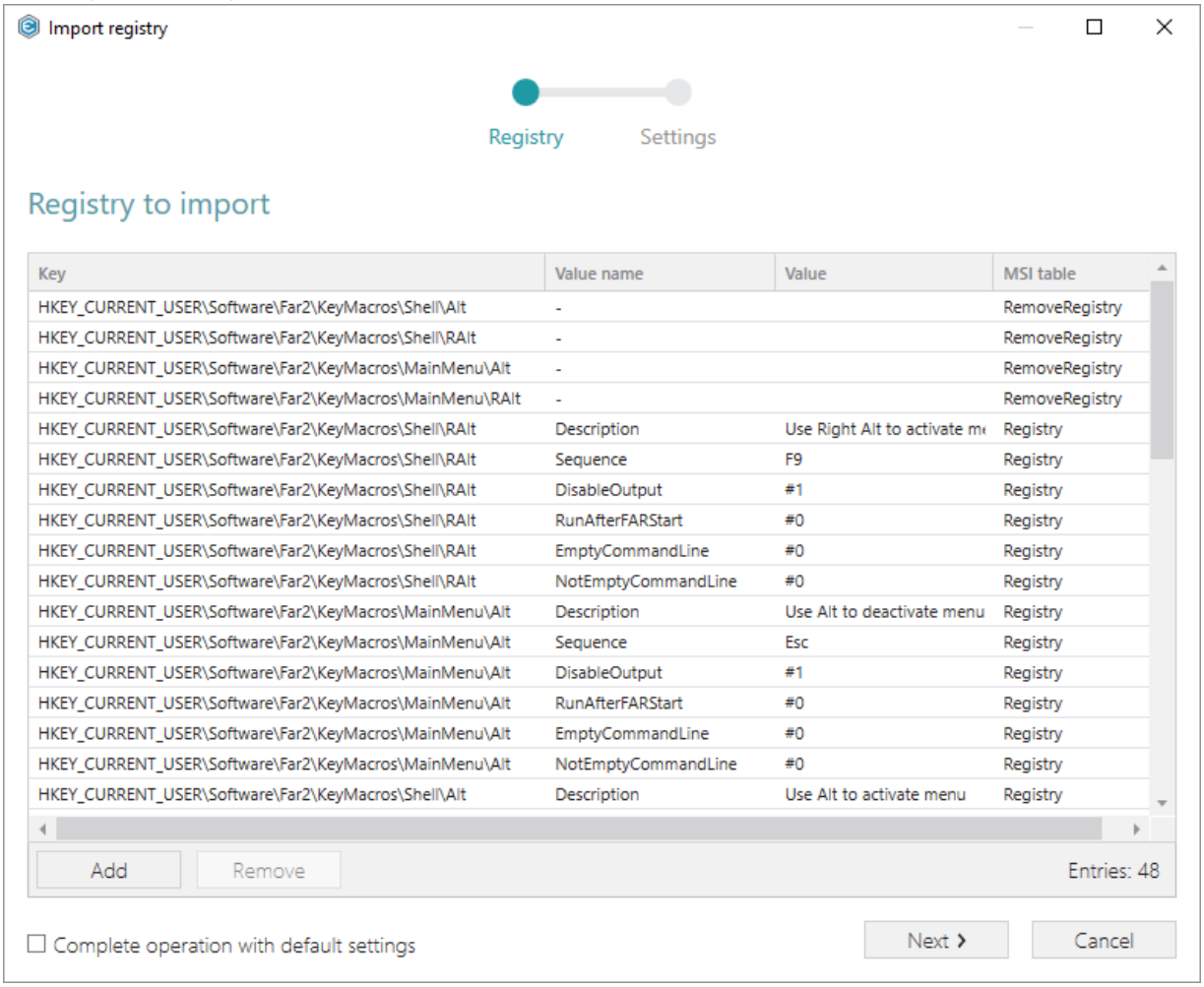

- Select Add -> Import registry from the toolbox.



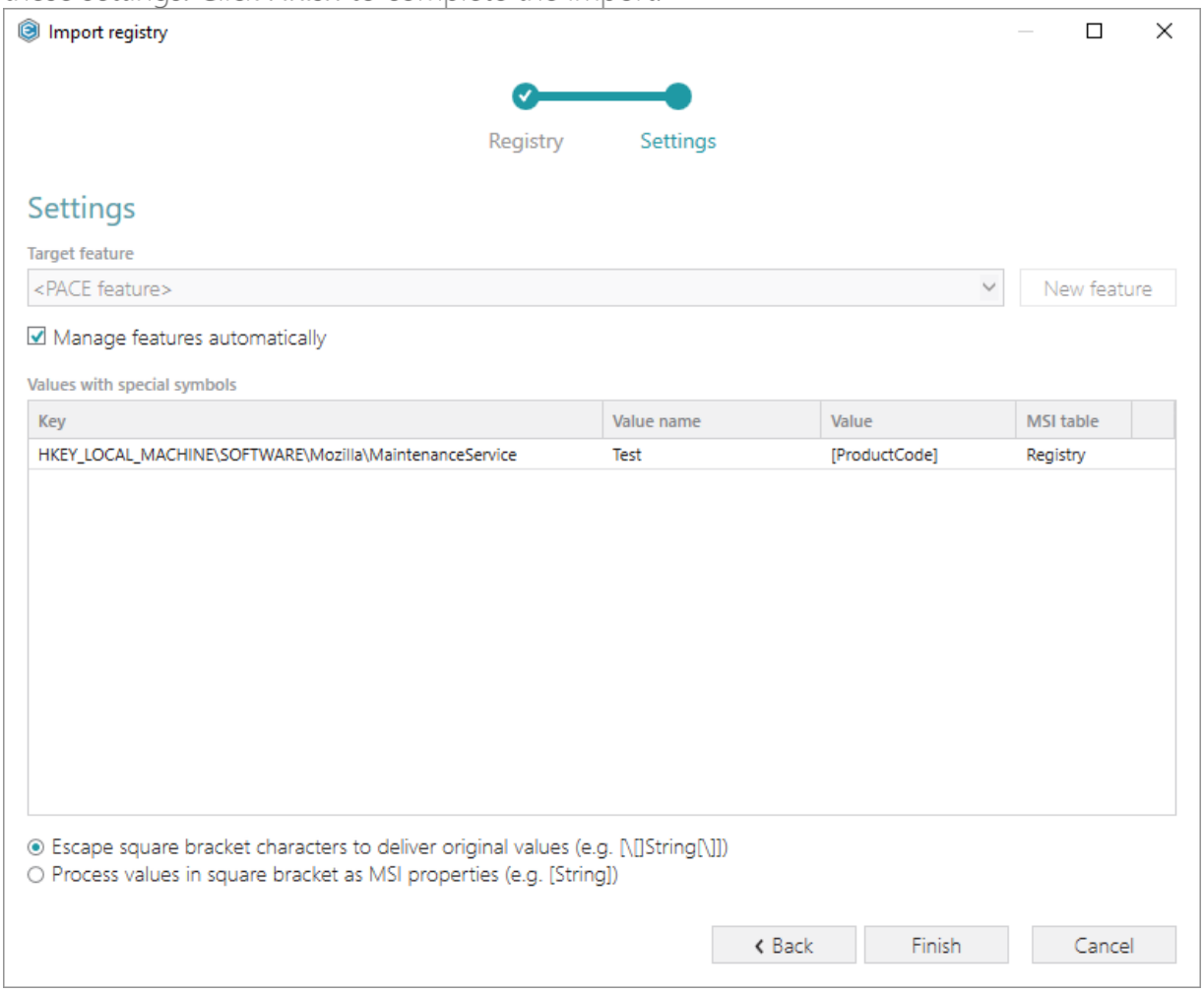
- Choose REG files, which you want to import to the package and click Open.




- Review the registry entries, parsed from the selected REG files, remove the unnecessary entries, if needed, 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

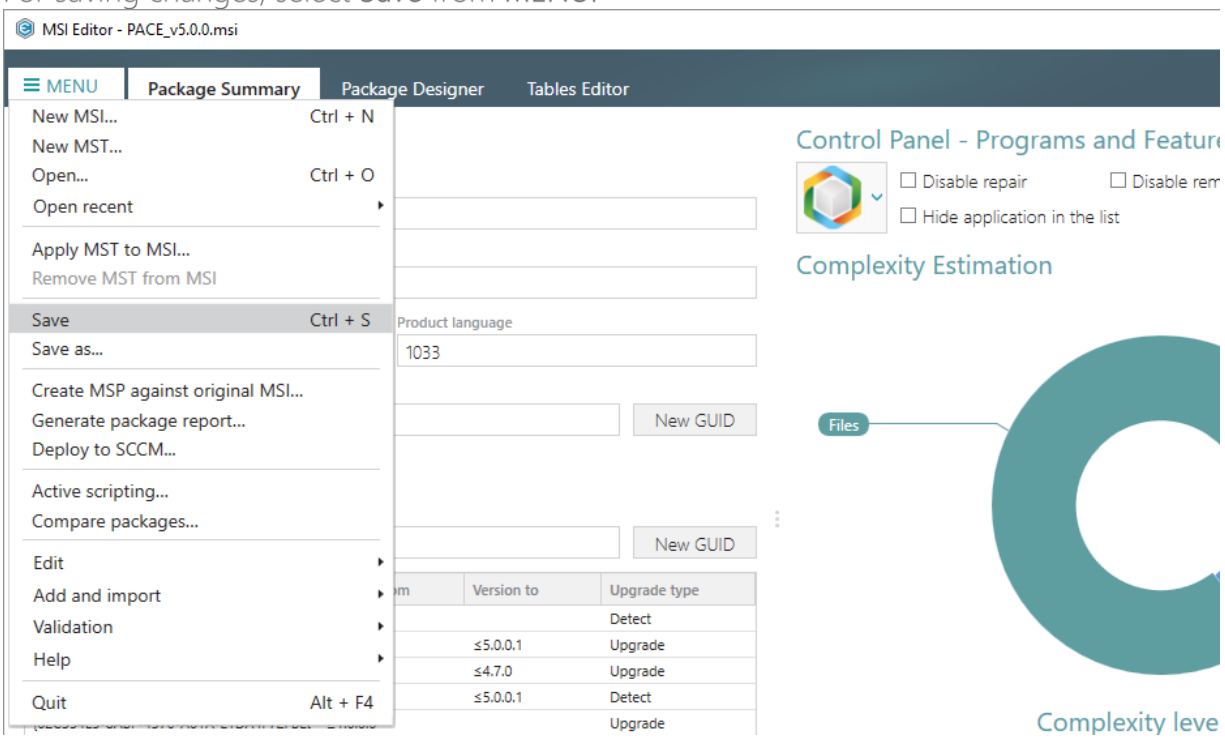
| | |
|---|---|
| Manage features automatically | The specific PACE features will be created (or used if they exist) for the components, containing the imported registry. Feature structure is based on the packaging best practices and depends on the registry key location. |
| <a feature, selected from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

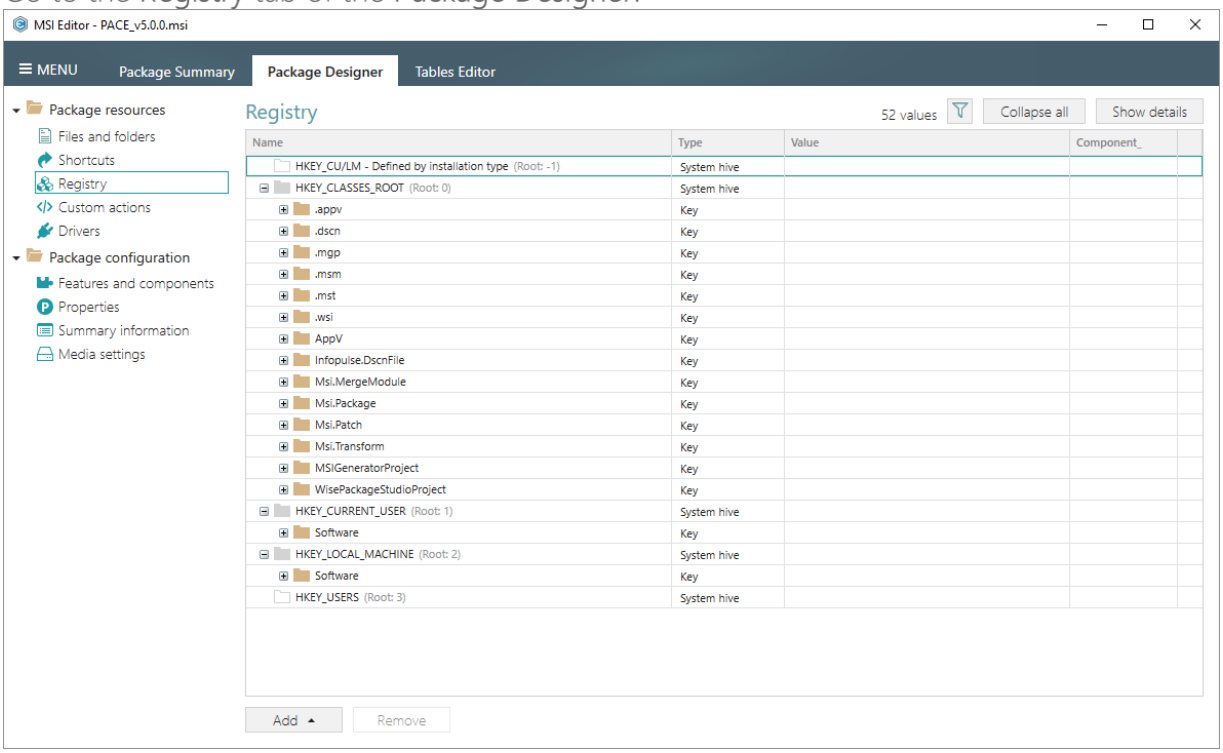
- For saving changes, select **Save** from **MENU**.



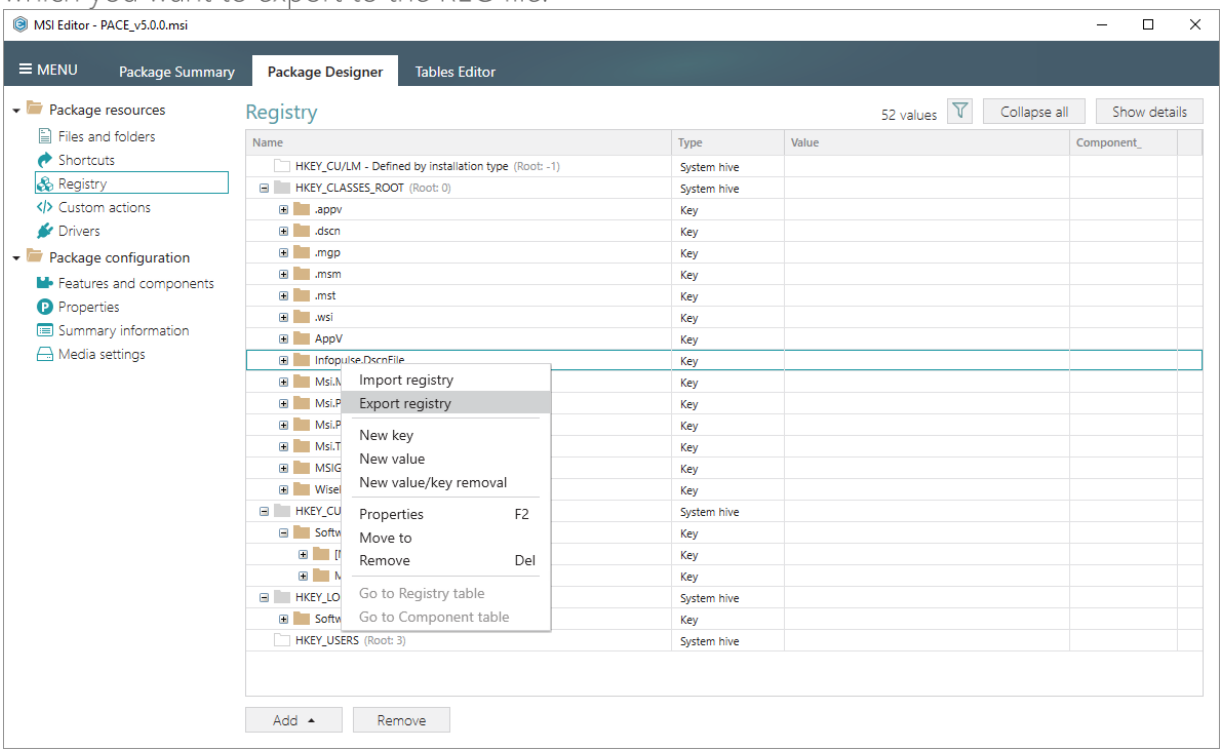
### 3.7.4.5 Export Registry

Export the selected registry to the REG file.

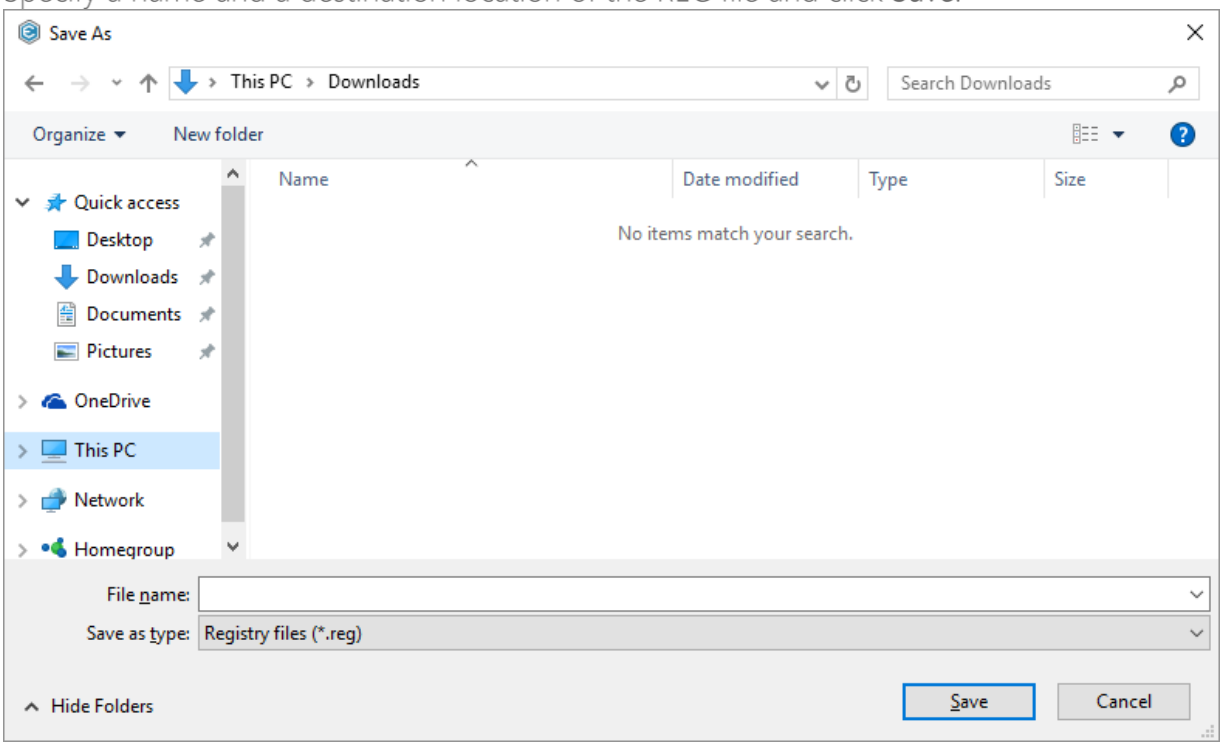
- Go to the **Registry** tab of the **Package Designer**.




- 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.



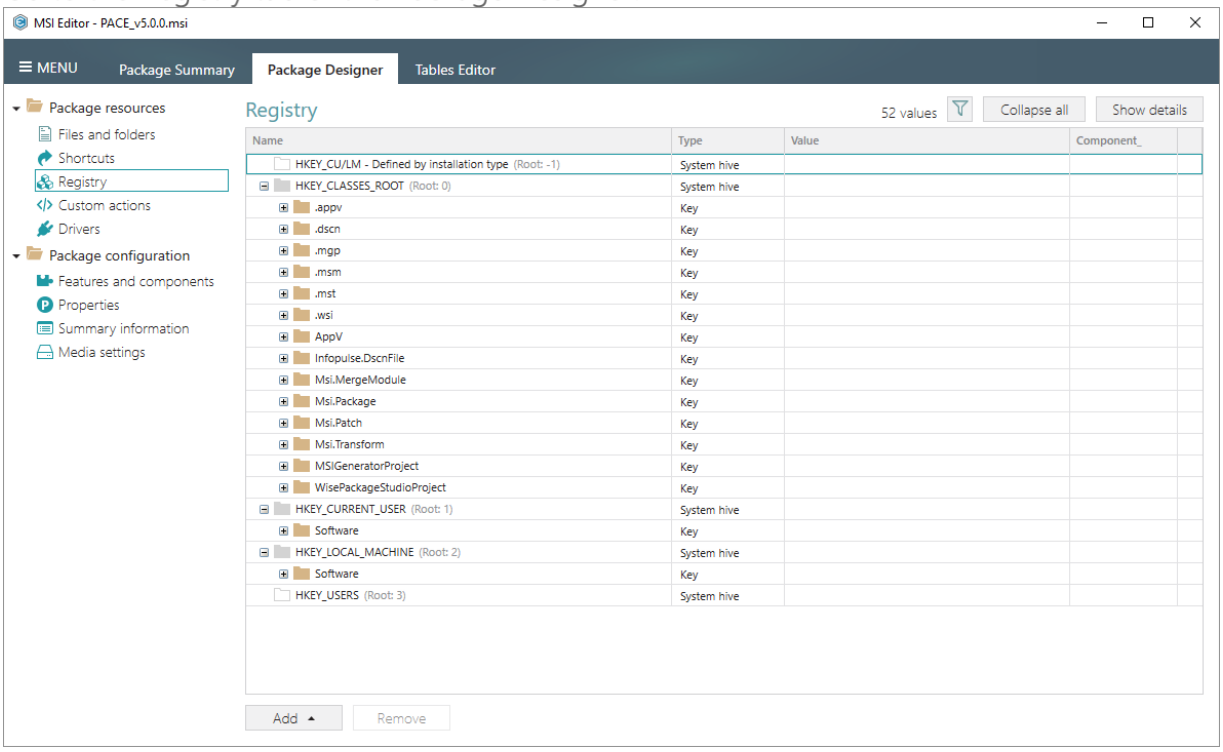
- Specify a name and a destination location of the REG file and click Save.



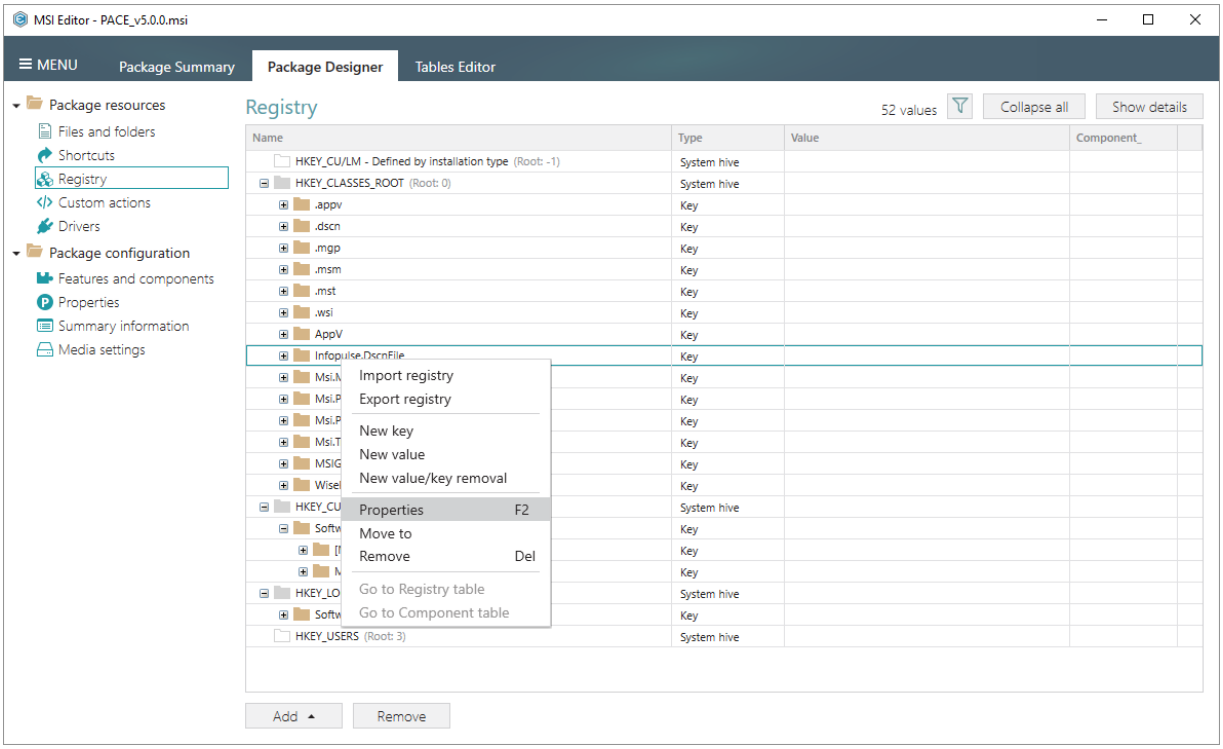

### 3.7.4.6 Set Permissions

Set custom permissions for the registry key.

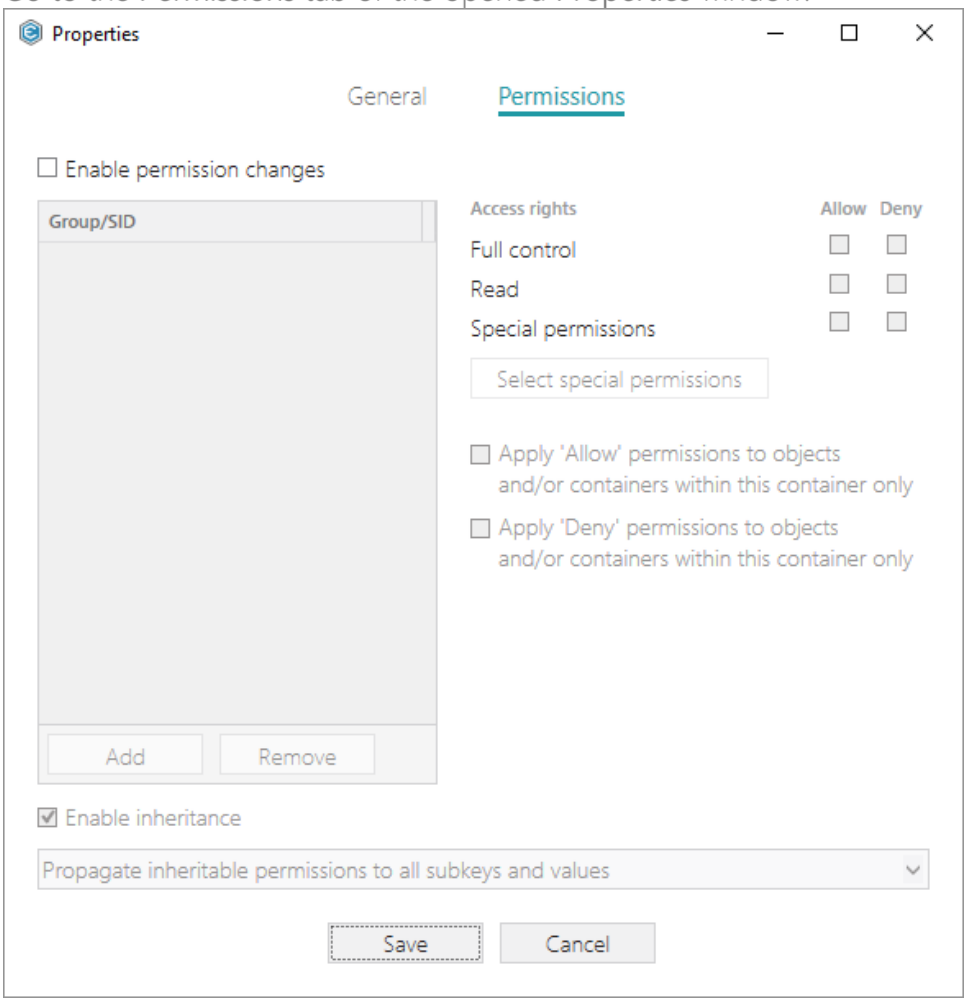
- Go to the Registry tab of the Package Designer.



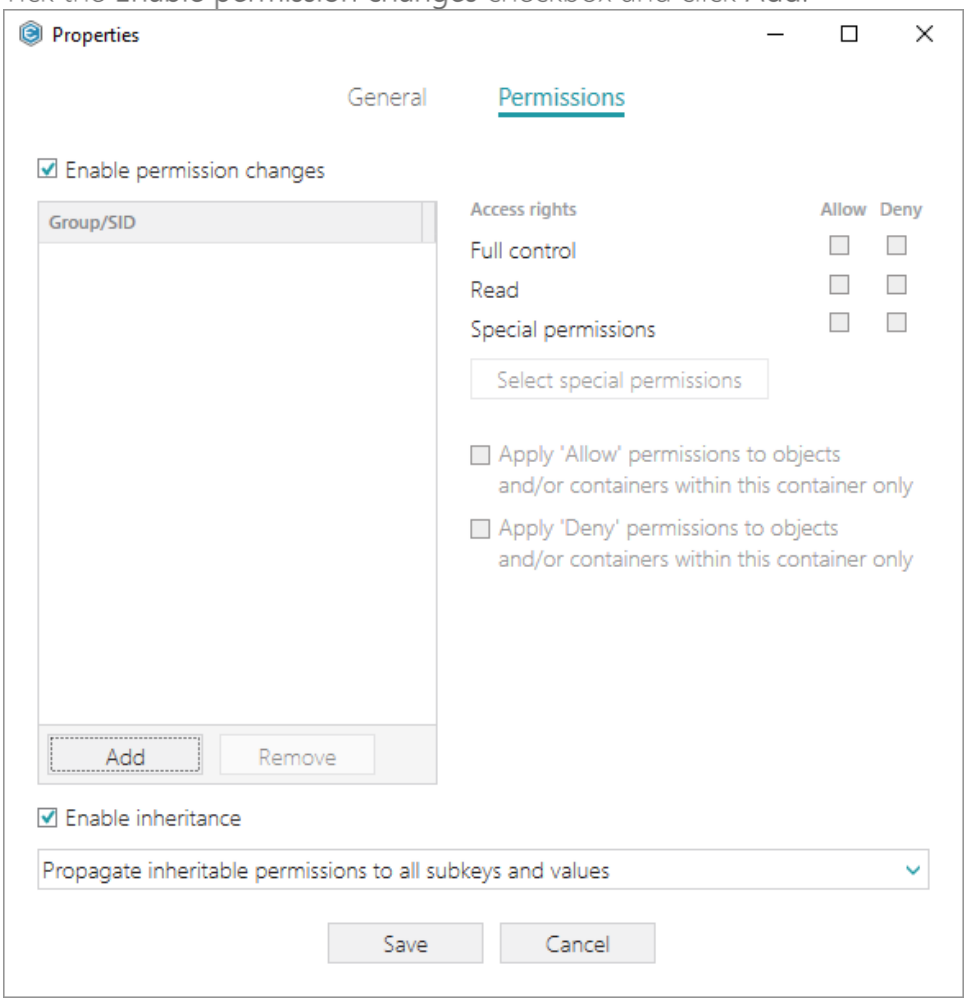
- Select **Properties** from the context menu of a registry key, for which you want to set permissions. Note that permissions could not be changed for the registry values and root hives.



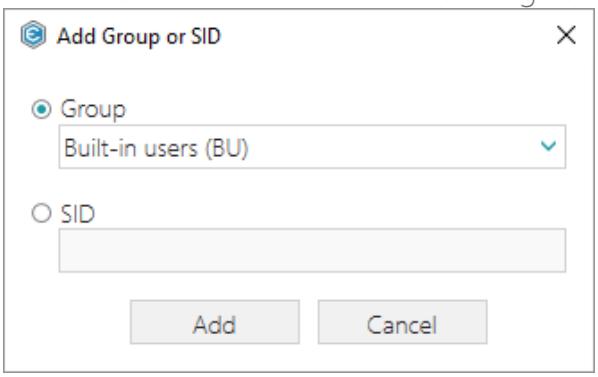

- Go to the **Permissions** tab of the opened **Properties** window.



- Tick the Enable permission changes checkbox and click Add.

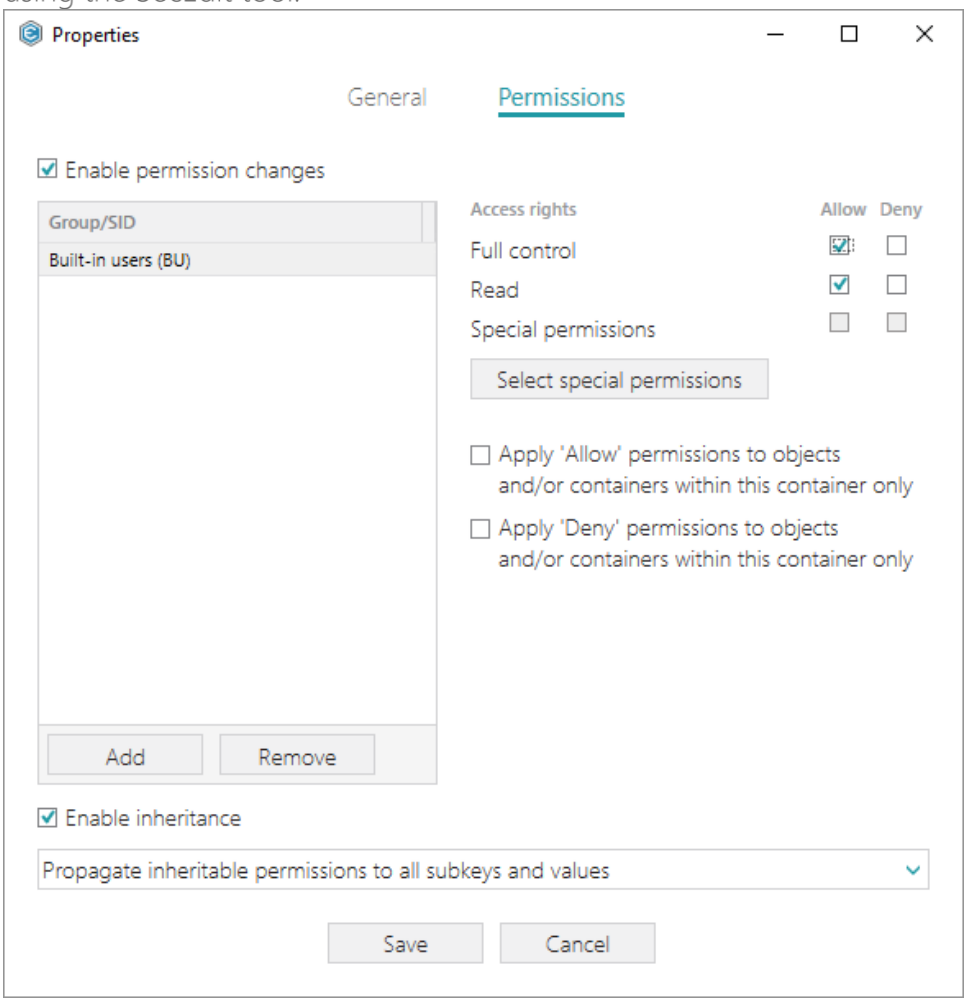



- Select a group from the list or enter SID manually, for which you want to set permissions, and click Add. Note that you can add as many groups and SID as you need. Each group and SID could have individual access rights.

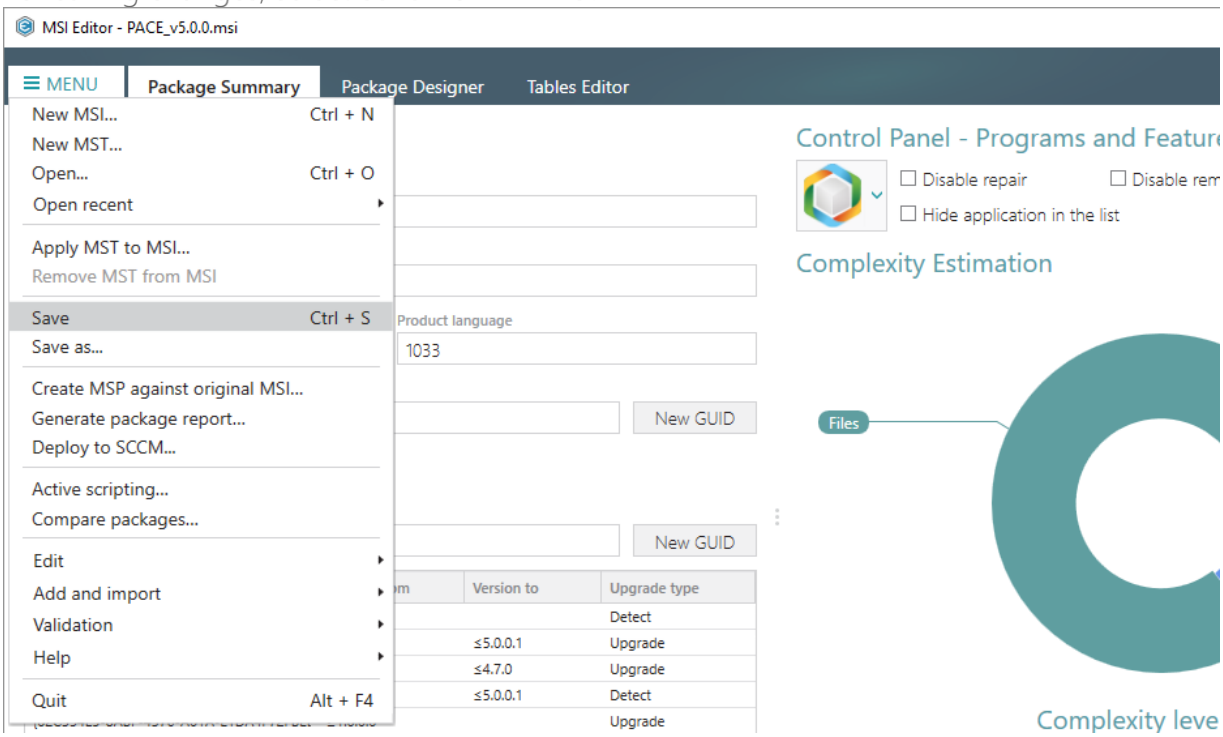



- Tick the necessary access rights for the selected group or SID, change (if needed) special permissions, inheritance options and behavior of applying the permissions. Click Save to save the changed permission settings to MSI. Note that permissions will be propagated

using the SecEdit tool.



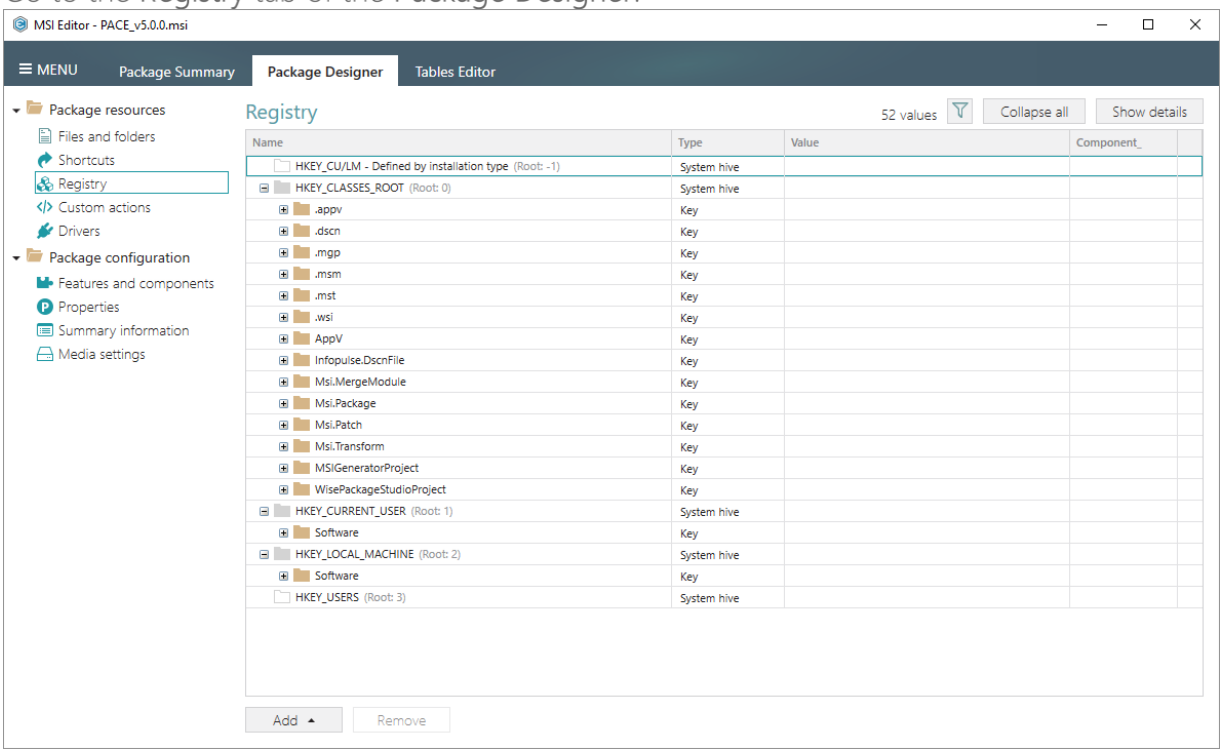

- For saving changes, select Save from MENU.



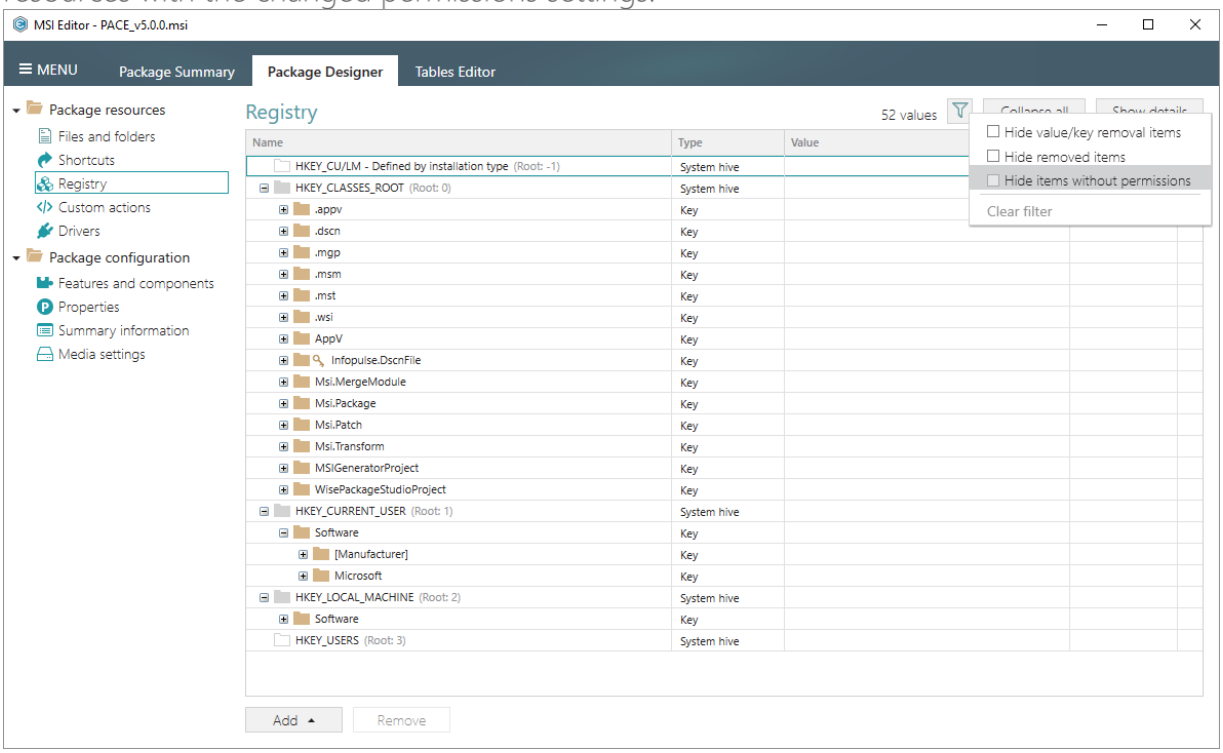
### 3.7.4.7 Remove Permissions

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

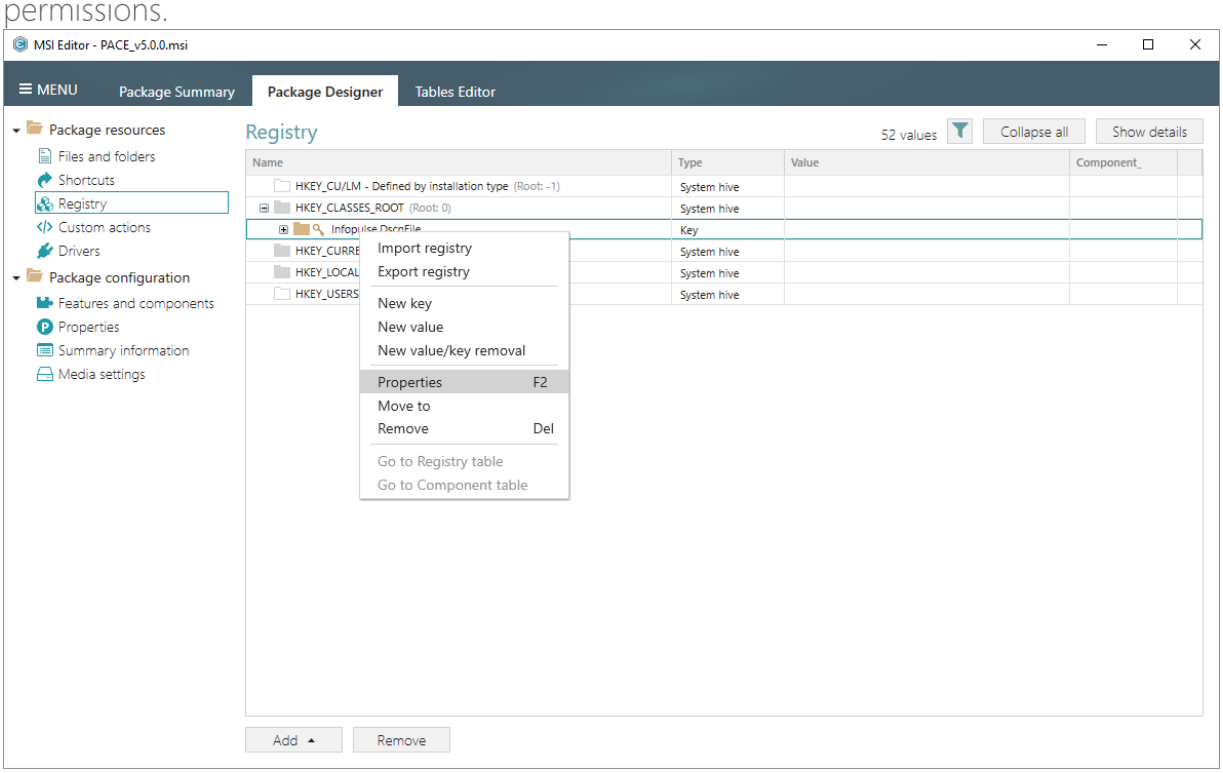
- Go to the **Registry** tab of the **Package Designer**.



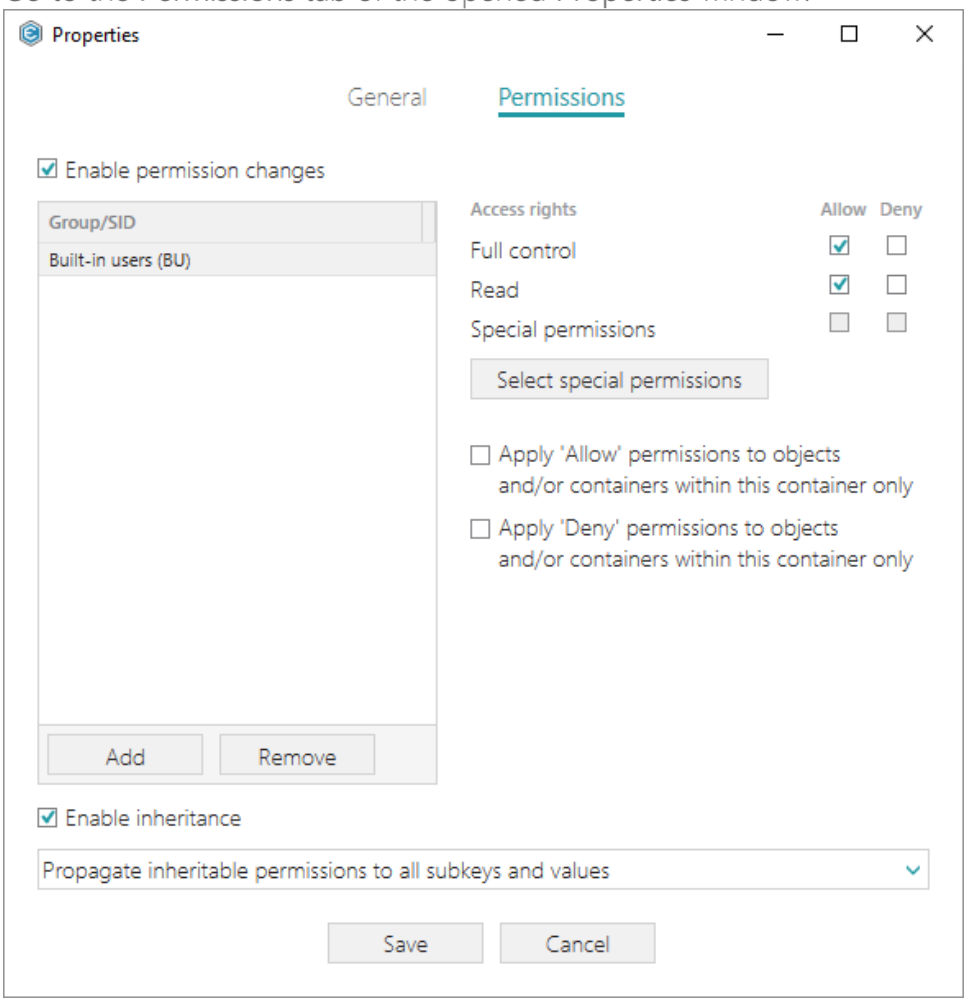

- (Optional) Click on the **Filter** and apply the **Hide items without permissions** filter to locate resources with the changed permissions settings.



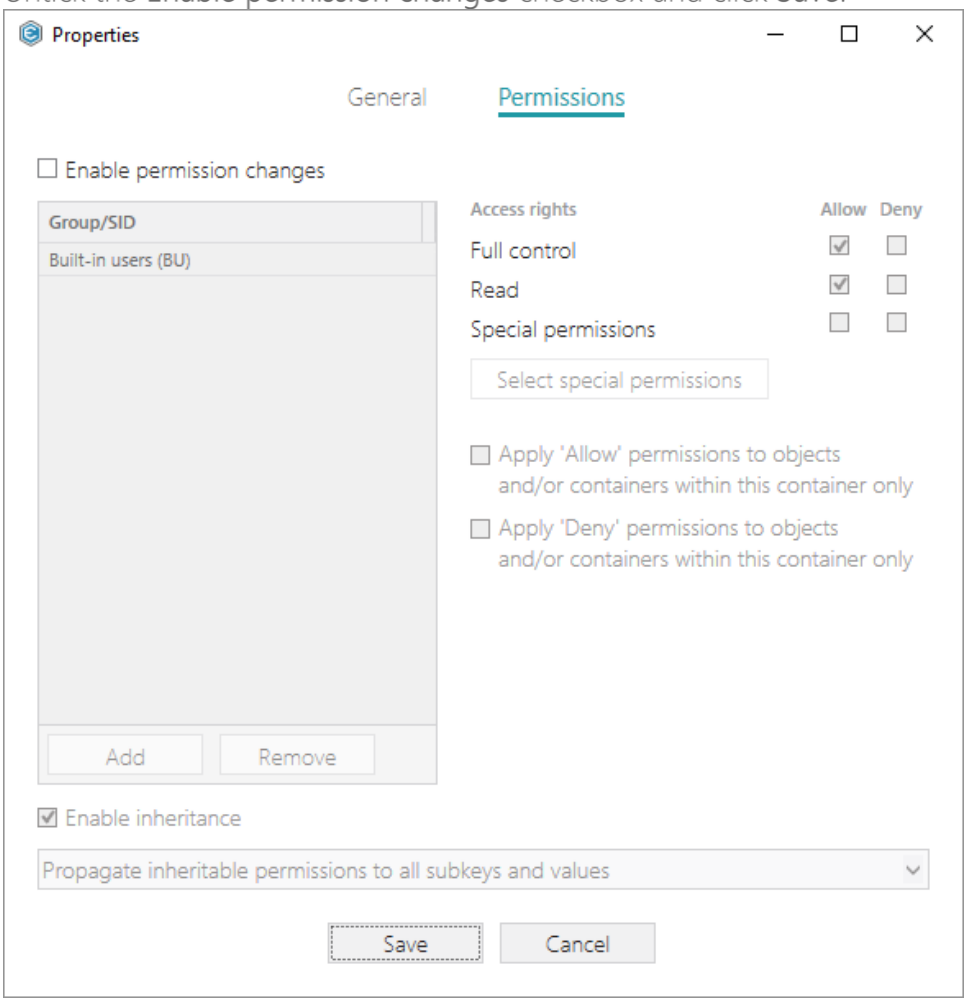
- Select **Properties** from the context menu of a registry key, for which you want to remove permissions.



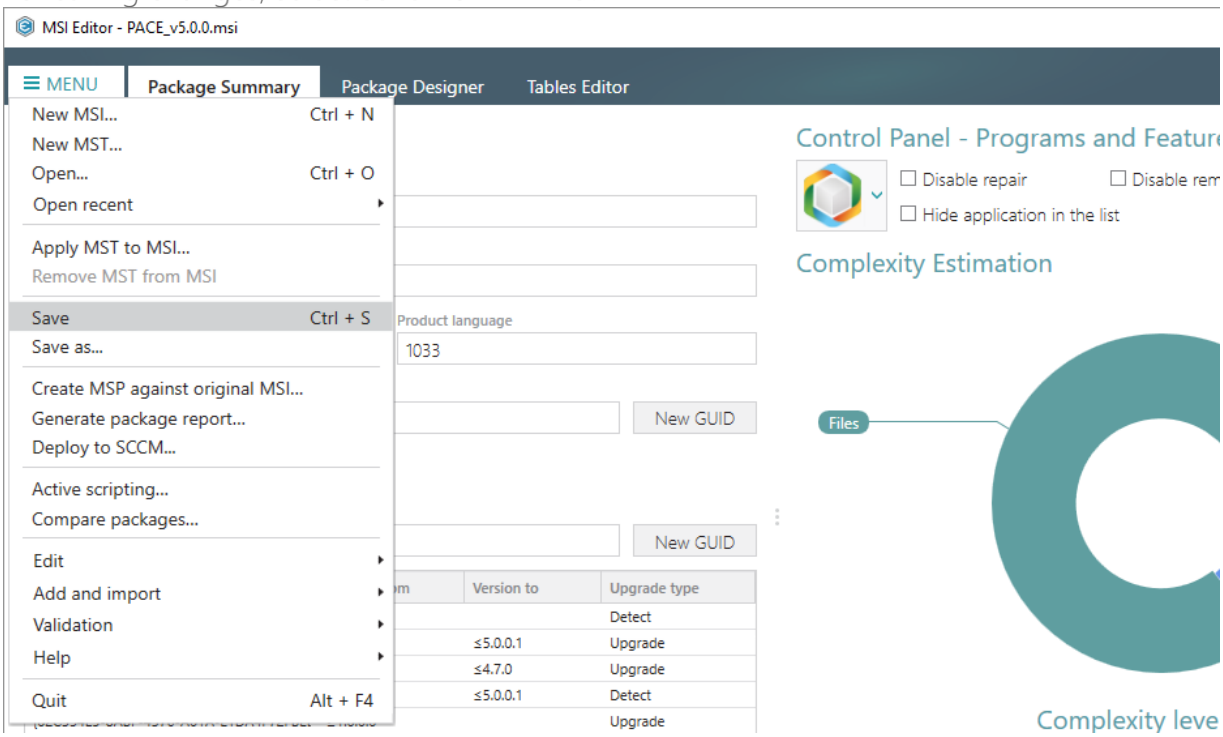

- Go to the **Permissions** tab of the opened **Properties** window.



- Untick the **Enable permission changes** checkbox and click **Save**.



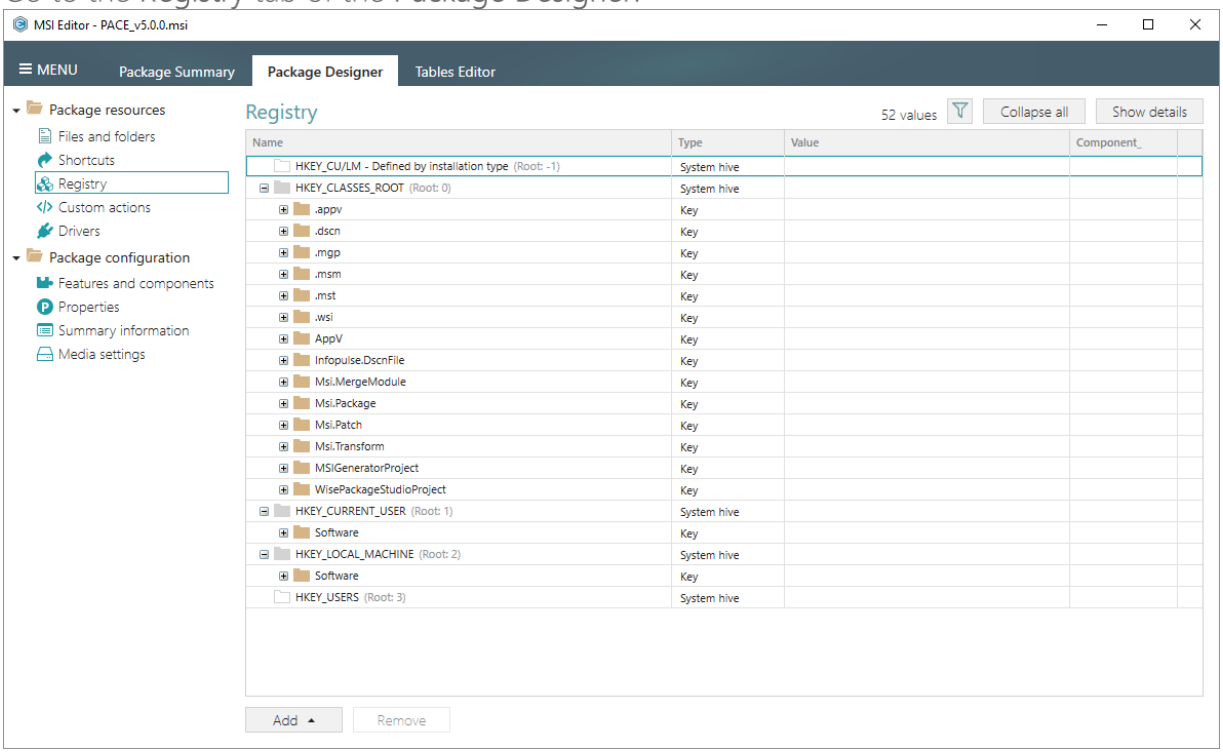

- For saving changes, select **Save** from **MENU**.



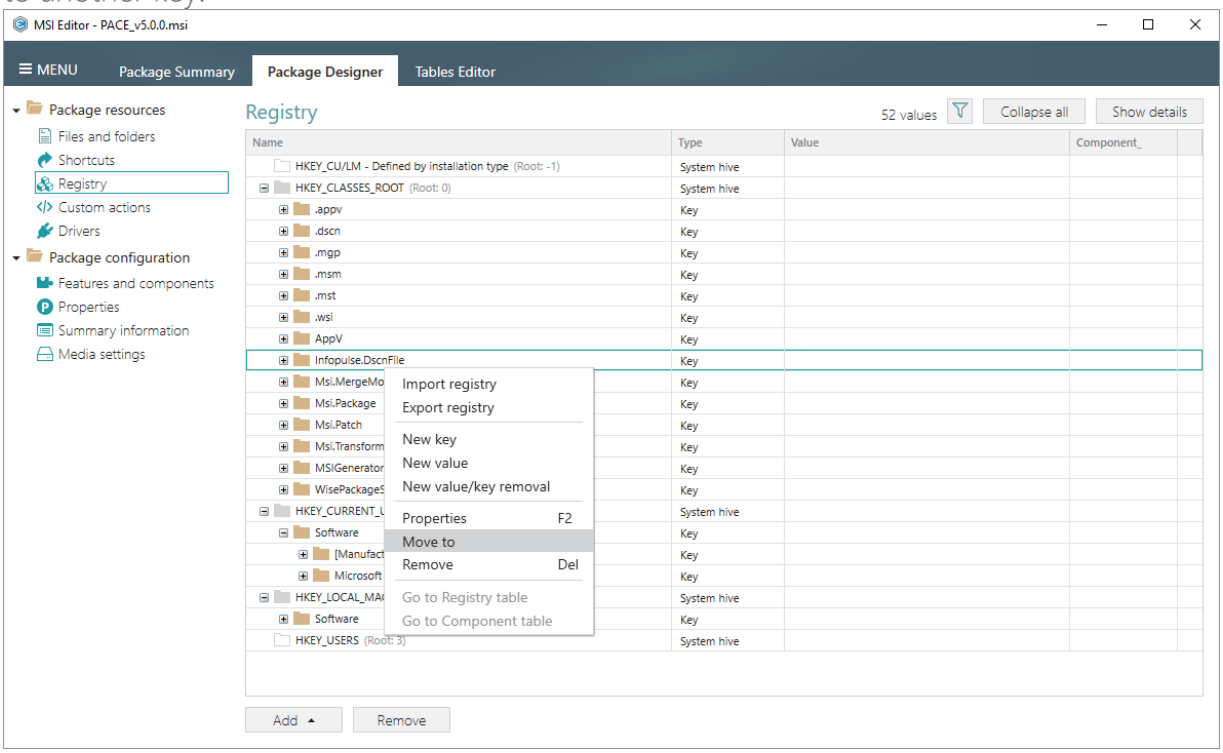
### 3.7.4.8 Move Value, Key

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

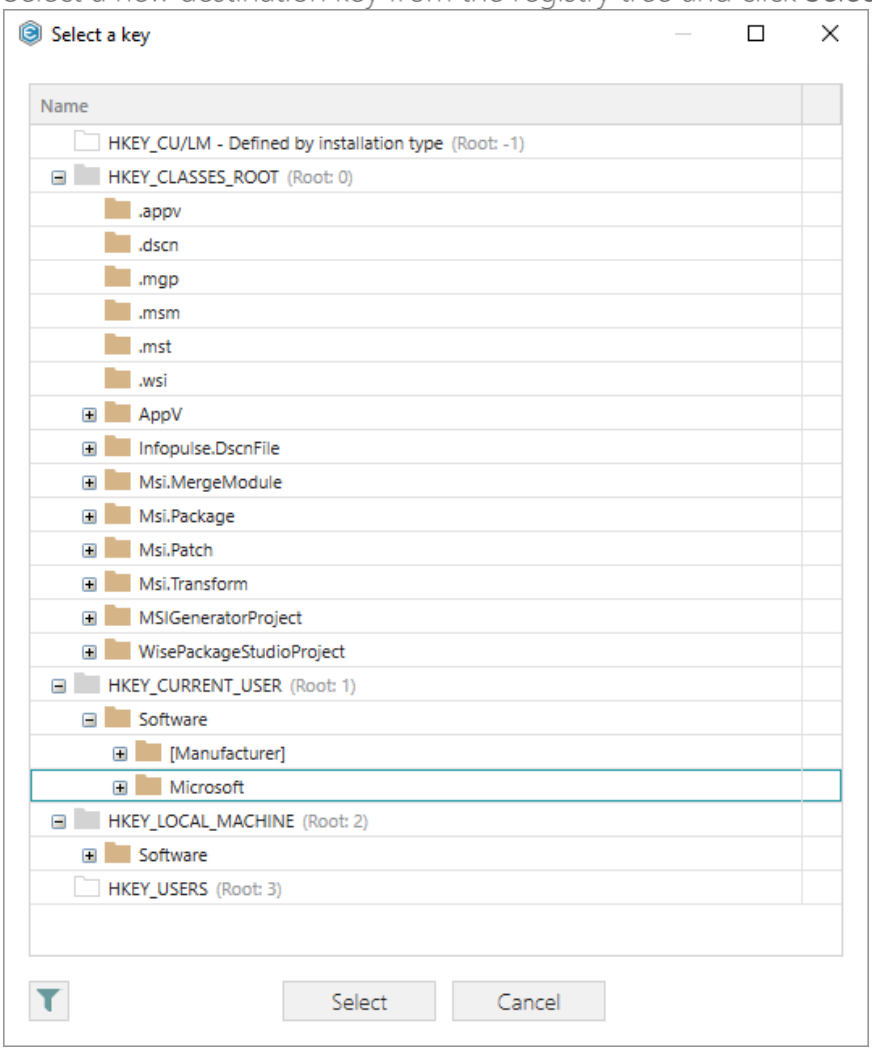
- Go to the Registry tab of the Package Designer.



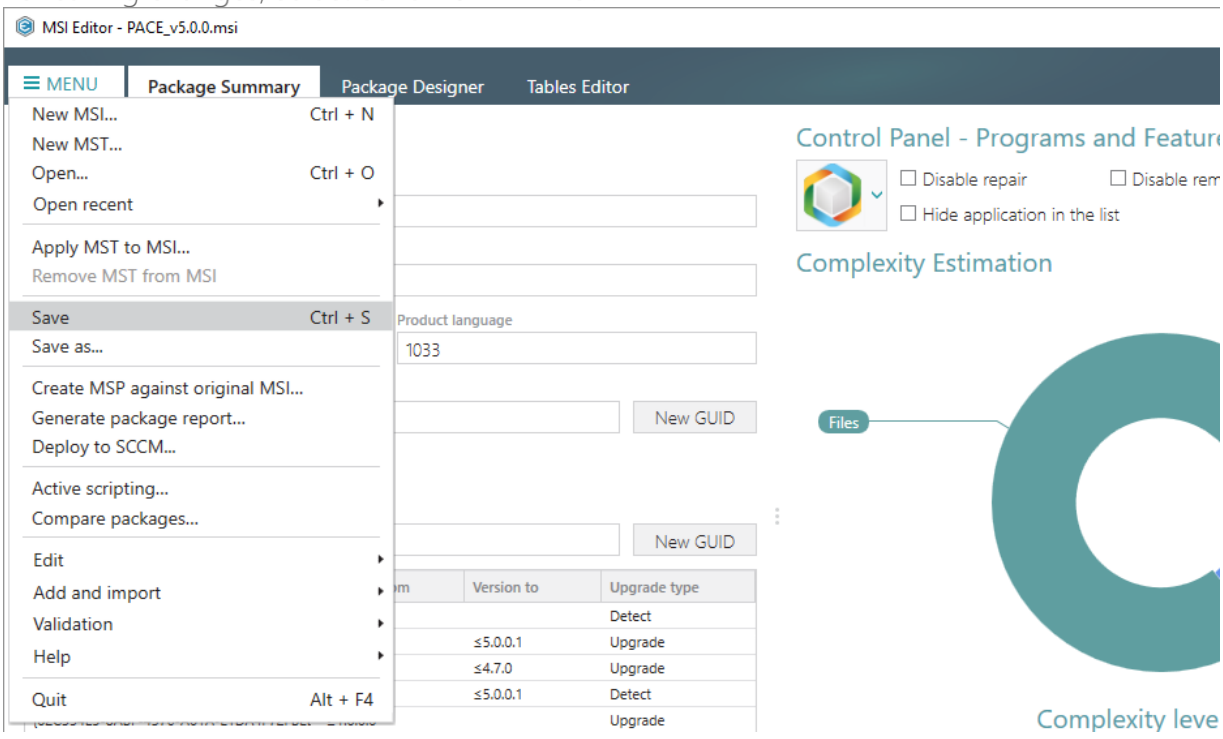

- Select Move to from the context menu of a registry key or a value, which you want to move to another key.



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



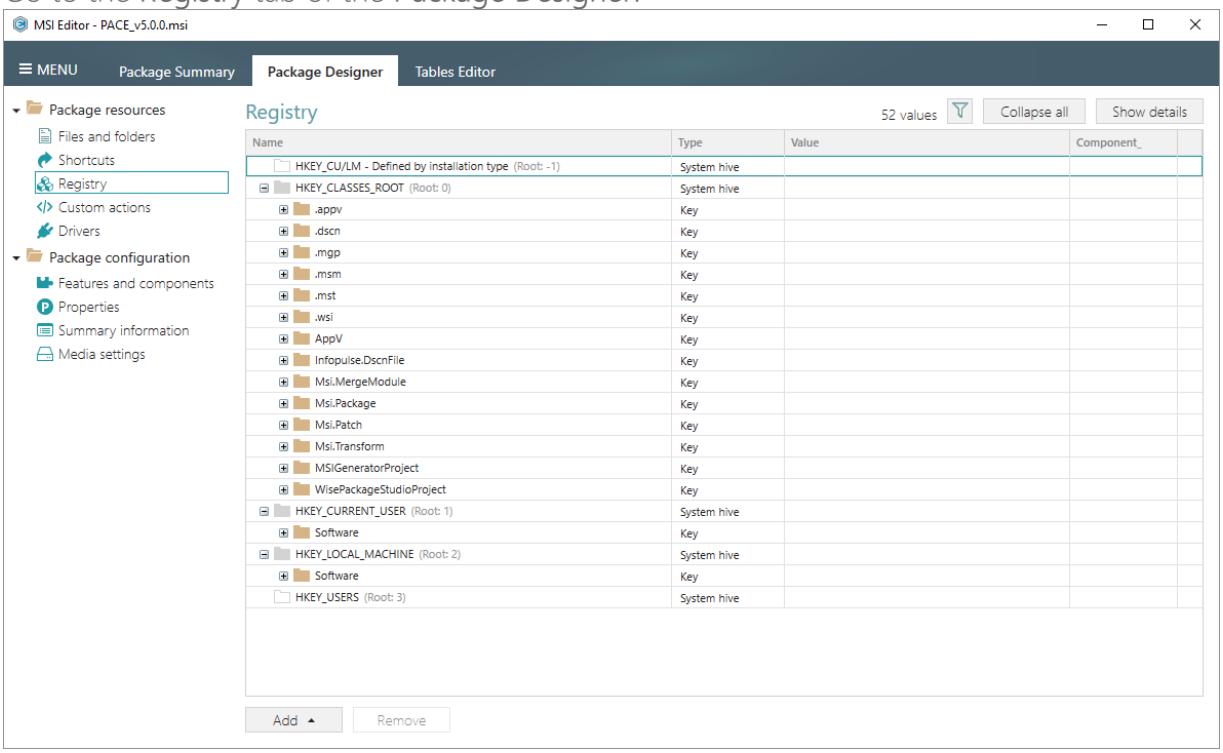

- For saving changes, select **Save** from **MENU**.



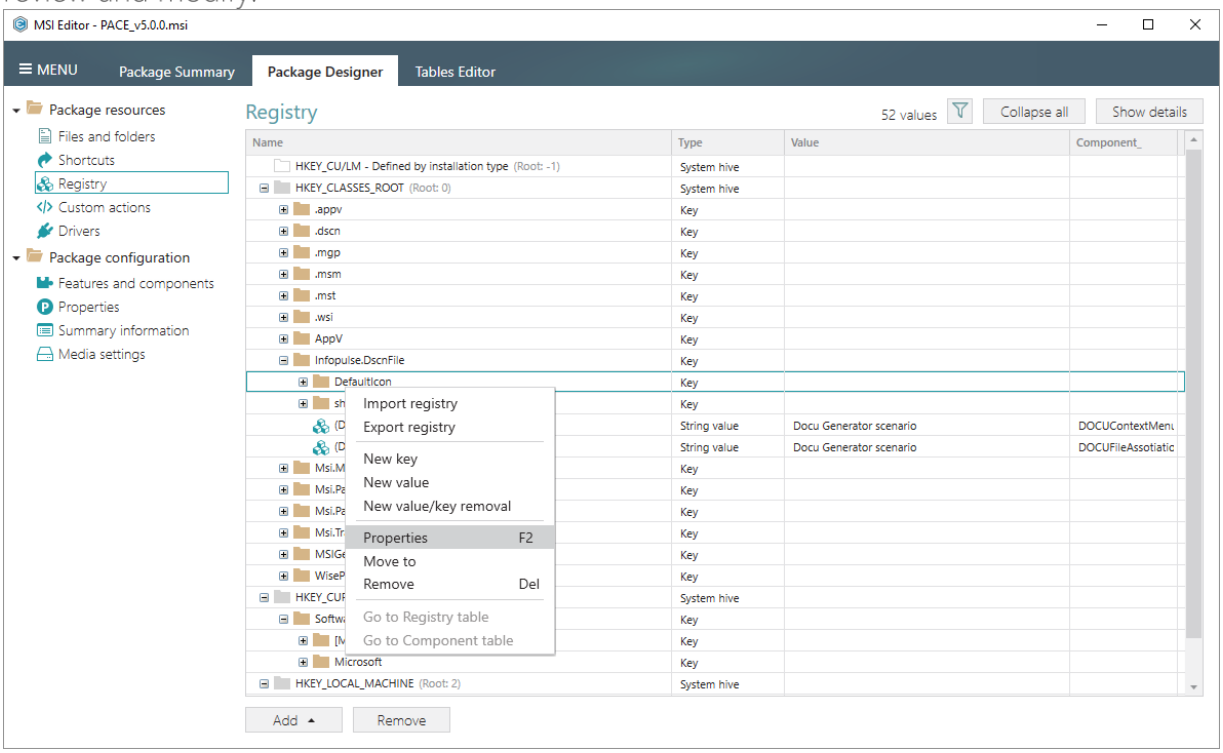
### 3.7.4.9 Edit Registry Key

Modify details and options of the selected registry key.

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



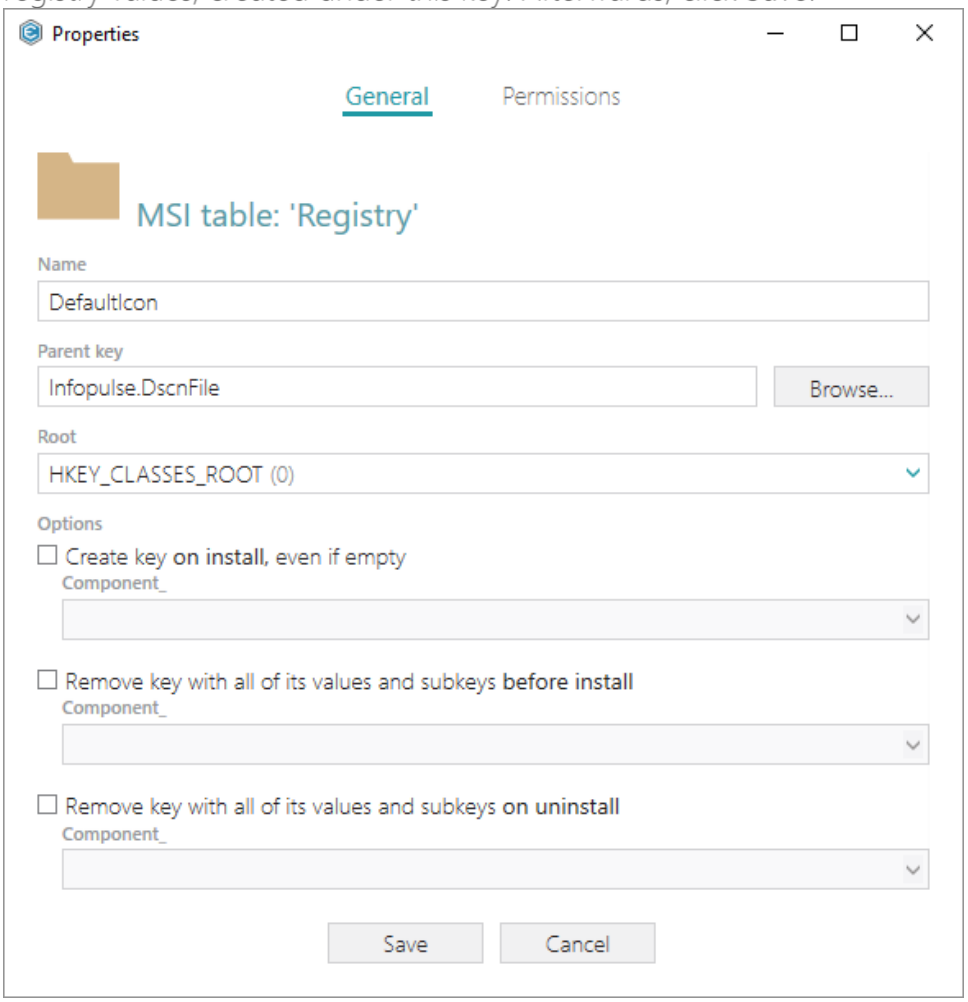

- Select **Properties** from the context menu of a registry key, details of which you want to review and modify.



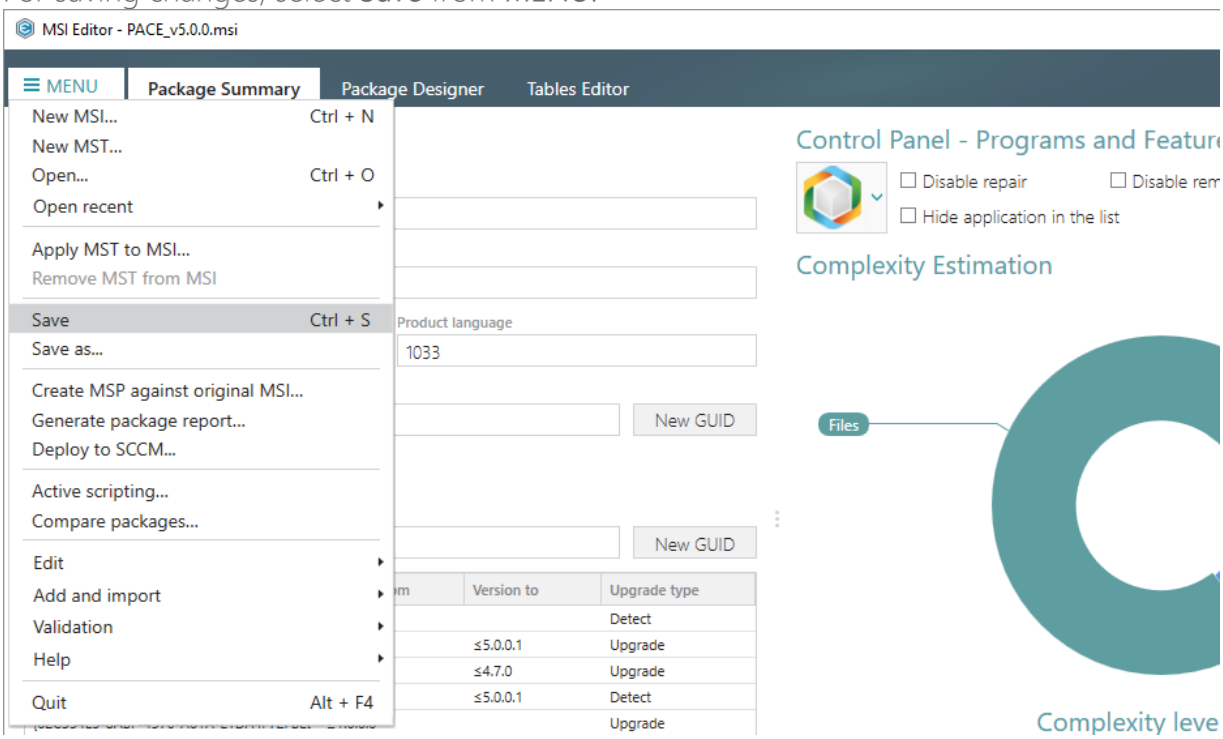

- Using the **Properties** window make the necessary changes to the key **Name**, its **Parent key** and **Root** hive. You can modify the **Parent key** directly in the text field or use **Browse** to choose a new key from the registry tree. Note that changing the **Parent key** and/or the

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.



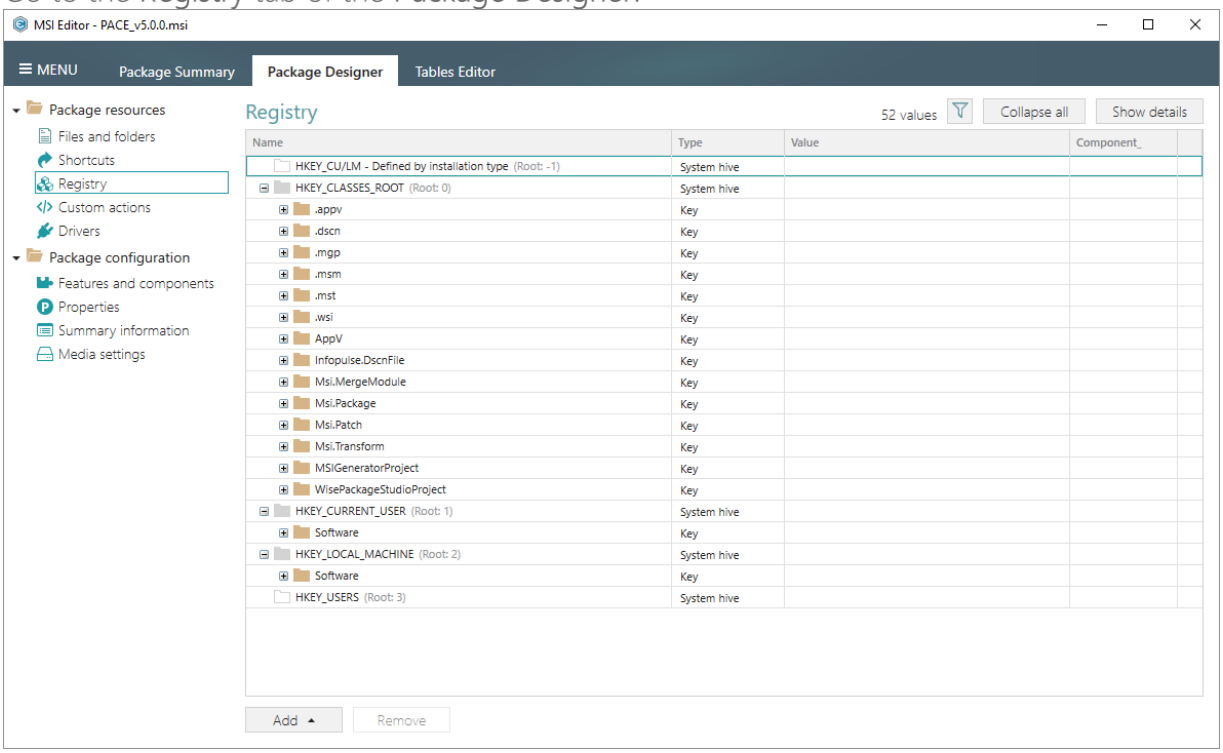
- For saving changes, select Save from MENU.



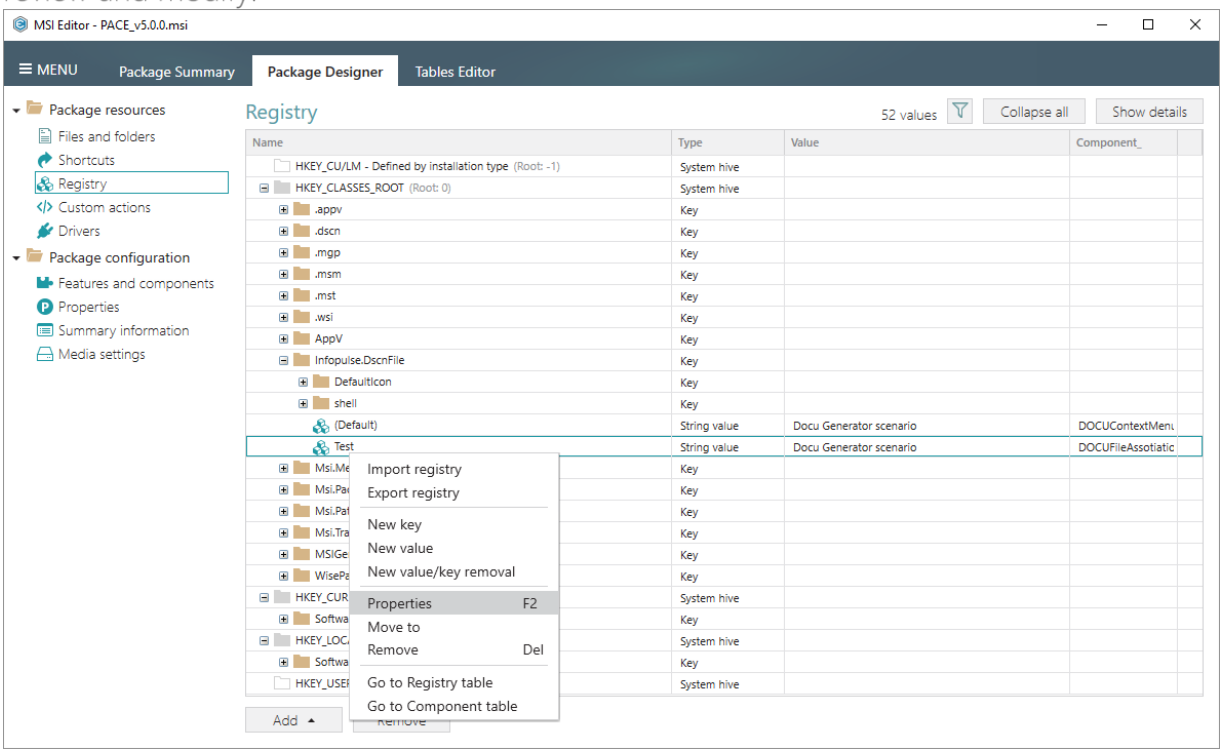

## 3.7.4.10 Edit Registry Value

Modify details of the selected registry value.

- Go to the Registry tab of the Package Designer.



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



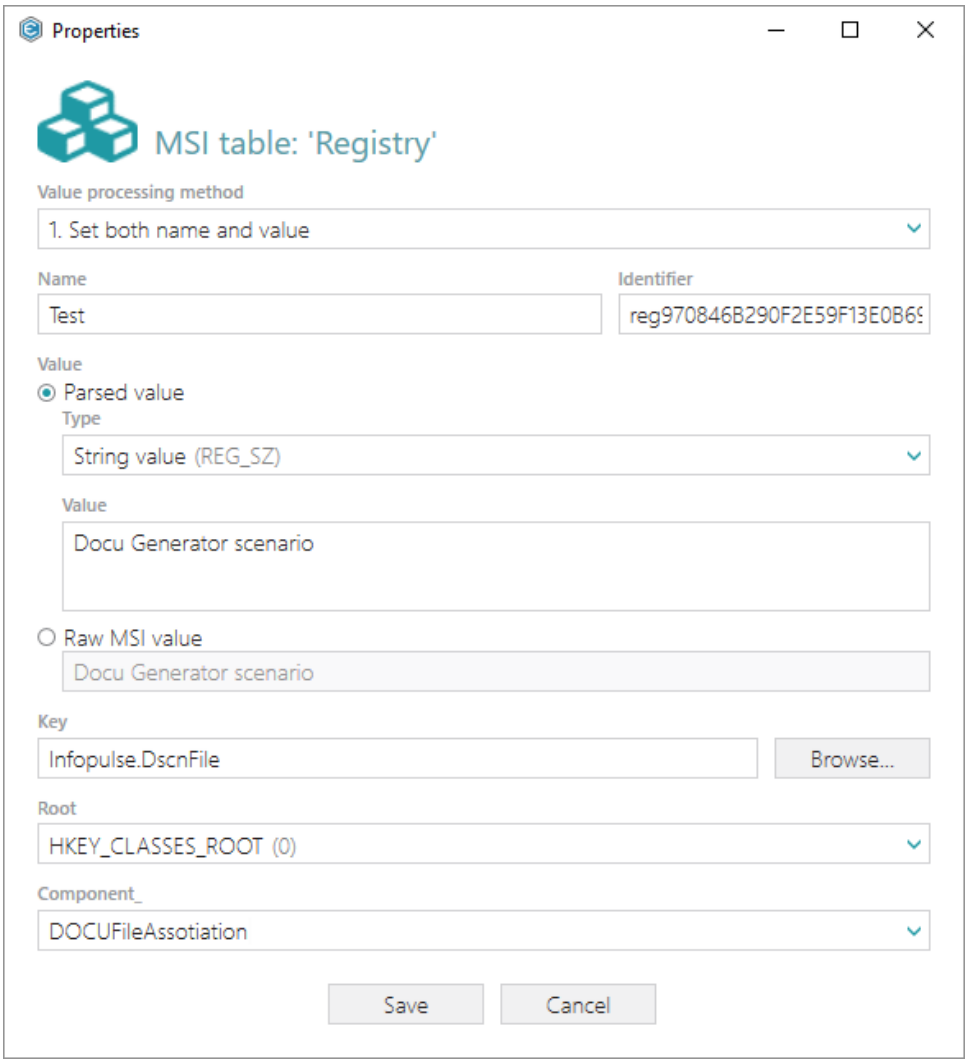

- Using the **Properties** window make the necessary changes to the value **Name**, **Value** itself, and **Component** that controls the installation of this registry value.

  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**.



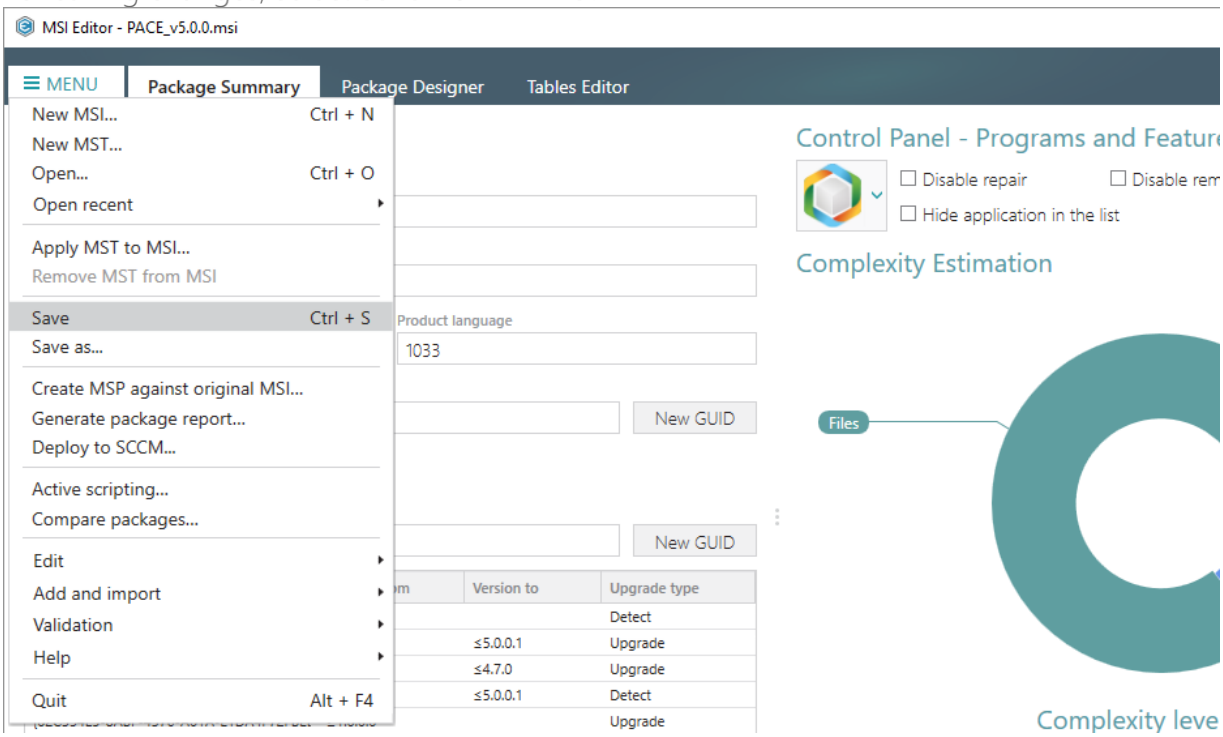

## Value processing method

| | |
|---|---|
| 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.<br><br>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)":<br><br>Name: (Default) \| Type: REG_SZ \| Data: Test<br><br>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.<br><br>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.<br><br>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.<br><br>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 3.7.4.3

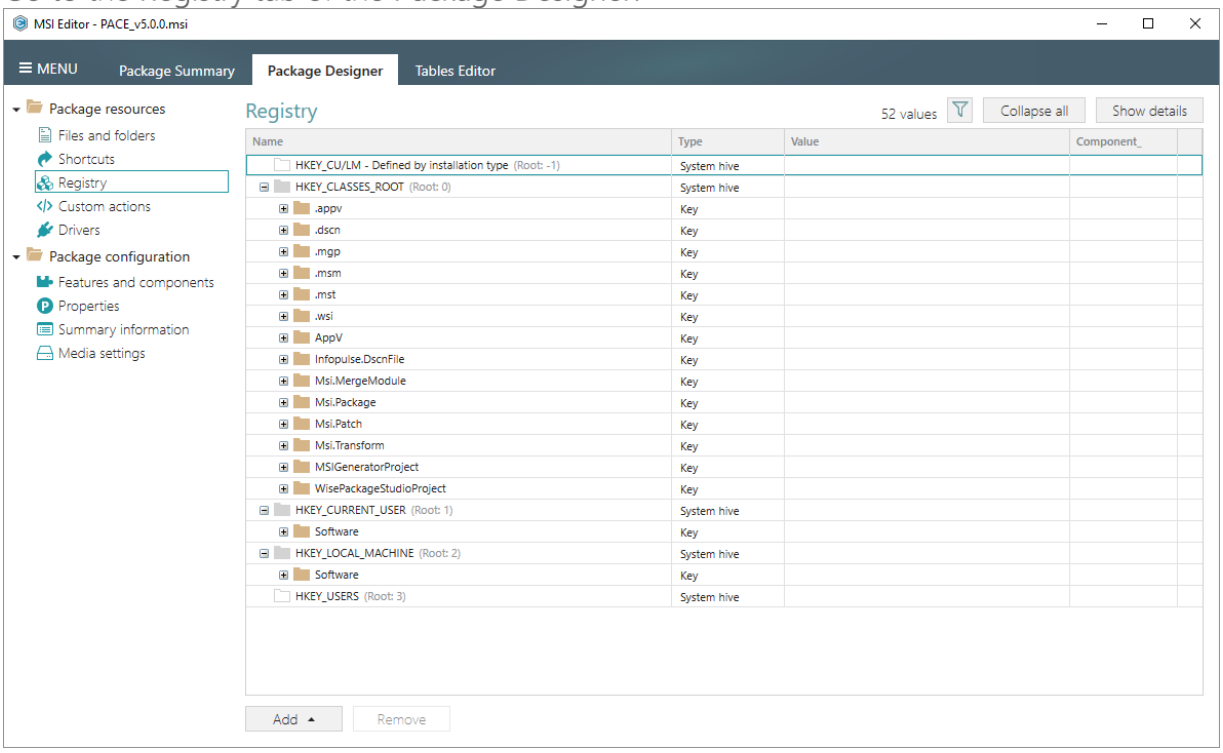
- For saving changes, select Save from MENU.



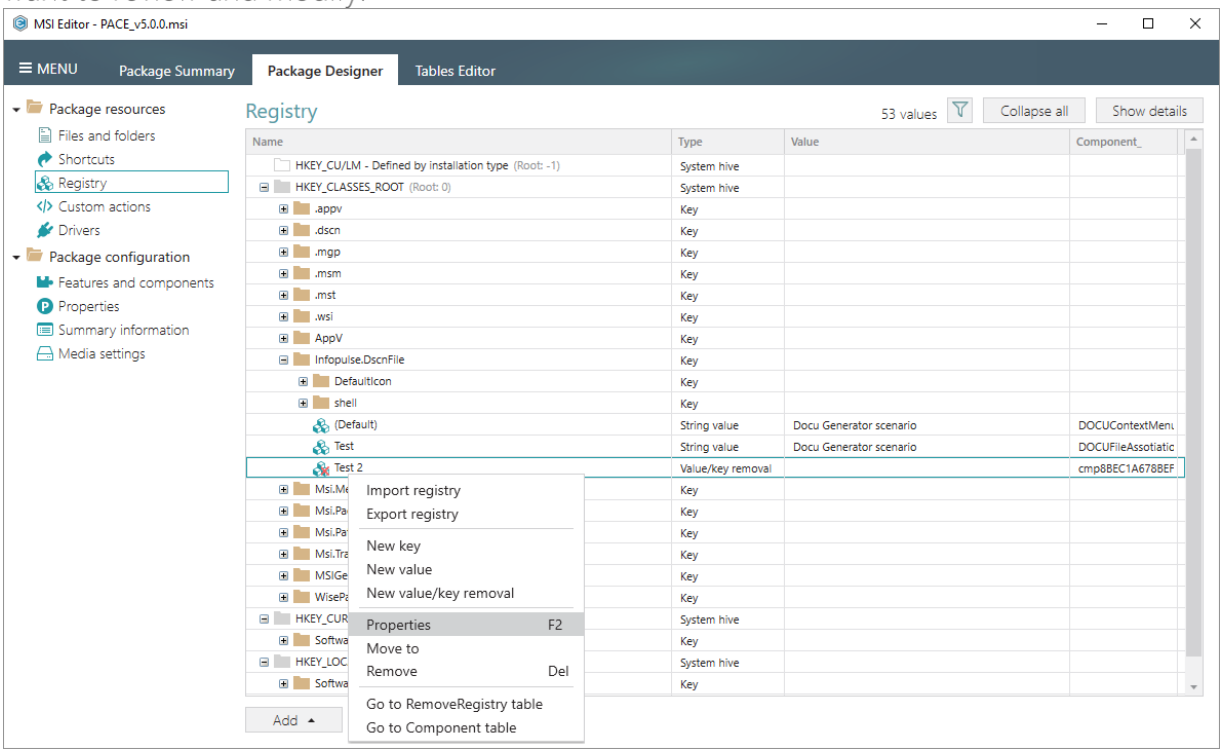
### 3.7.4.11 Edit Value/Key Removal

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 **Registry** tab of the **Package Designer**.




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

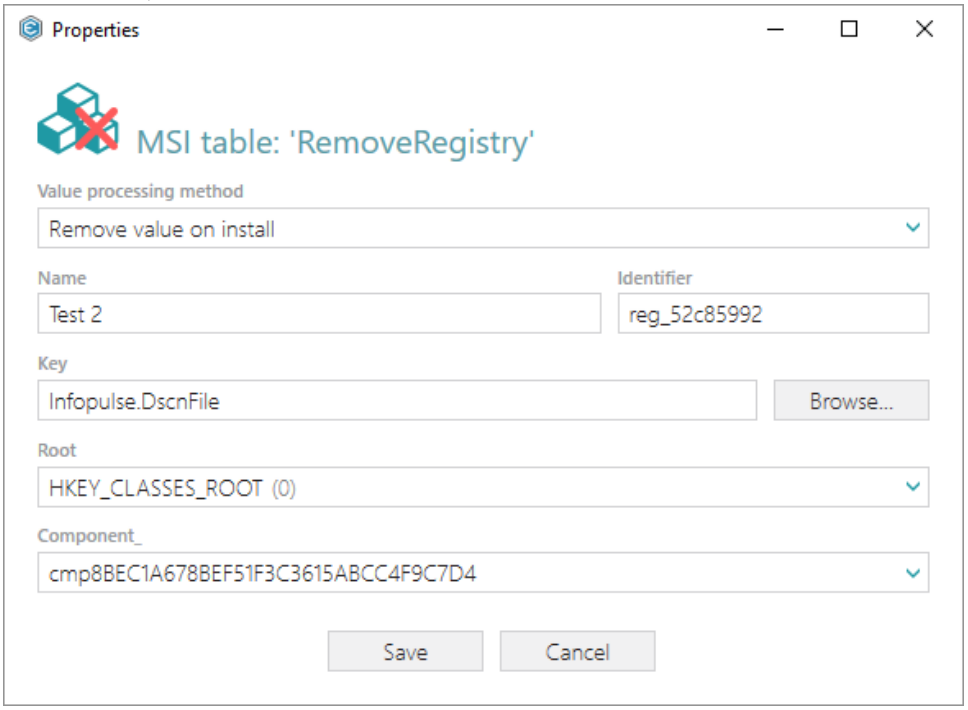



- Using the **Properties** window make the necessary changes to a value **Name**, its **Key**, **Root** hive, and **Component** that controls the deletion of this registry value. Note that the

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.

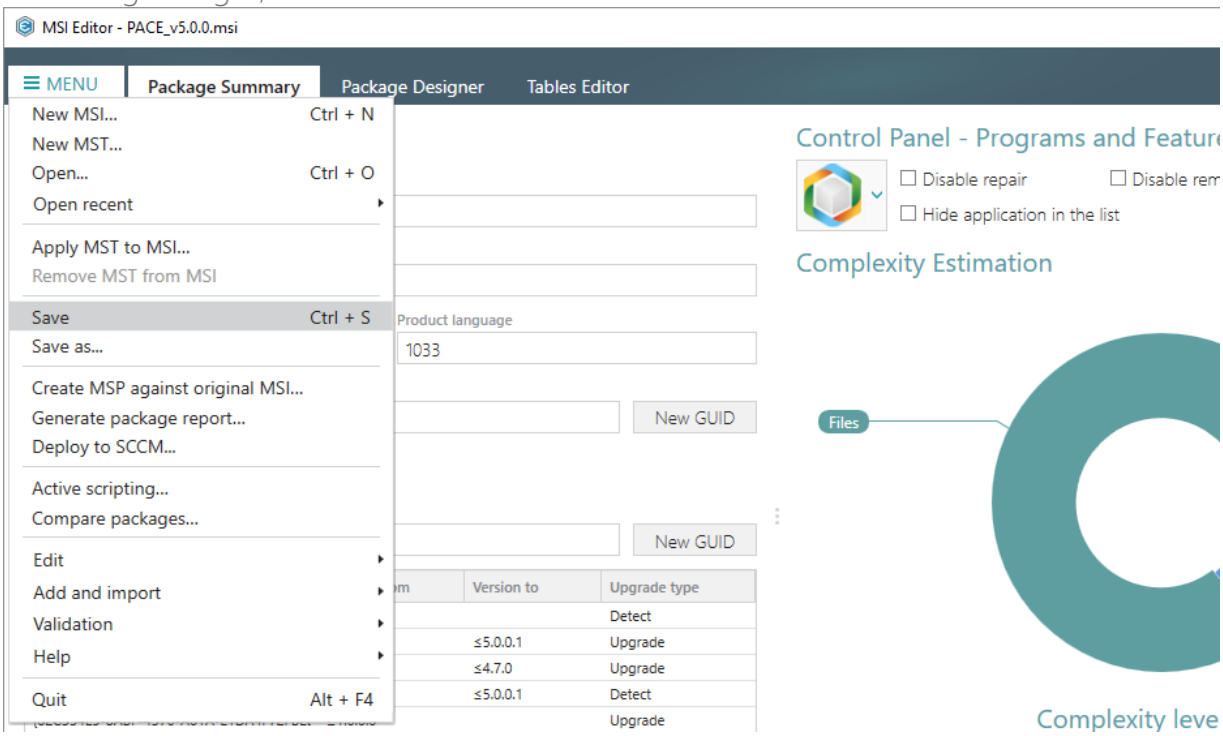
Afterwards, click Save.




## 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.<br><br>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)":<br>Name: (Default) \| Type: REG_SZ \| Data: (value not set)<br><br>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.<br><br>On uninstall, no additional actions take place. |

- For saving changes, select Save from MENU.




## 3.7.5 Custom Actions

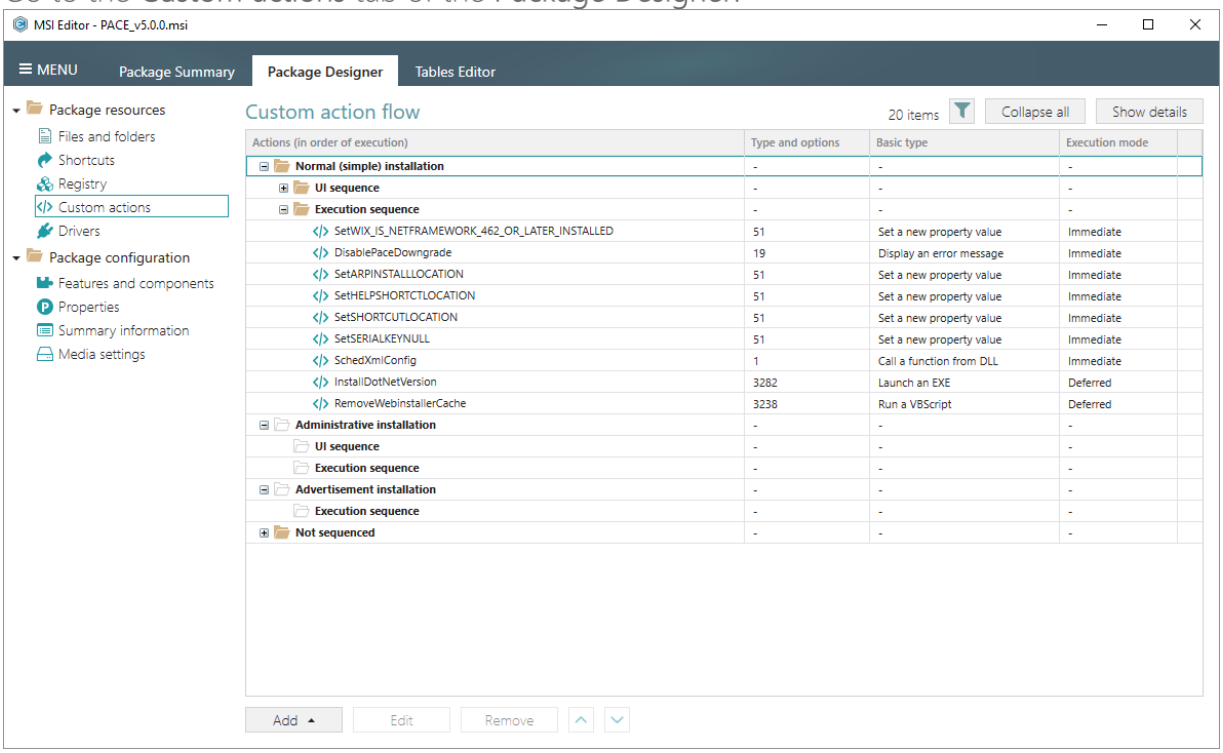
Choose a scenario that better suits your needs:

- Add Standard Actions, described in section 3.7.5.1
  Add new standard actions to the necessary sequences.
- Add Custom Action, described in section 3.7.5.2
  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.
- Edit Custom Action, described in section 3.7.5.3
  Edit basic type, resources, sequence and other settings of the existing custom action.

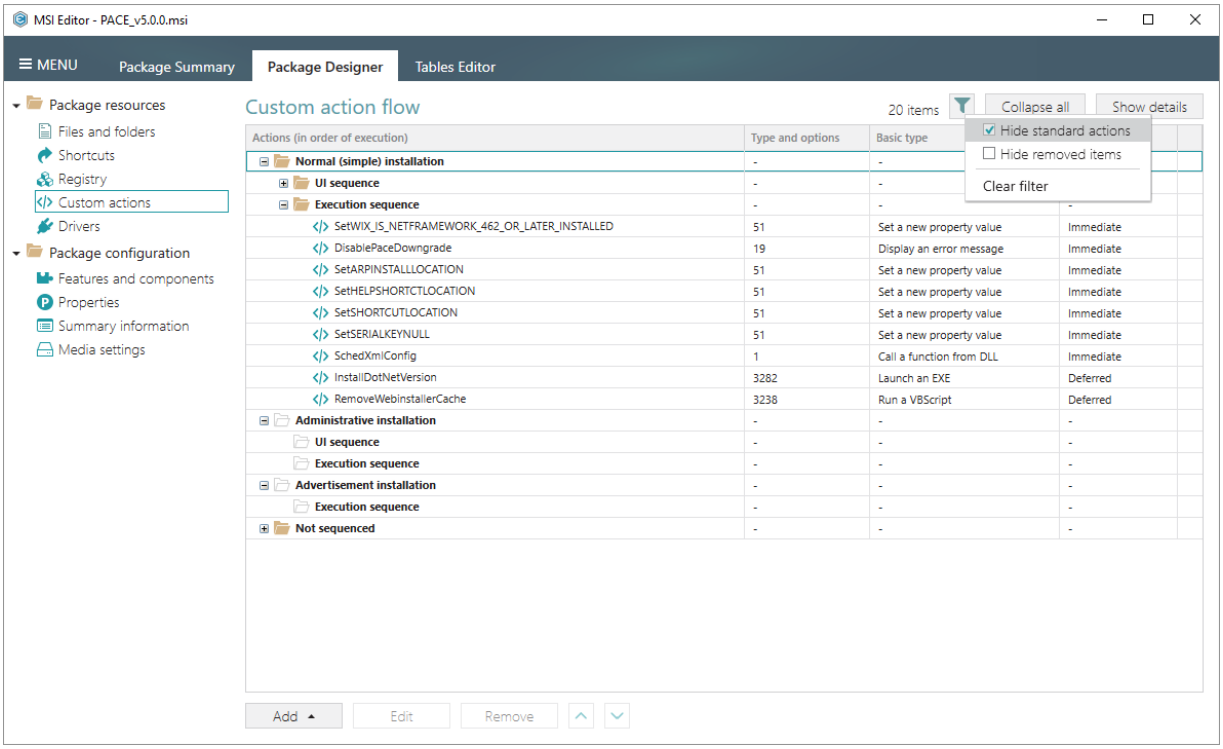
### 3.7.5.1 Add Standard Actions

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

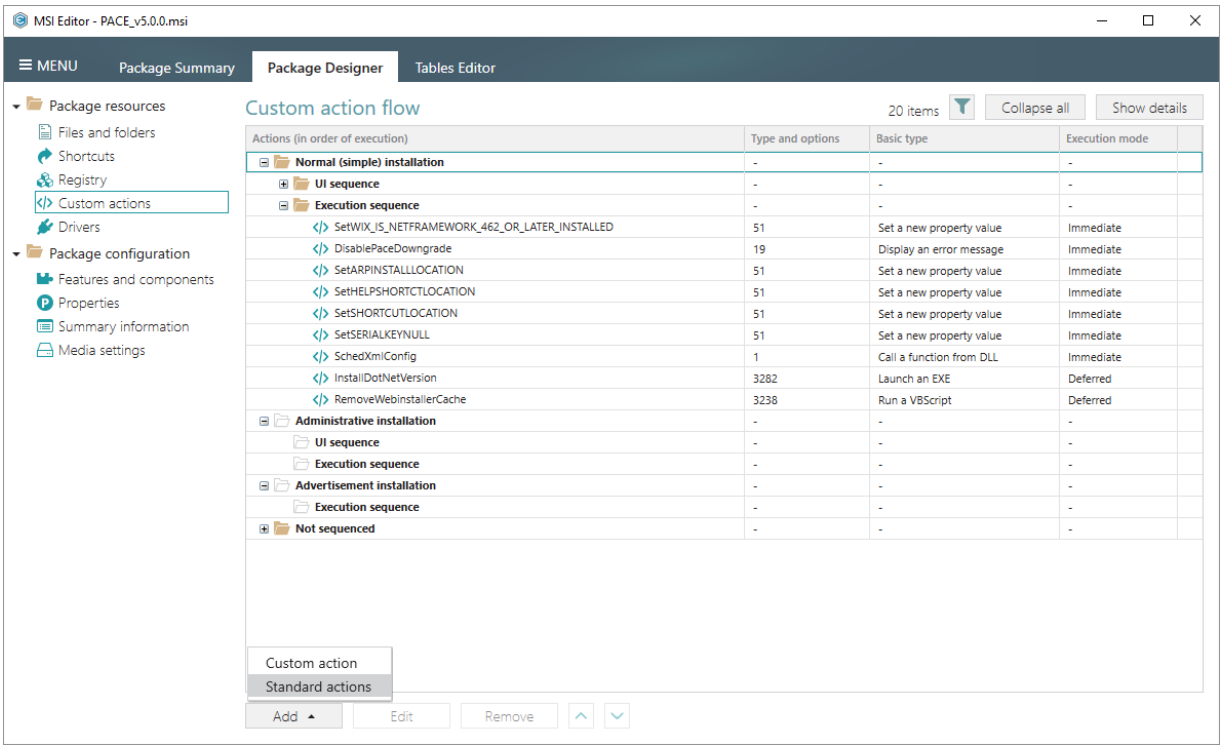
- Go to the Custom actions tab of the Package Designer.



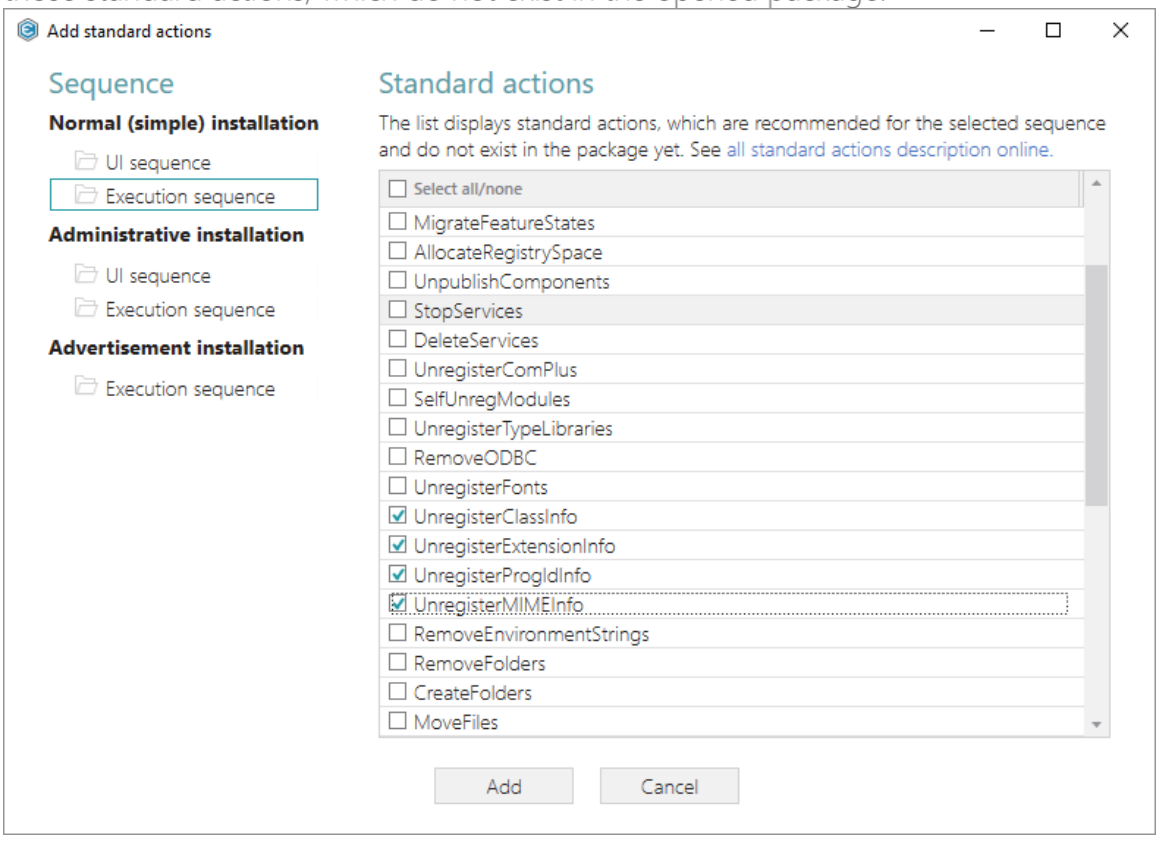

- (Optional) To review the existing standard actions, hidden by default, click on the and untick the 'Hide standard actions' filter.



- Select Add -> Standard actions from the toolbox.

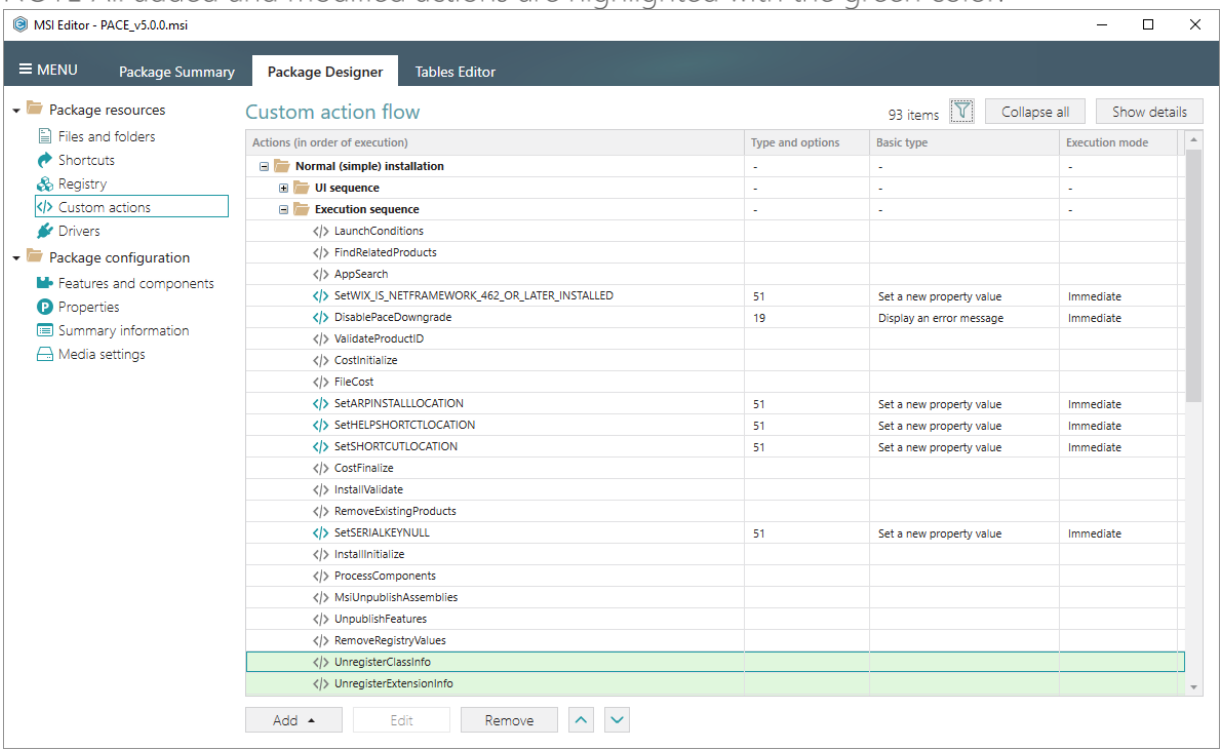



- Tick standard actions you want to add and click Add. Note that you can add many actions to the different installation sequences, displayed on the left, at once. The list displays only those standard actions, which do not exist in the opened package.

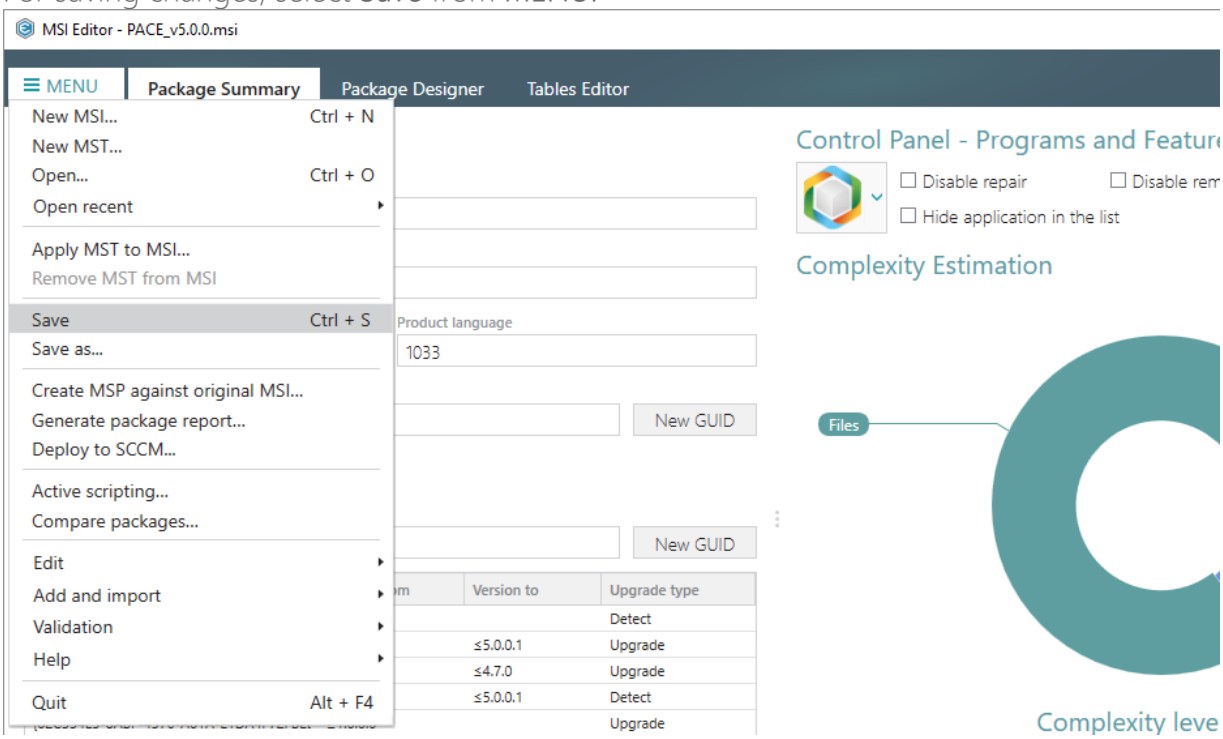



- Find the standard actions of the package by expanding the installation sequences. The standard actions are displayed using the grey color icon.

NOTE All added and modified actions are highlighted with the green color.



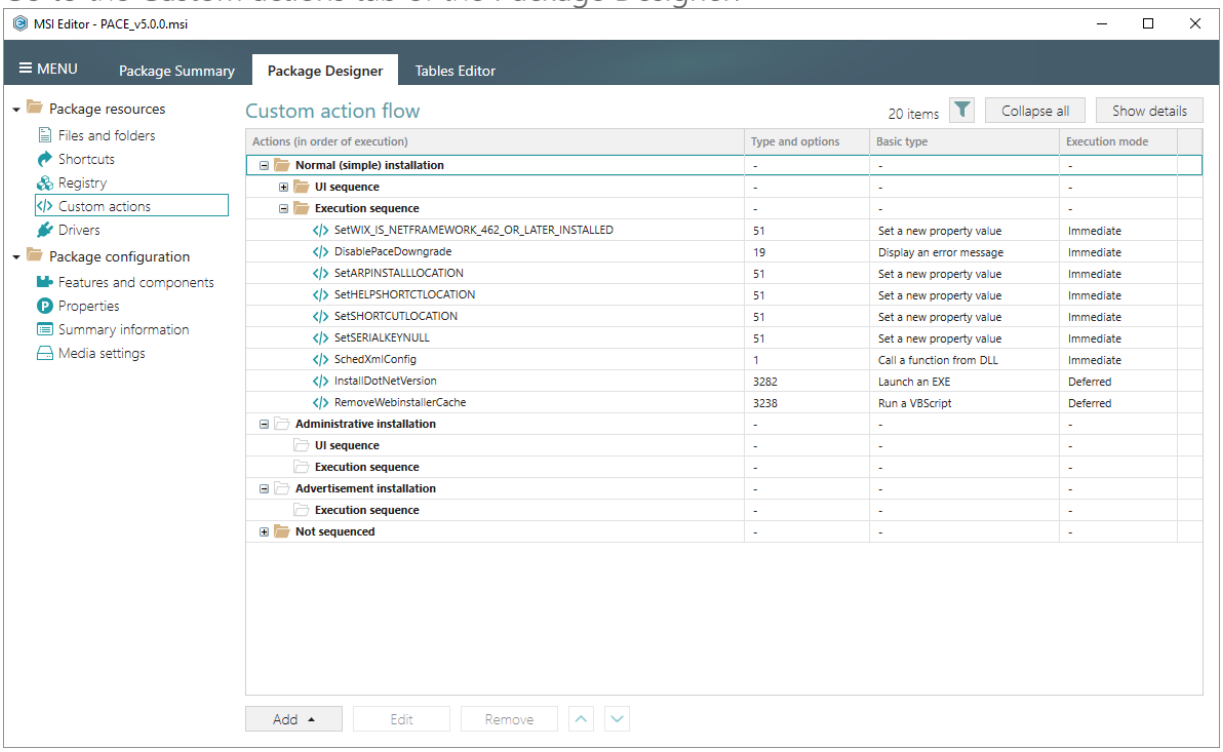

- For saving changes, select Save from MENU.



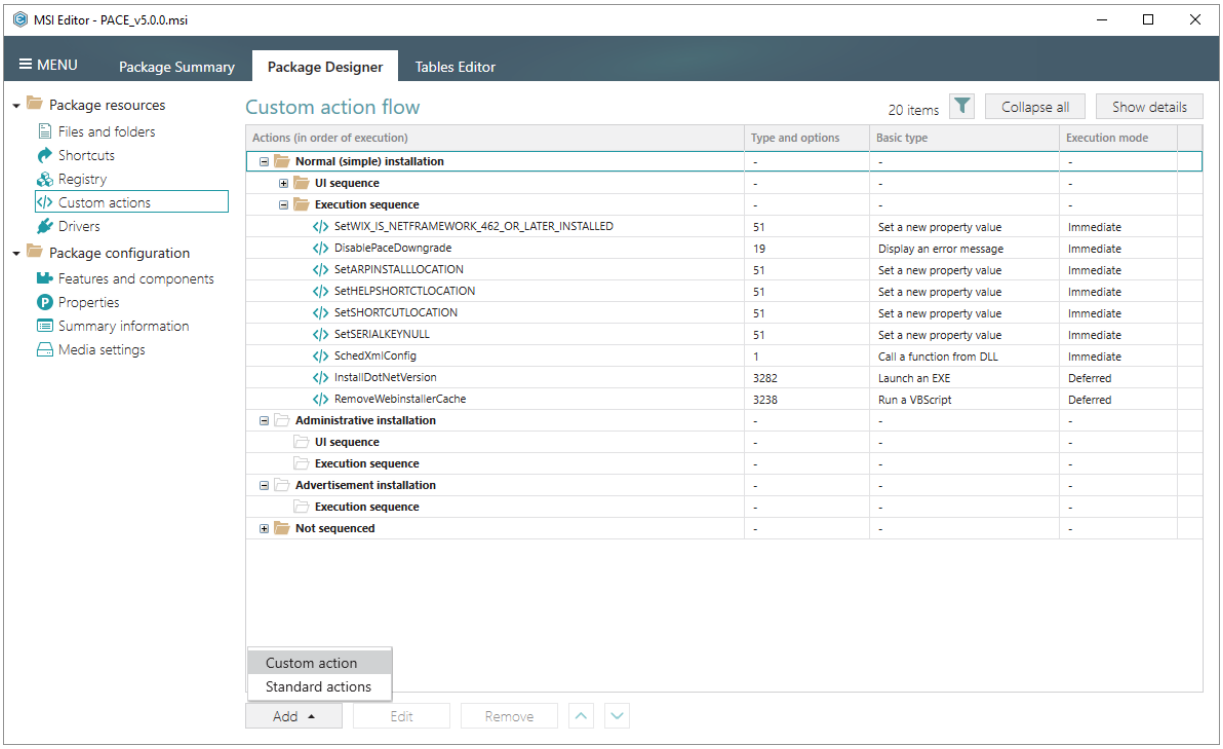
### 3.7.5.2 Add Custom Action

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.

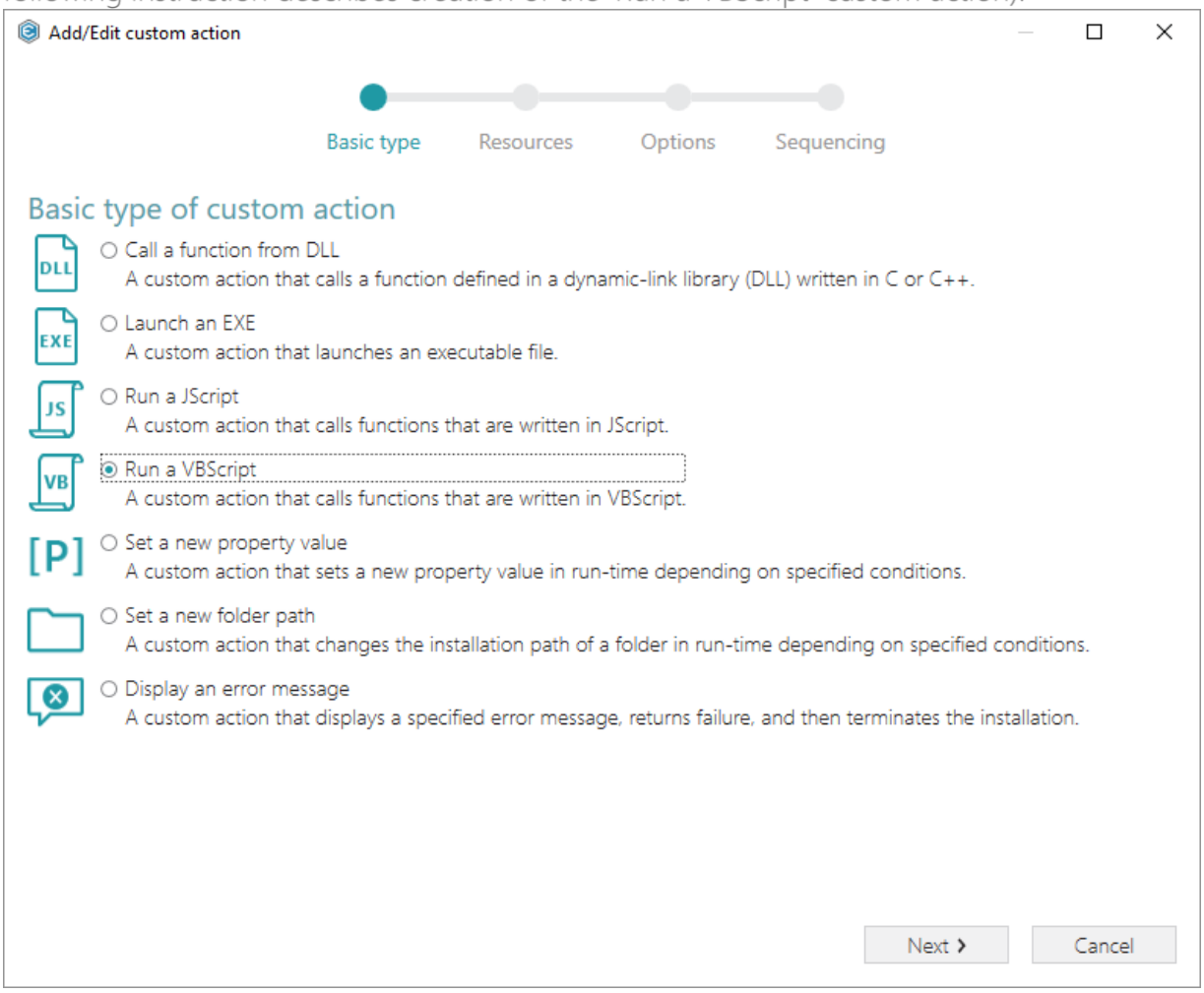
- Go to the Custom actions tab of the Package Designer.



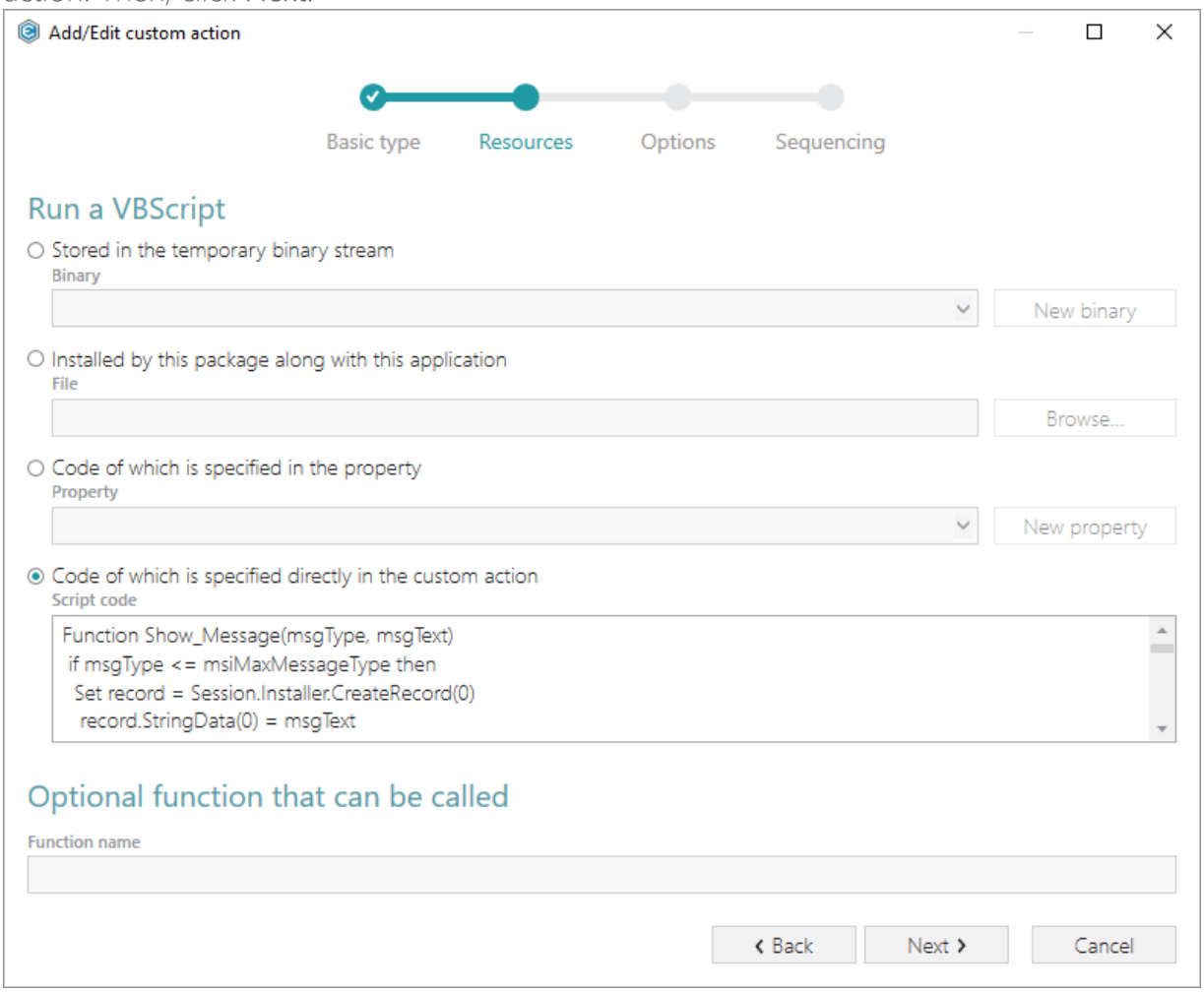

- Select Add -> Custom action from the toolbox.



- 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).

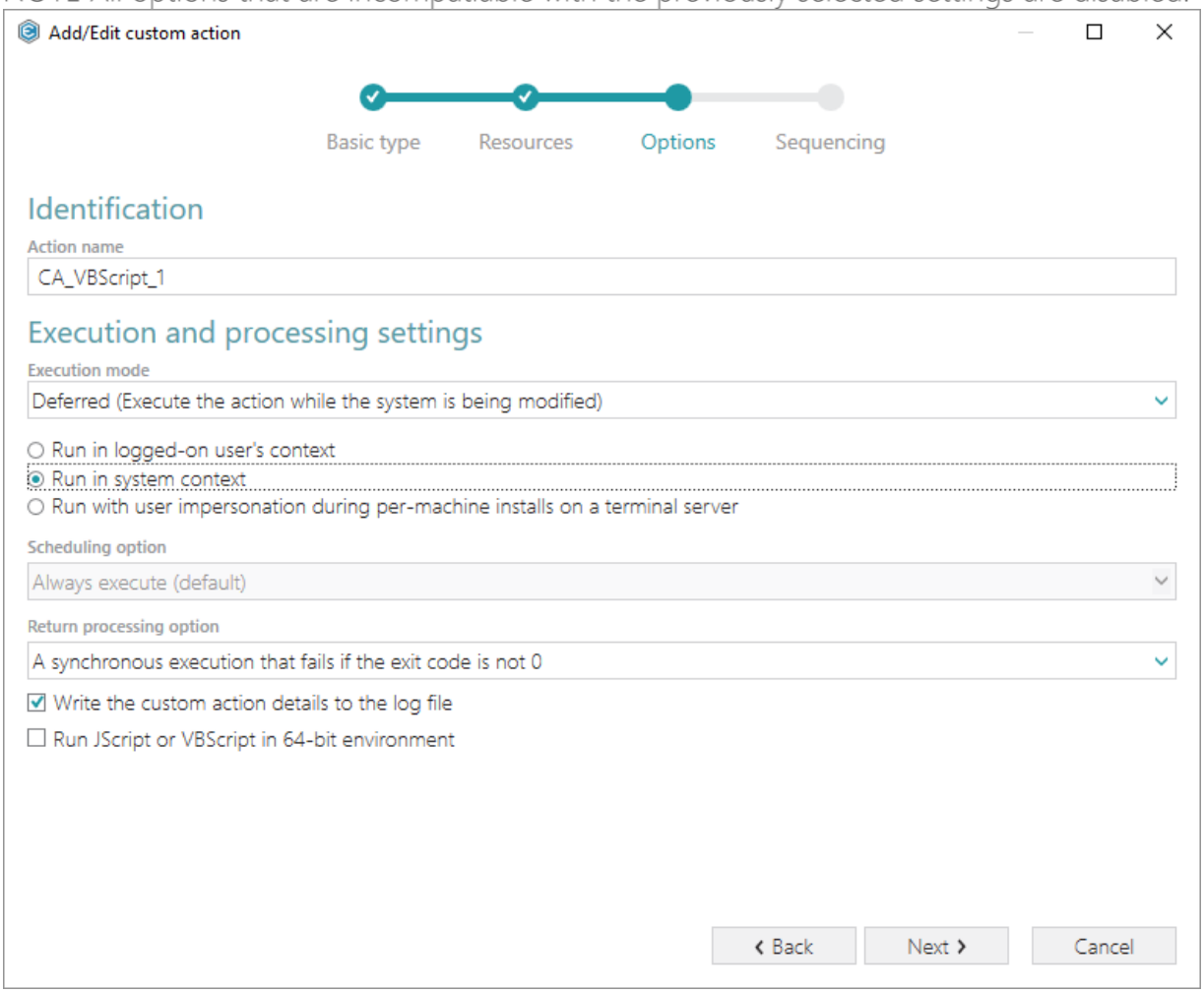


- Specify where the VBScript file is located or embed a script body directly into the custom action. Then, click Next.



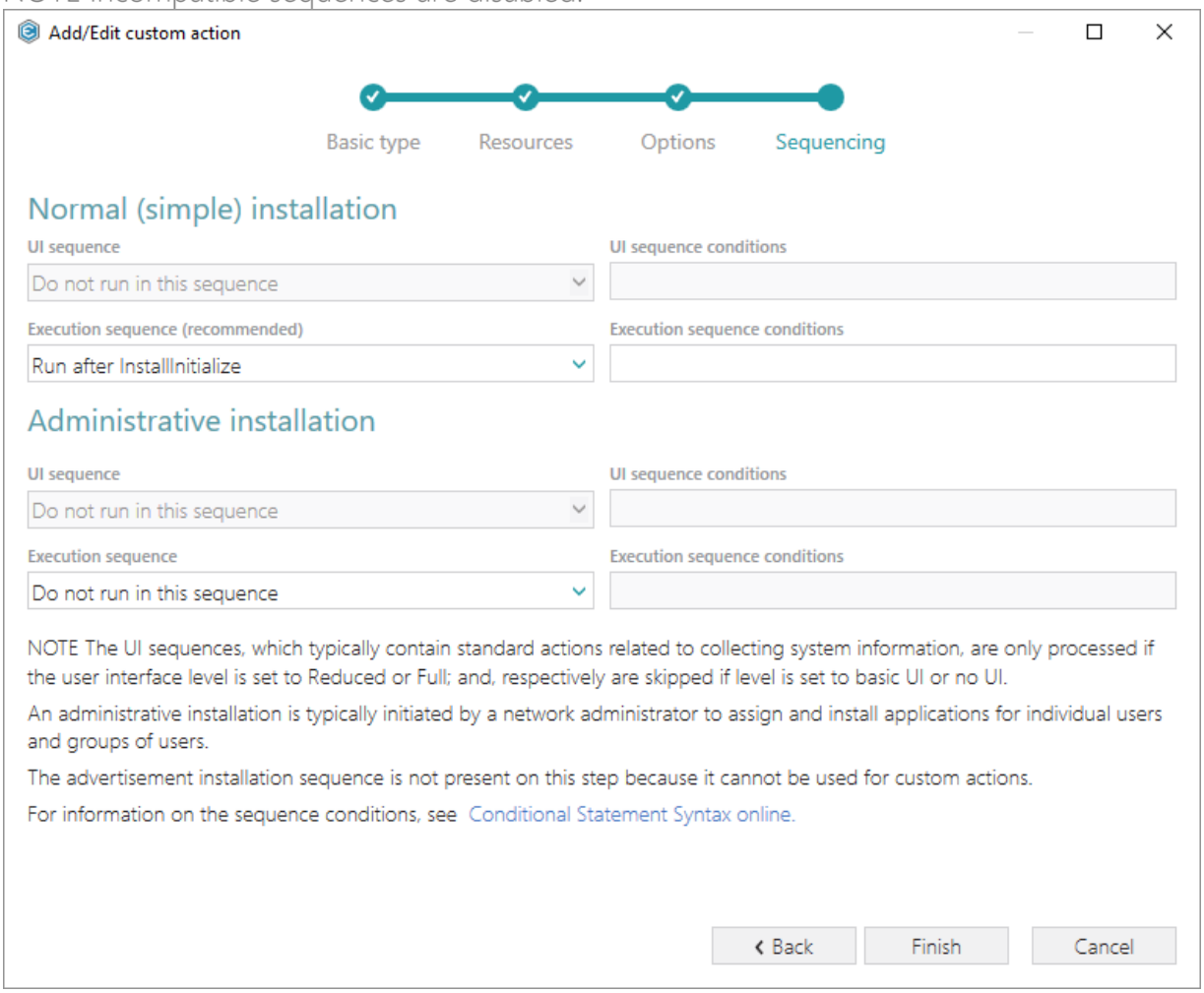

- At the Options step you can set an Action name, select Execution mode (Immediate, Deferred, Rollback and Commit), execution context, scheduling, return processing and other options. Then, click Next.

NOTE All options that are incompatiable with the previously selected settings are disabled.

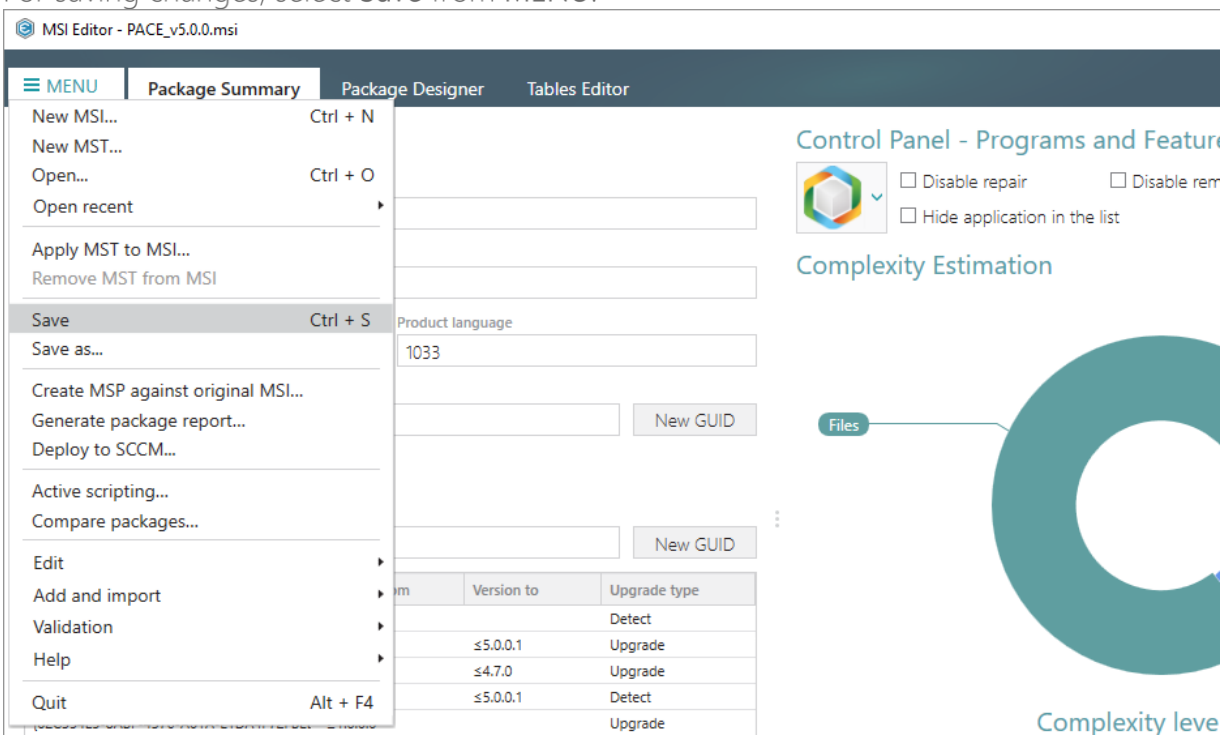



- The last step allows selecting secuences, where your custom action will be runned. The sequence drop-down list allows you to select only compatible positions, taking into account your previously selected options. Also, you are free to specify execution conditions either by entering them manually or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

NOTE Incompatible sequences are disabled.



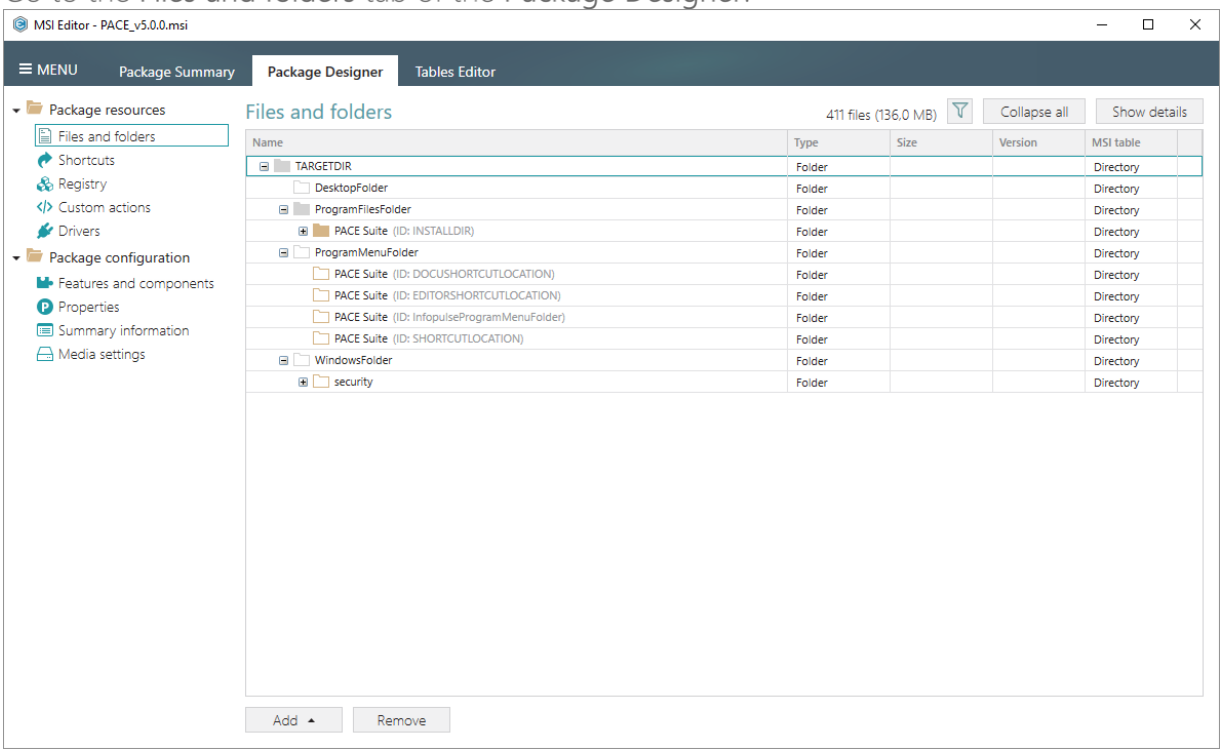

- For saving changes, select Save from MENU.



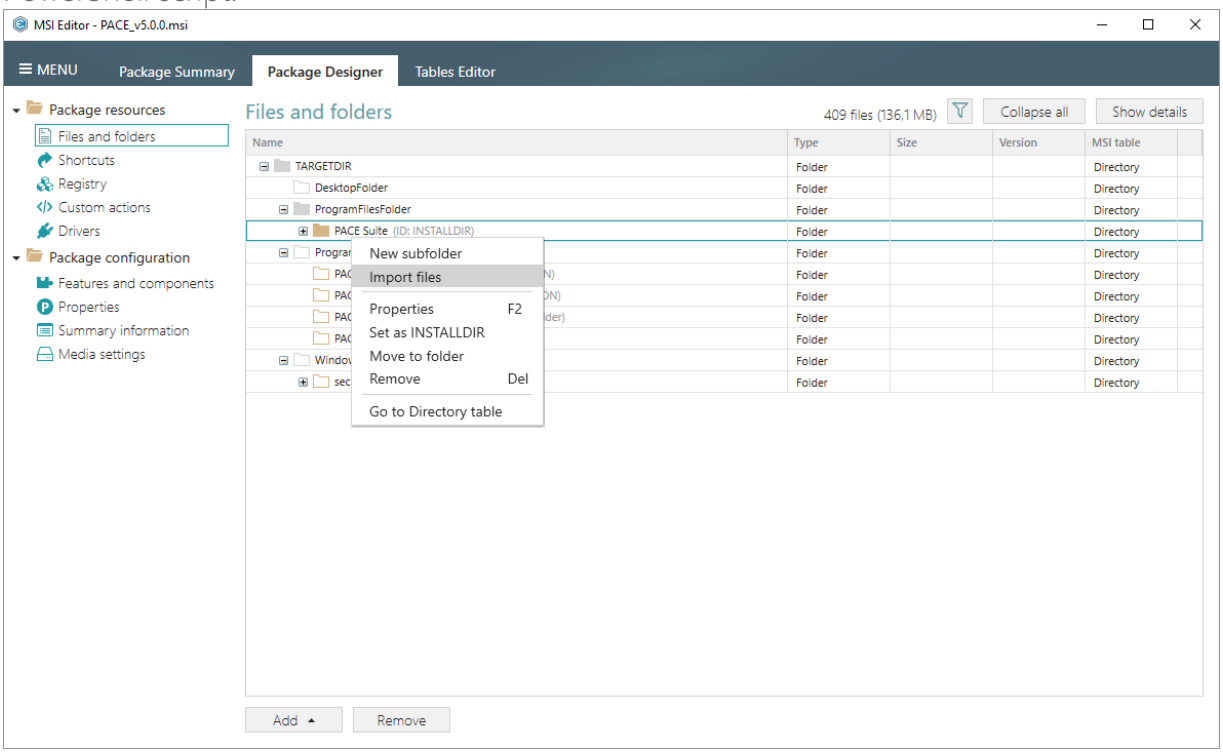
### 3.7.5.2.1 Run PowerShell script

Run a PowerShell script from MSI during the installation or de-installation process. This solution consists of two parts: (1) importing PowerShell script to the package; (2) adding a Custom action for running this script.

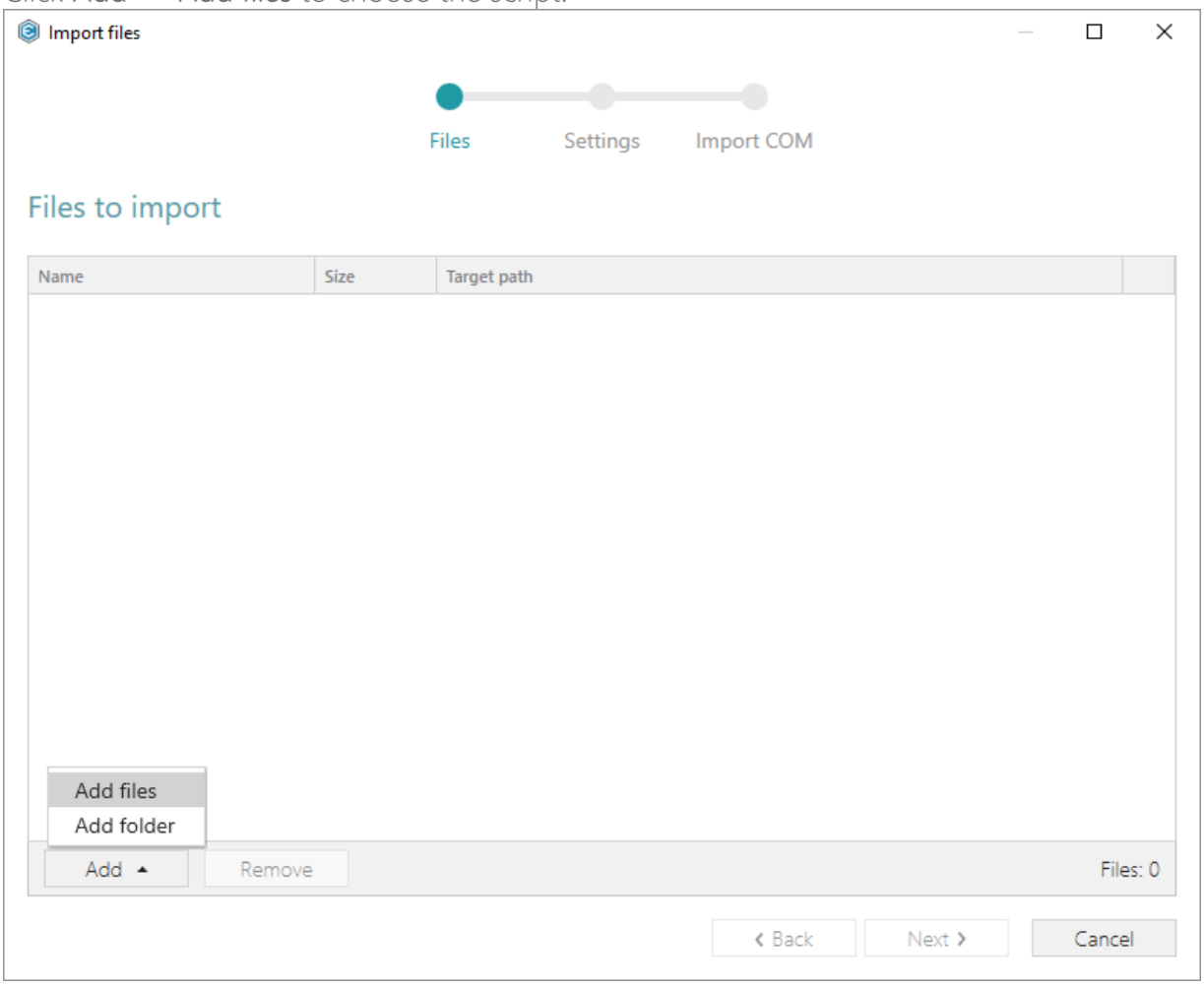
- Go to the **Files and folders** tab of the **Package Designer**.



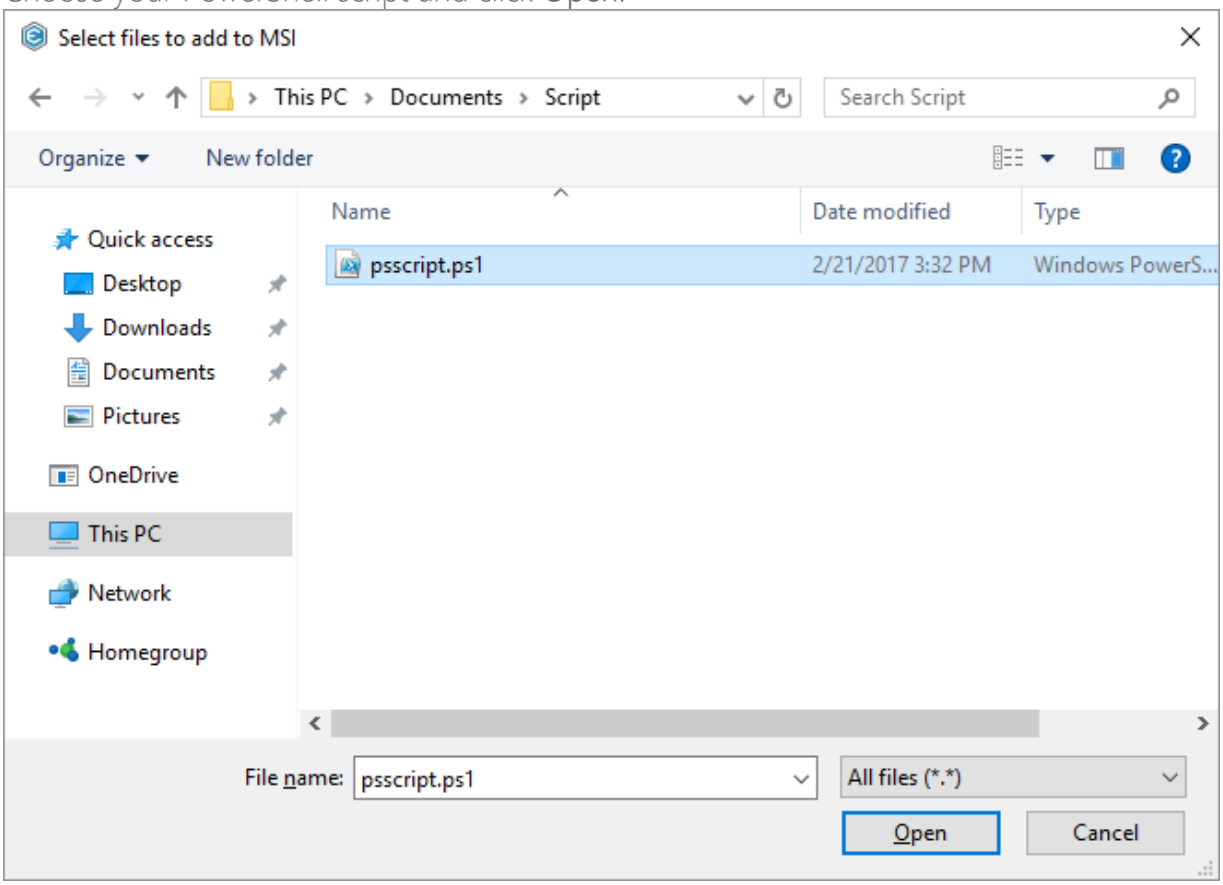

- Select **Import files** from the context menu of a folder, to which you want to import the PowerShell script.



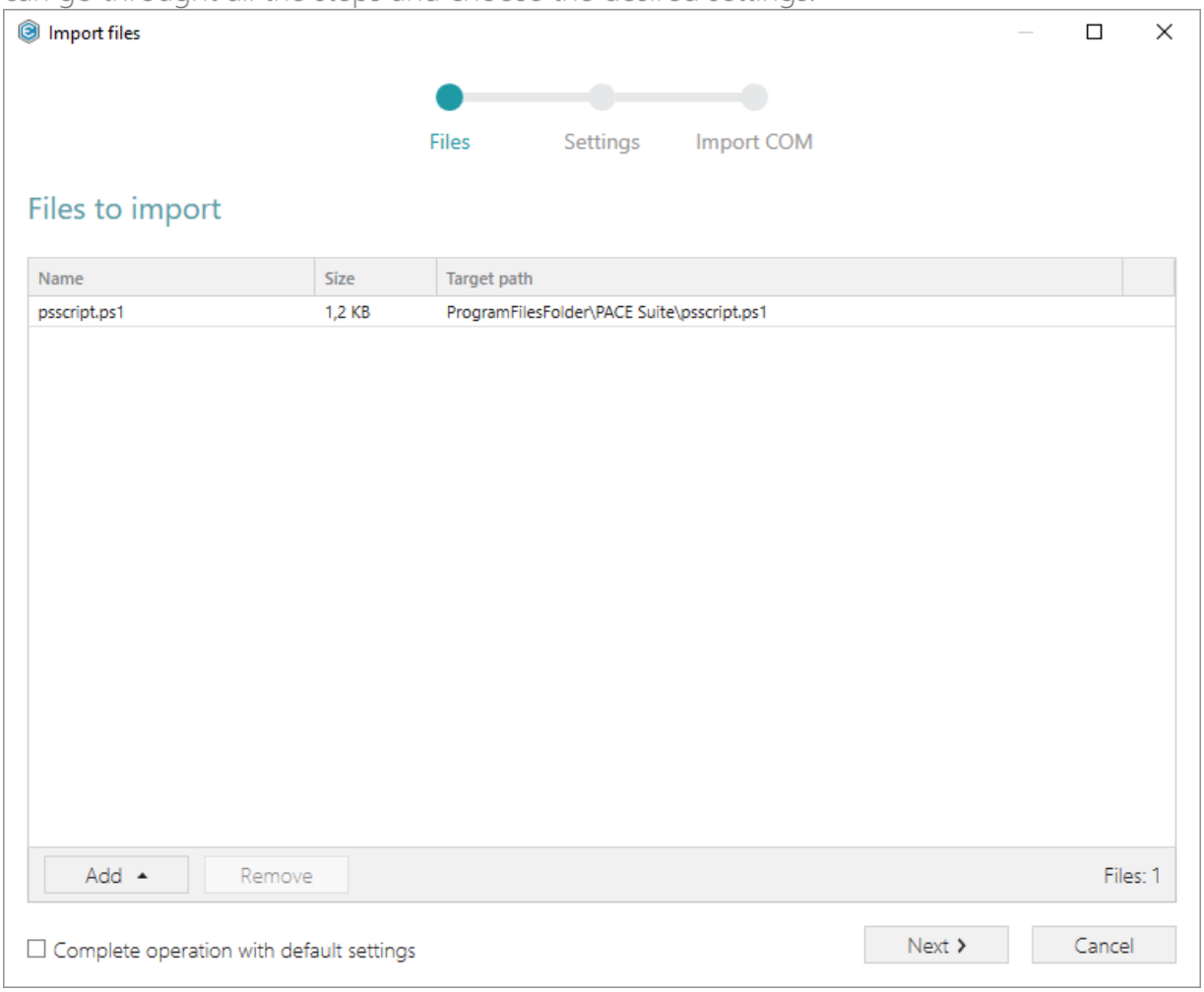
- Click Add -> Add files to choose the script.



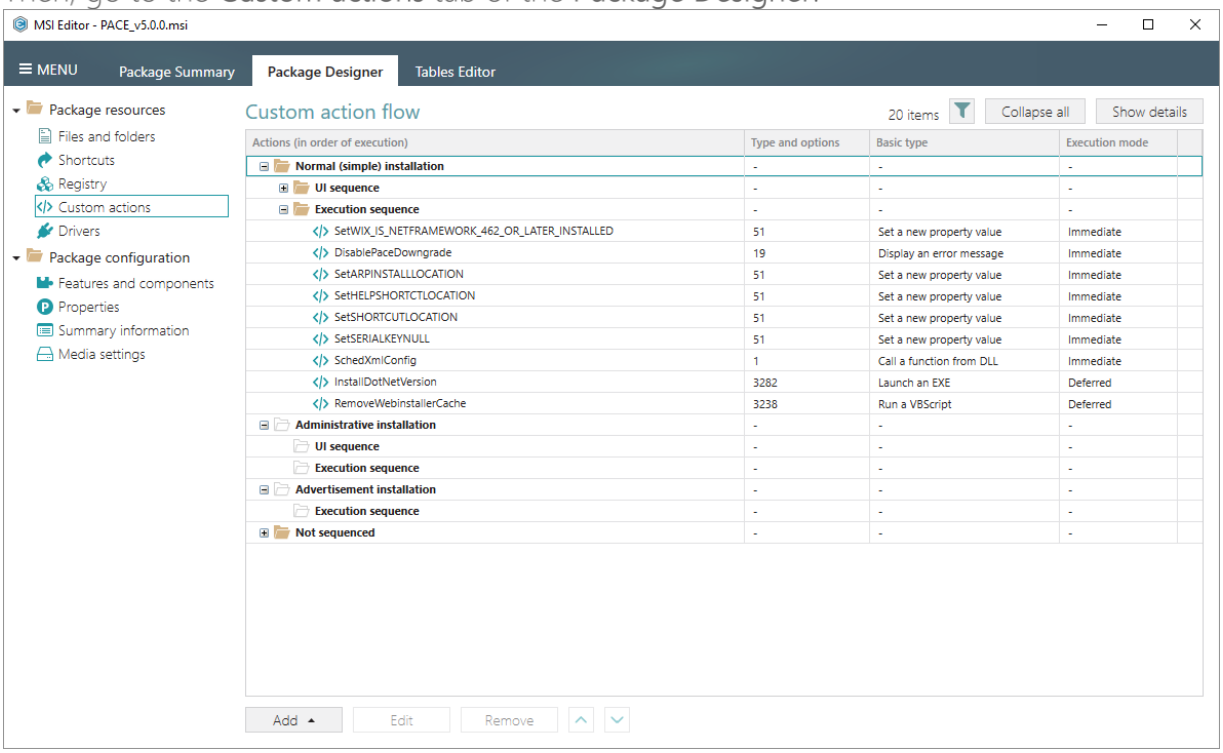
- Choose your PowerShell script and click Open.



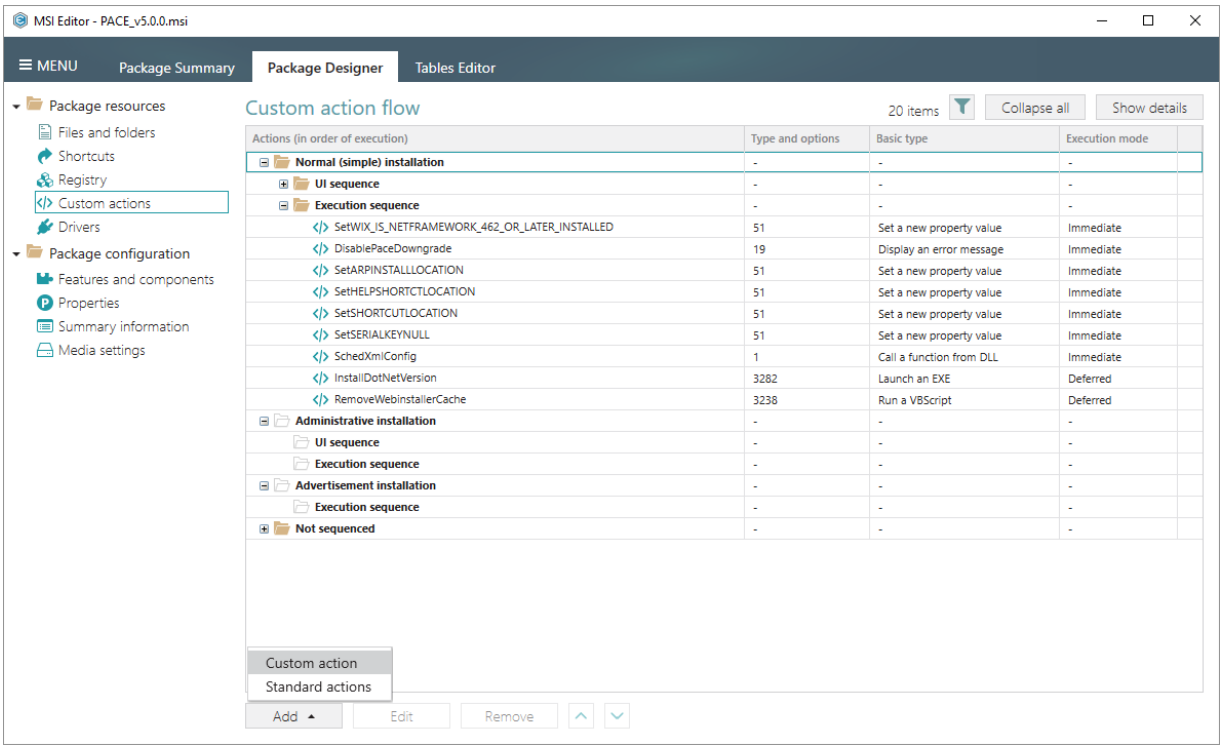
- Tick the **Complete operation with default settings** option and click **Finish**. Alternatively, you can go throught all the steps and choose the desired settings.



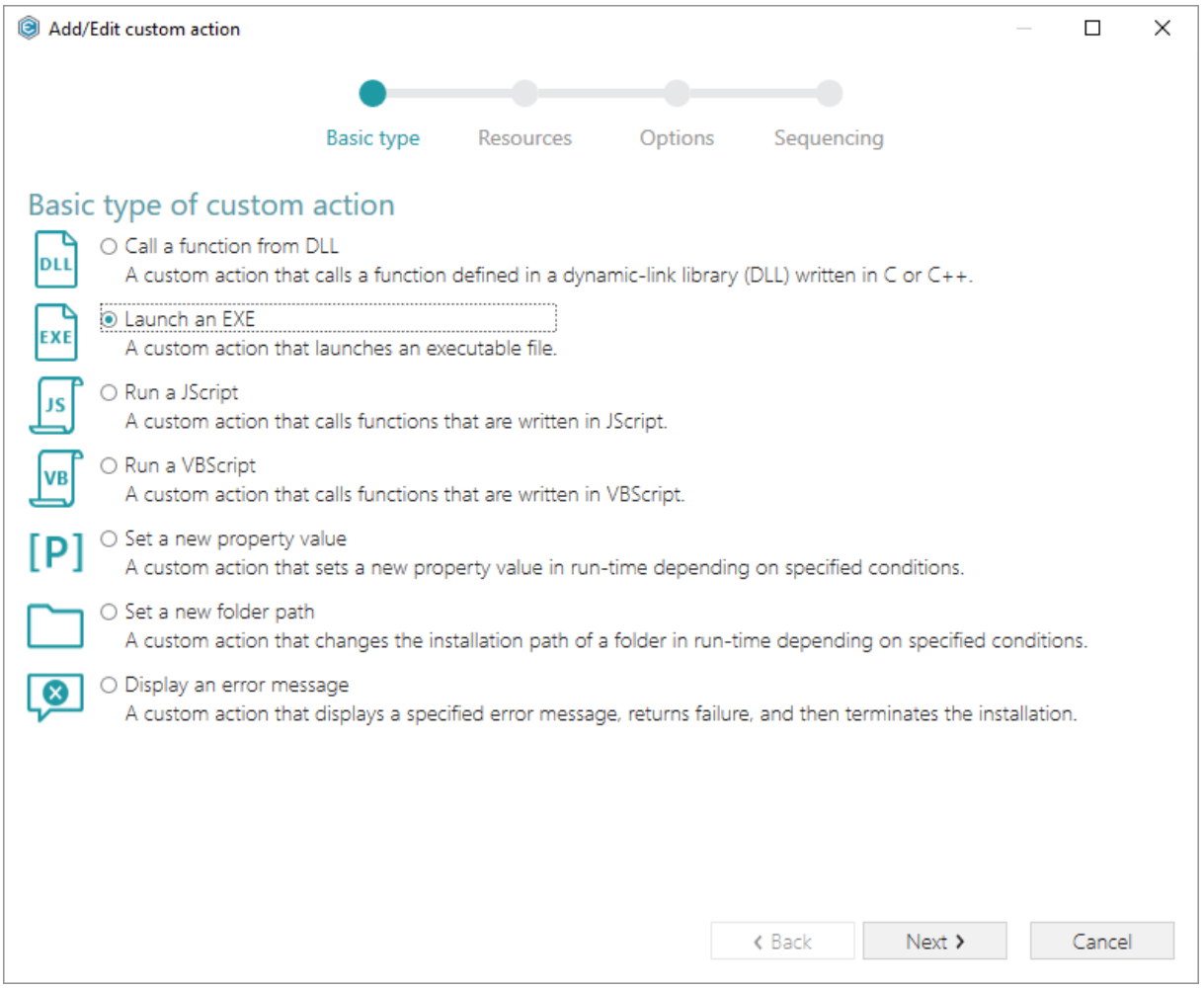

- Then, go to the **Custom actions** tab of the **Package Designer**.



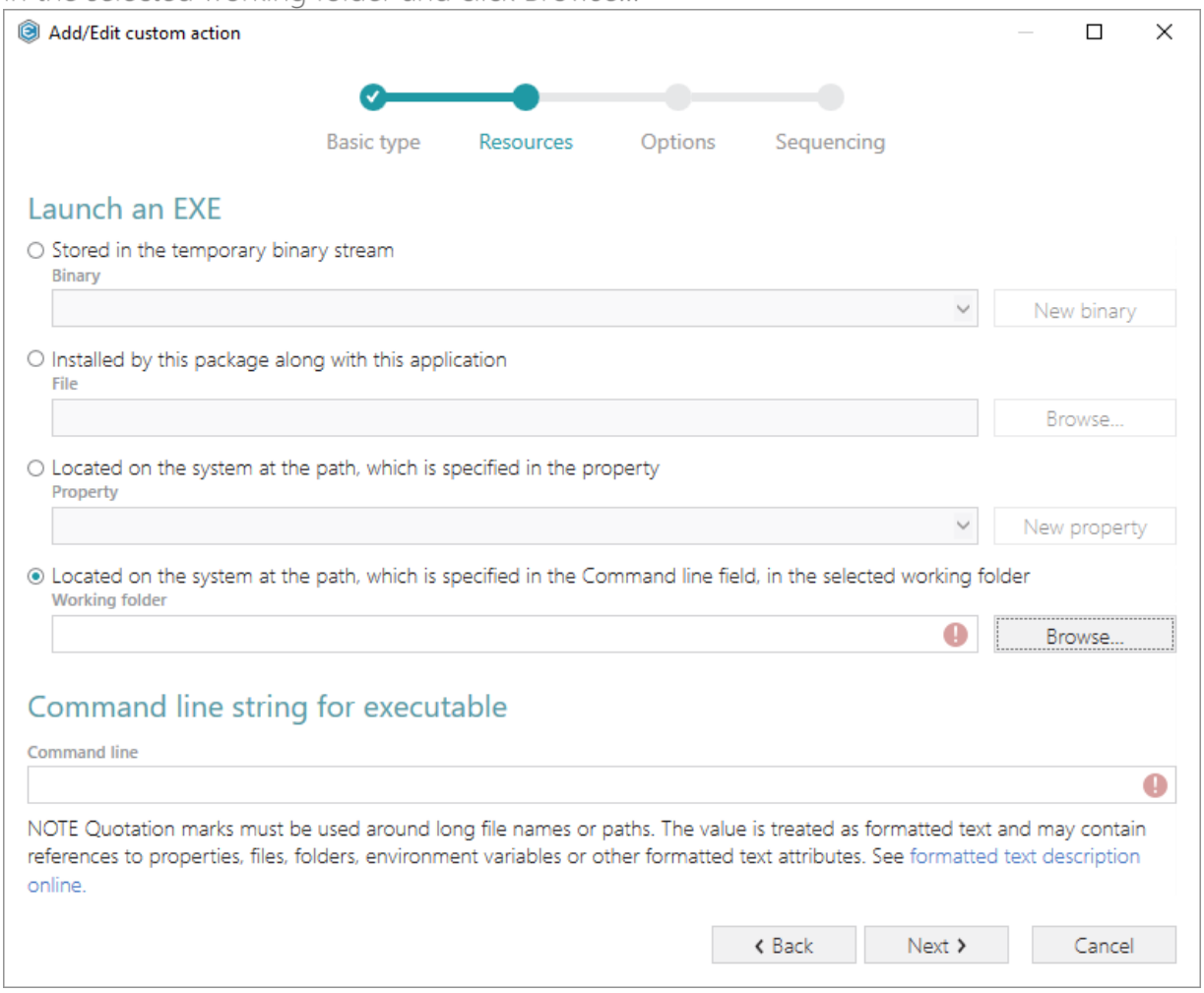
- Select Add -> Custom action from the toolbox.



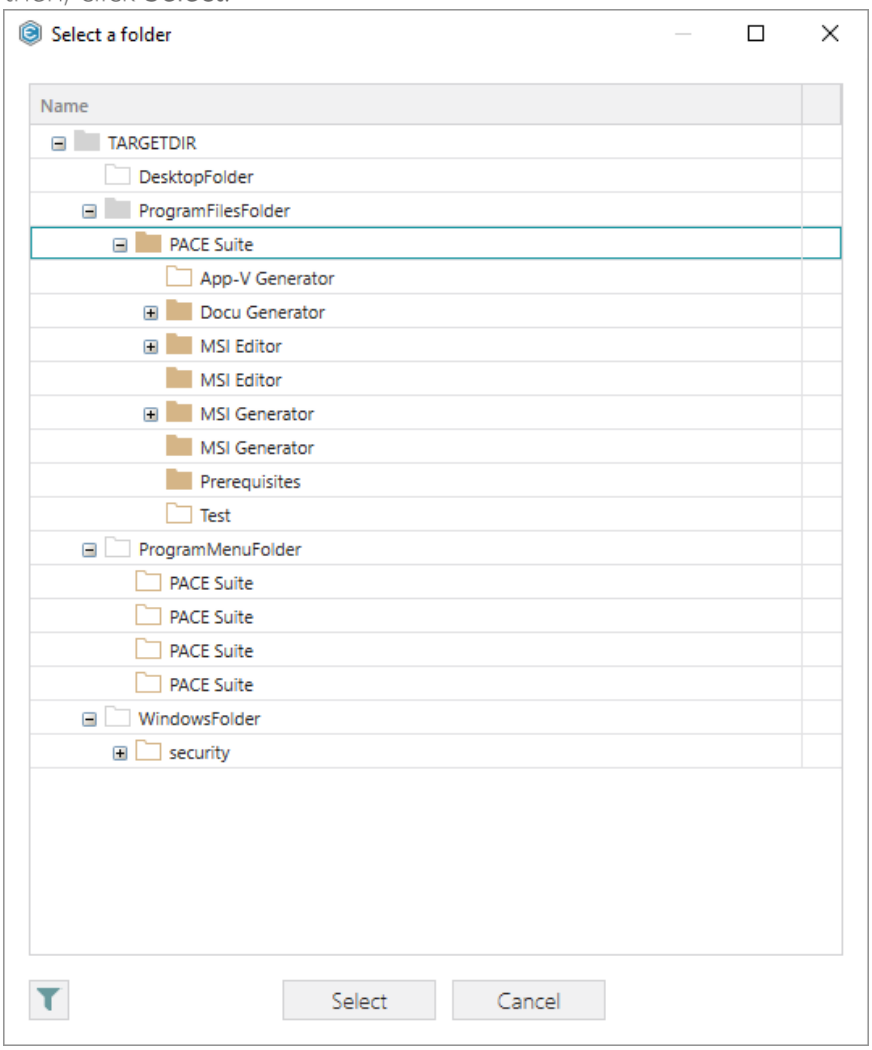

- Select Launch an EXE and click Next.



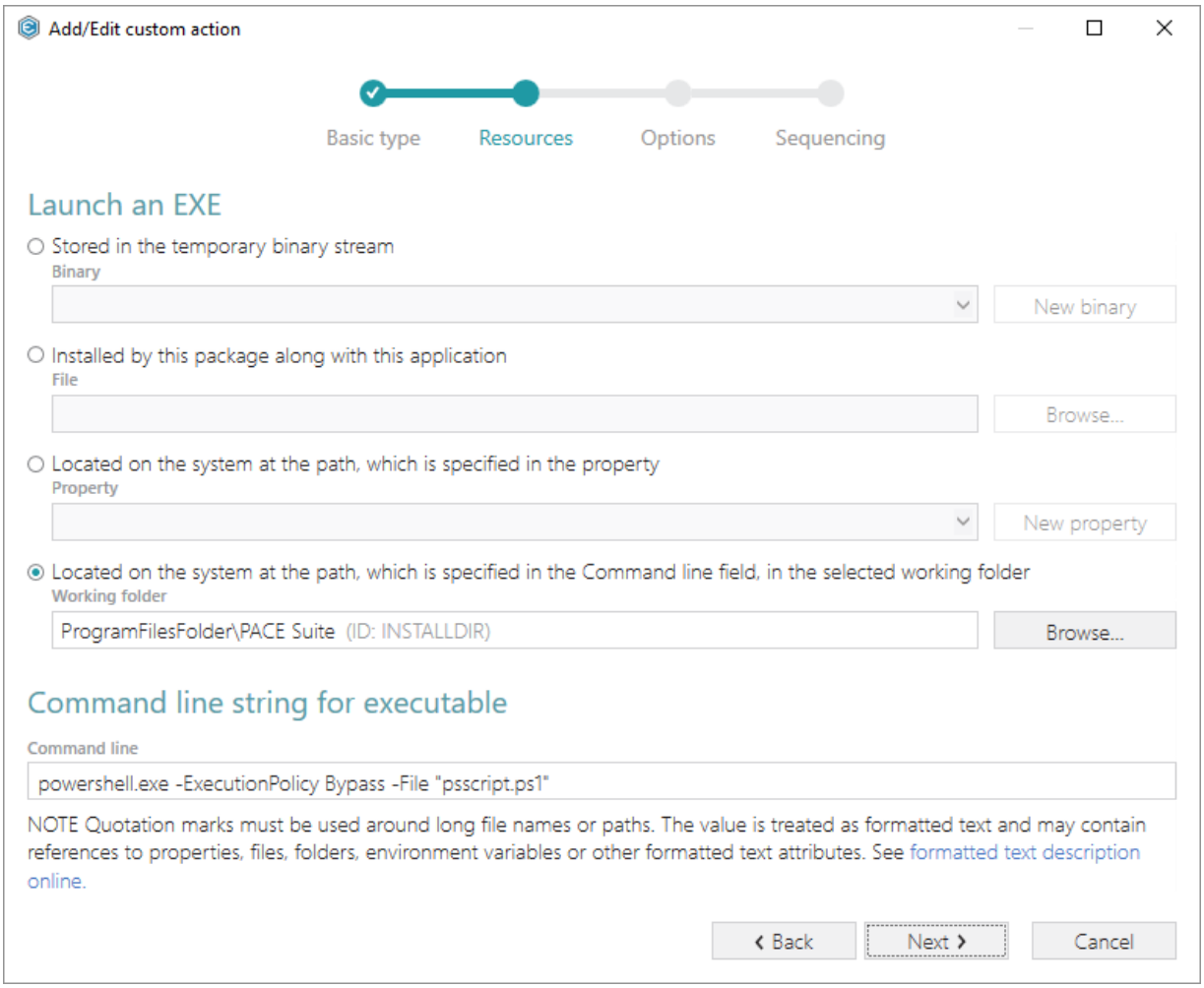
- Select the Located on the system at the path, which is specified in the Command line field, in the selected working folder and click Browse...



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



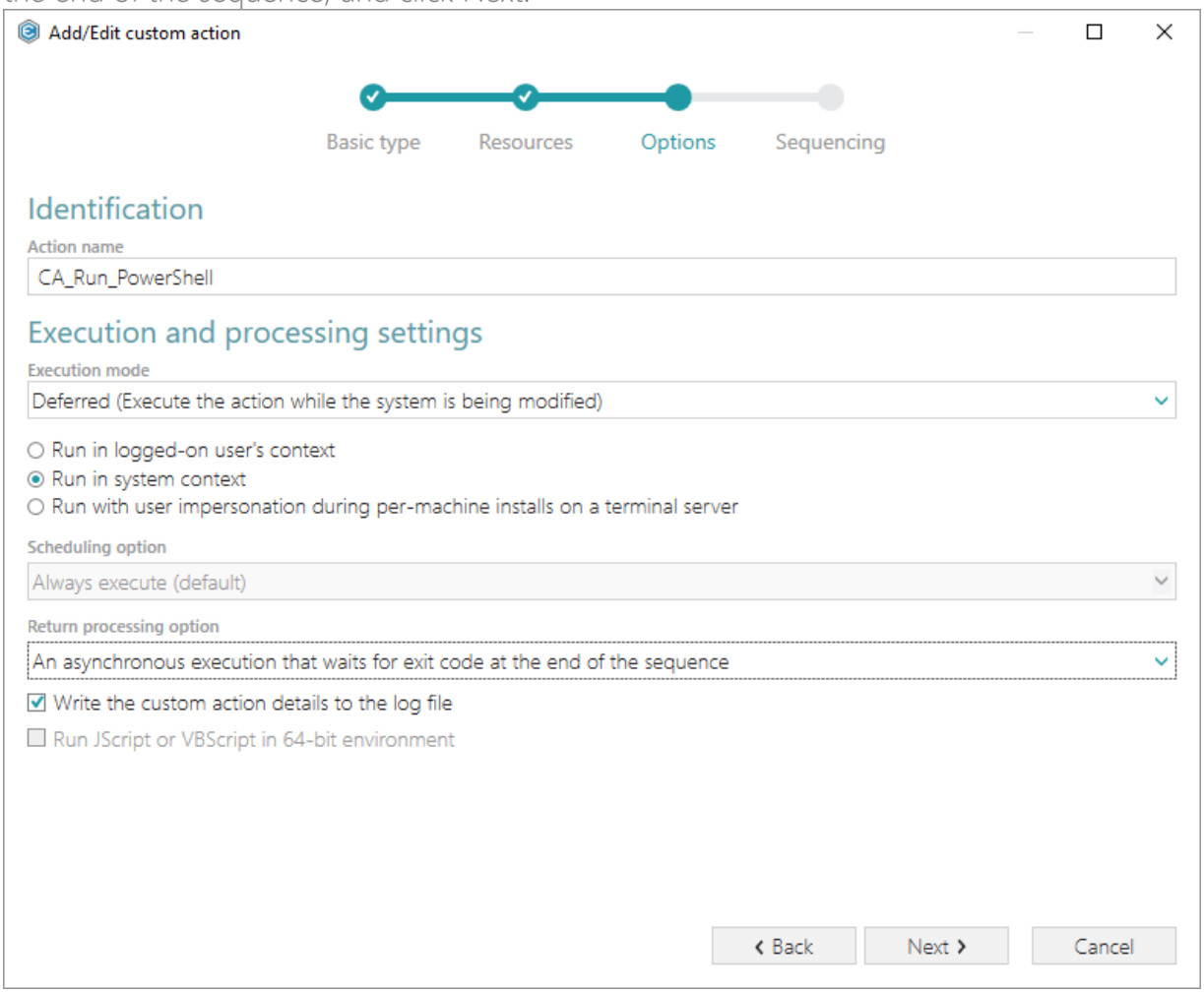

- Enter the *powershell.exe -ExecutionPolicy Bypass -File "<script file name>.ps1"* to the Command line field, where *<script file name>* is replaced with a name of your PowerShell script (e.g. psscript.ps1). Then, click Next.




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

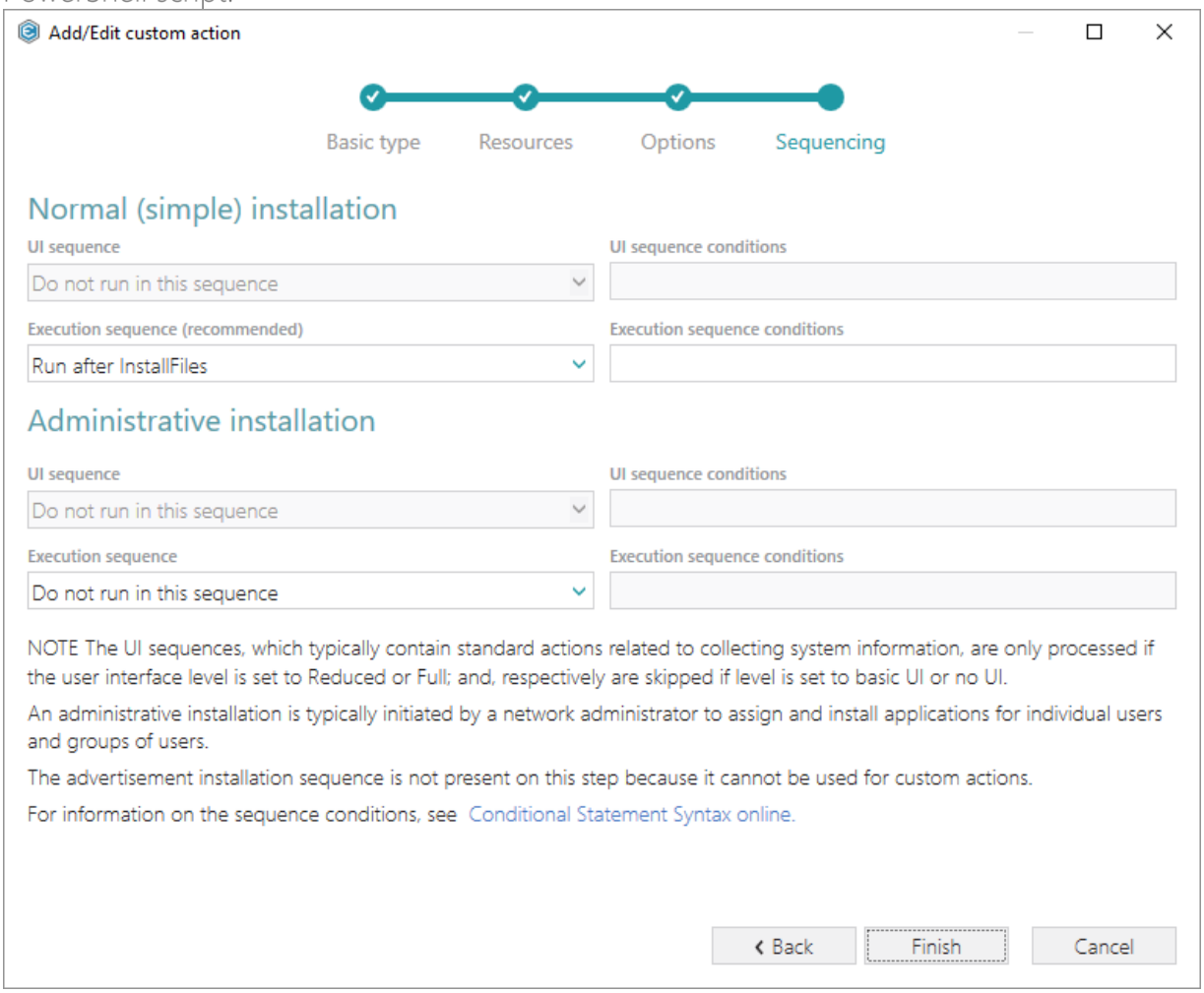
- Select the Deferred execution mode, and the Run in system context option. From the Return processing options list, choose An asynchronous execution that waits for exit code at

the end of the sequence, and click Next.

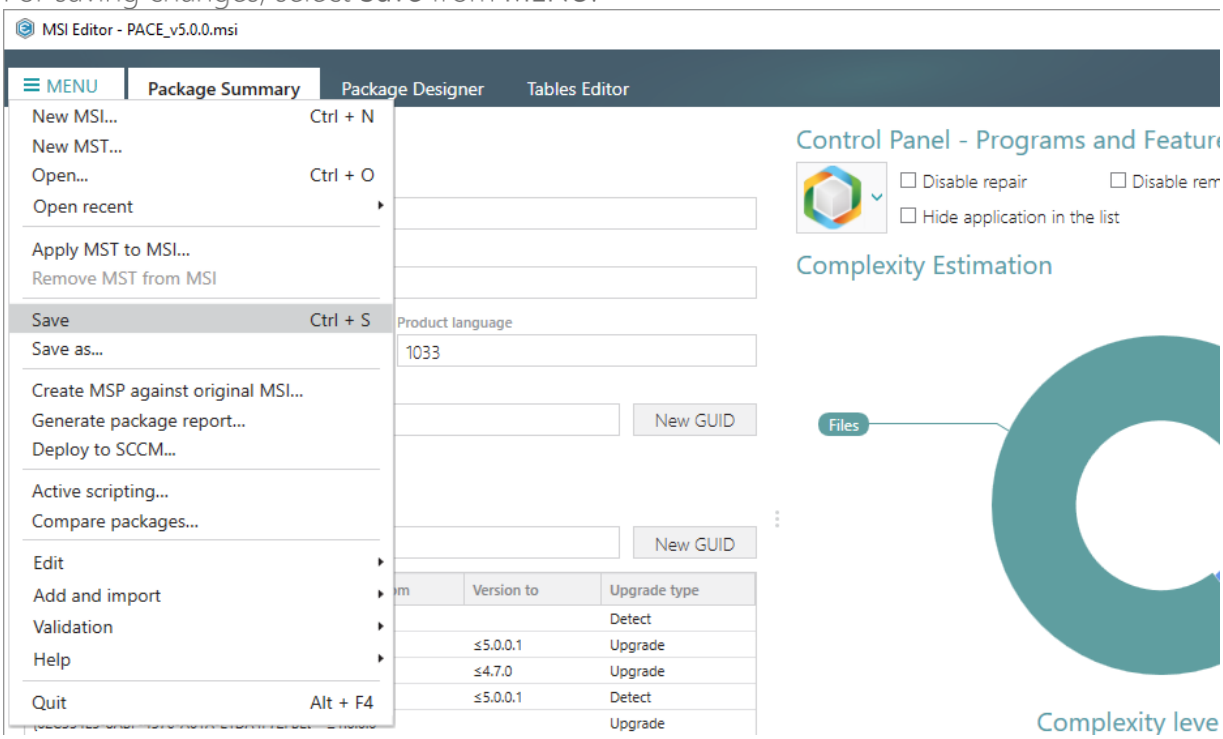



- From the **Execution sequence (recommended)** list, select **Run after InstallFiles** (or any other position, located below). If needed, specify the execution conditions - enter them manually or choose from the dropped list. Click **Finish** to add your custom action for running

PowerShell script.



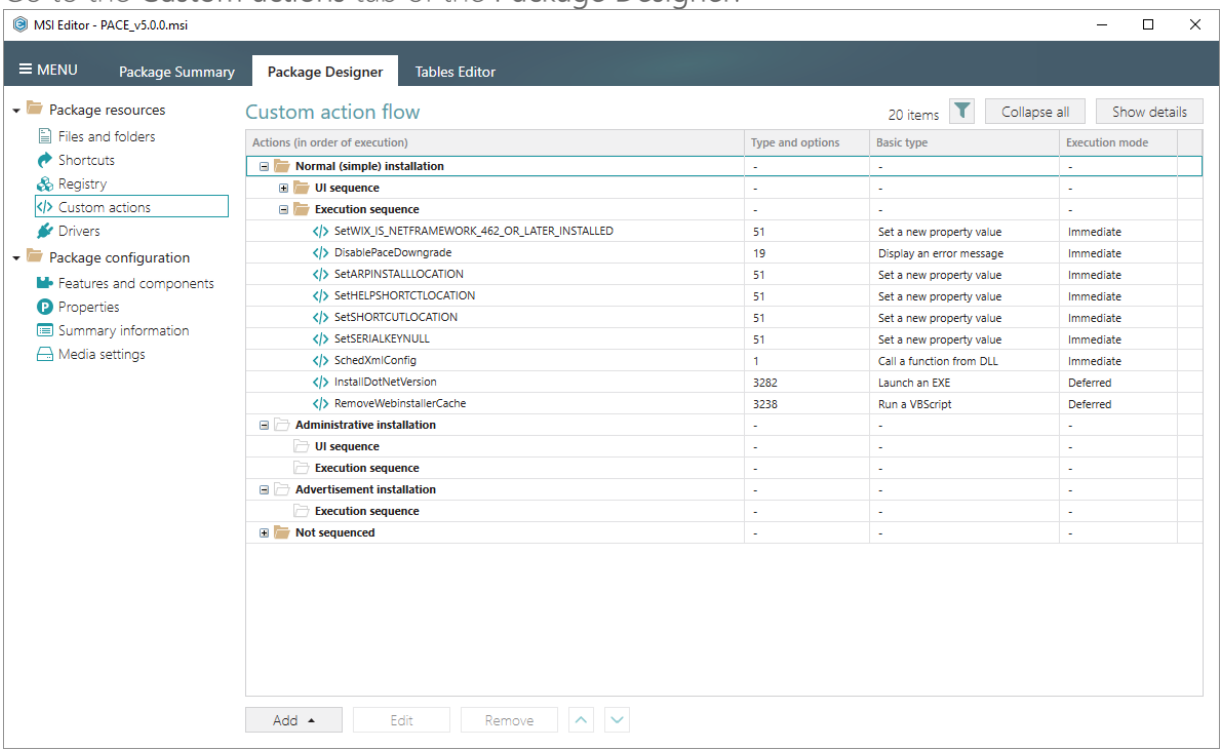

- For saving changes, select Save from MENU.



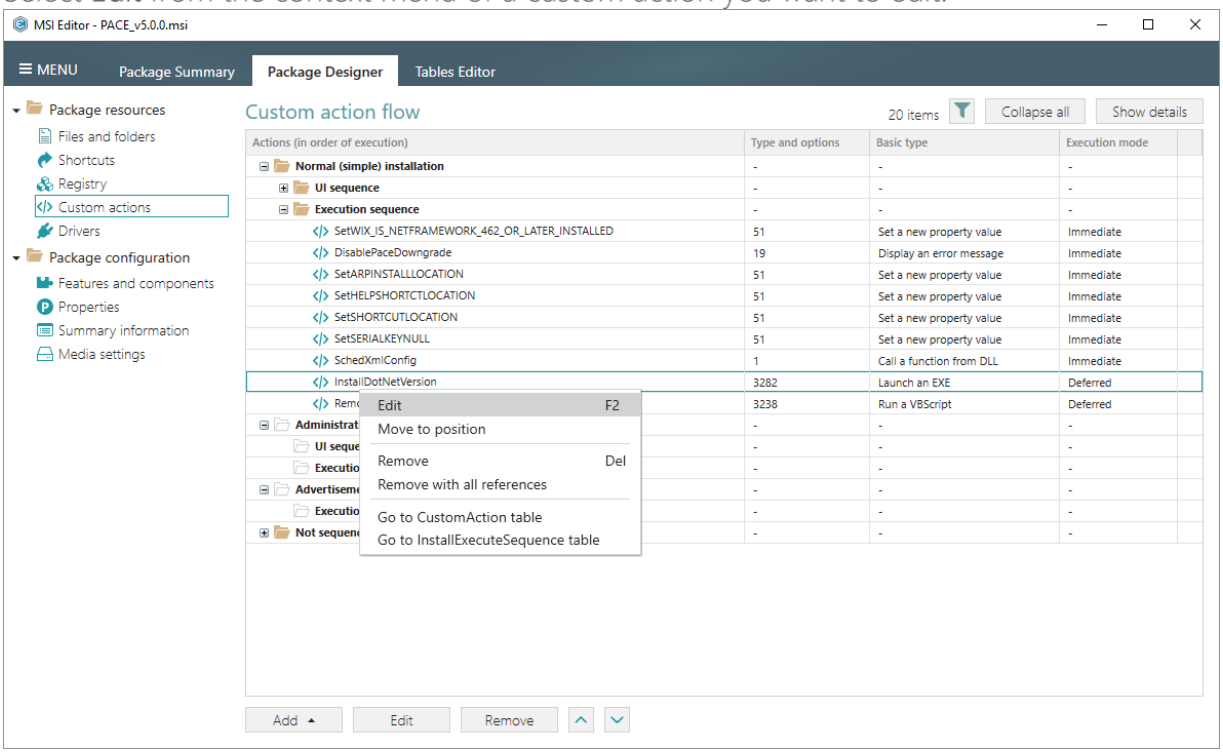
### 3.7.5.3 Edit Custom Action

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

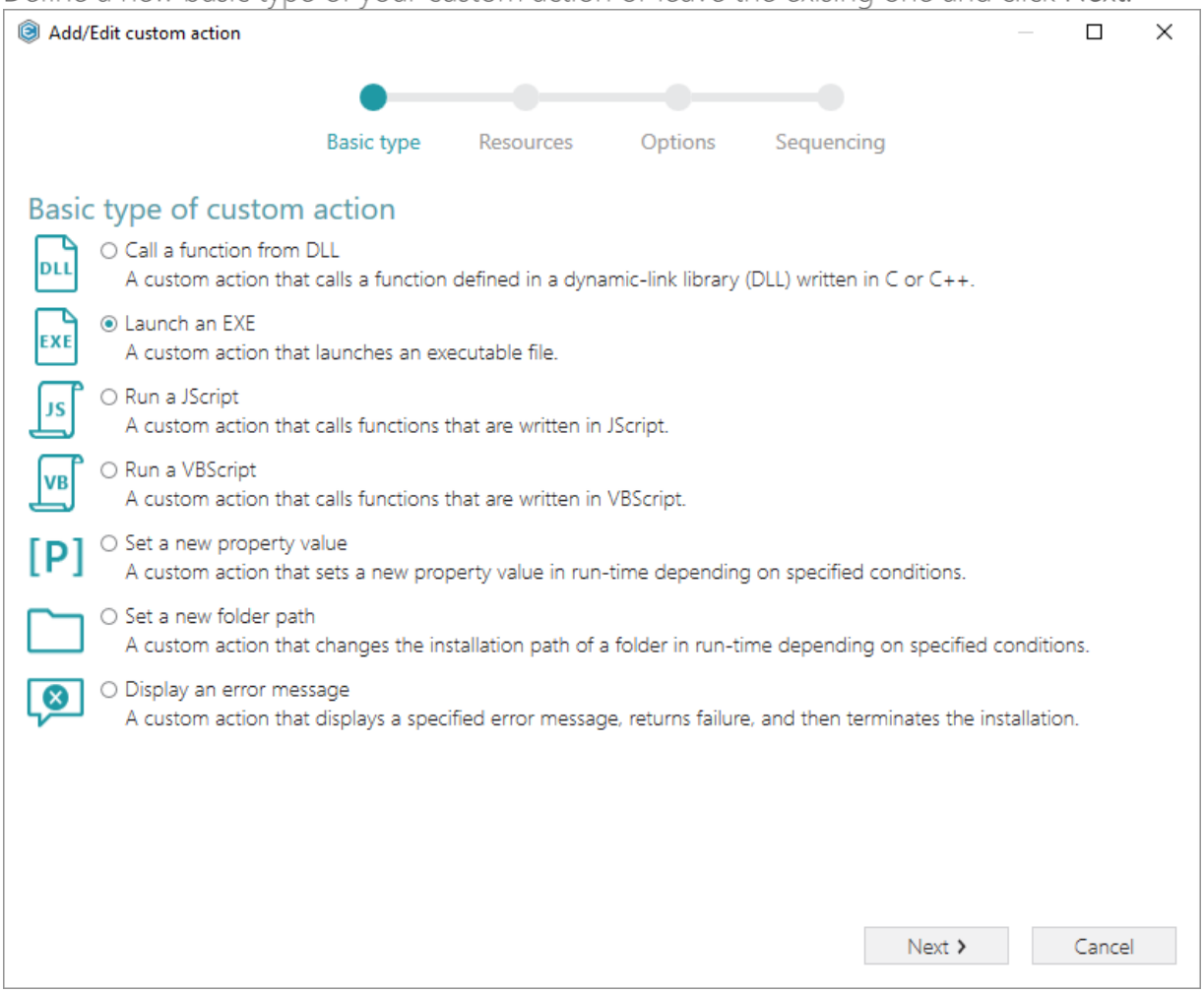
- Go to the Custom actions tab of the Package Designer.




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

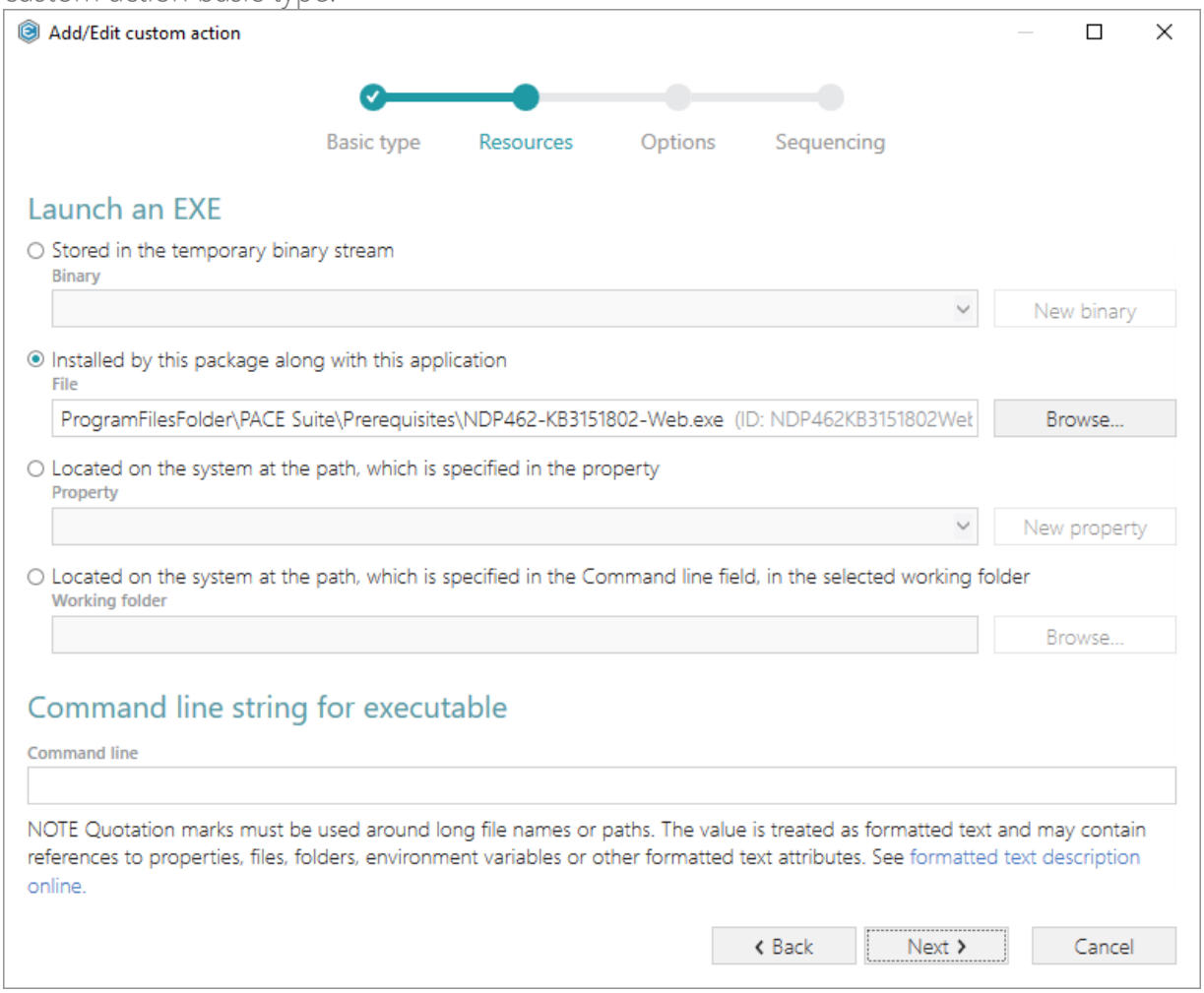


- Define a new basic type of your custom action or leave the exising one and click Next.



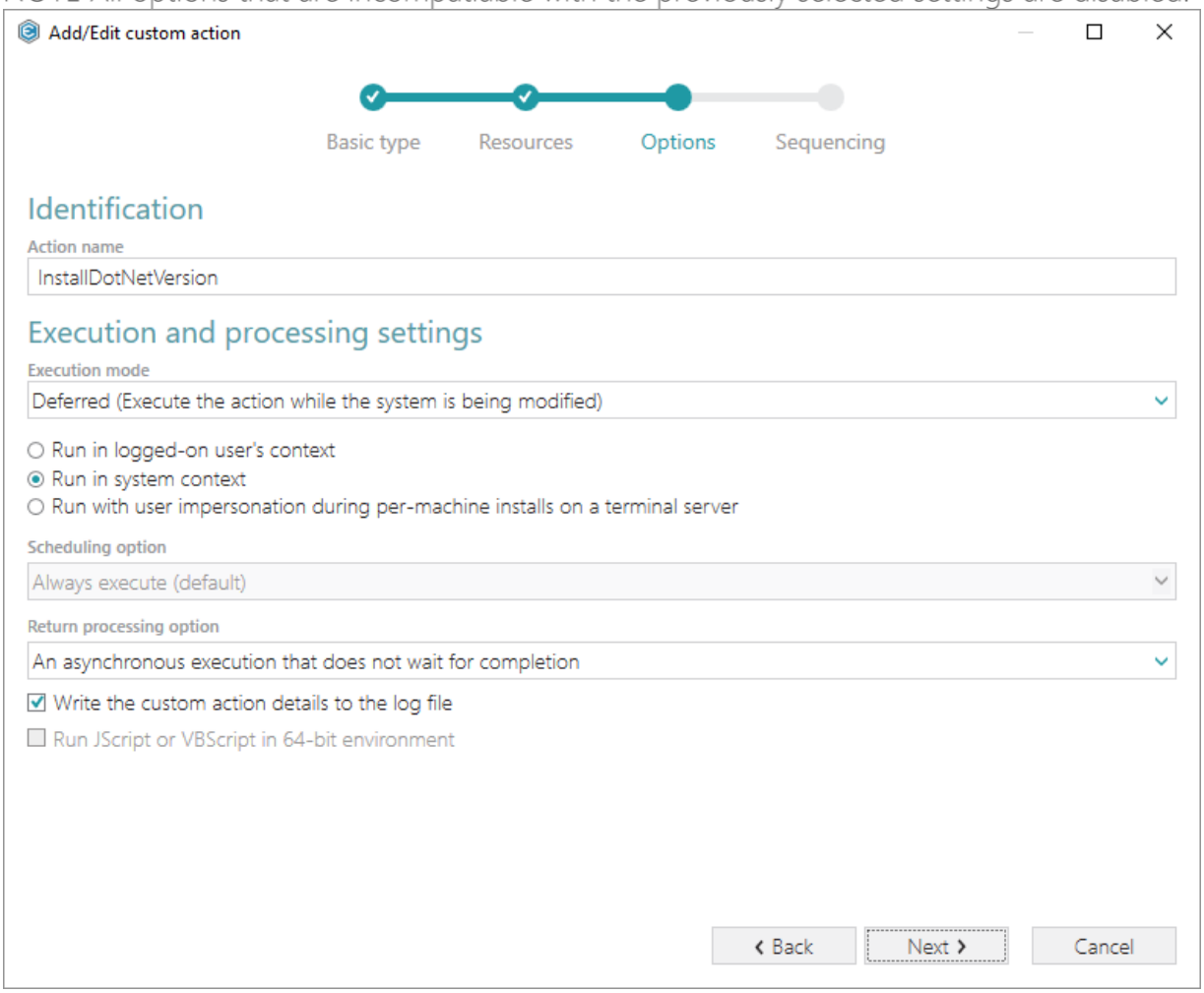

- On the second step you can specify a new file, launched by the custom action. Click Next to continue.
  NOTE The wizard displays only compatible fields and options for the previously selected

custom action basic type.



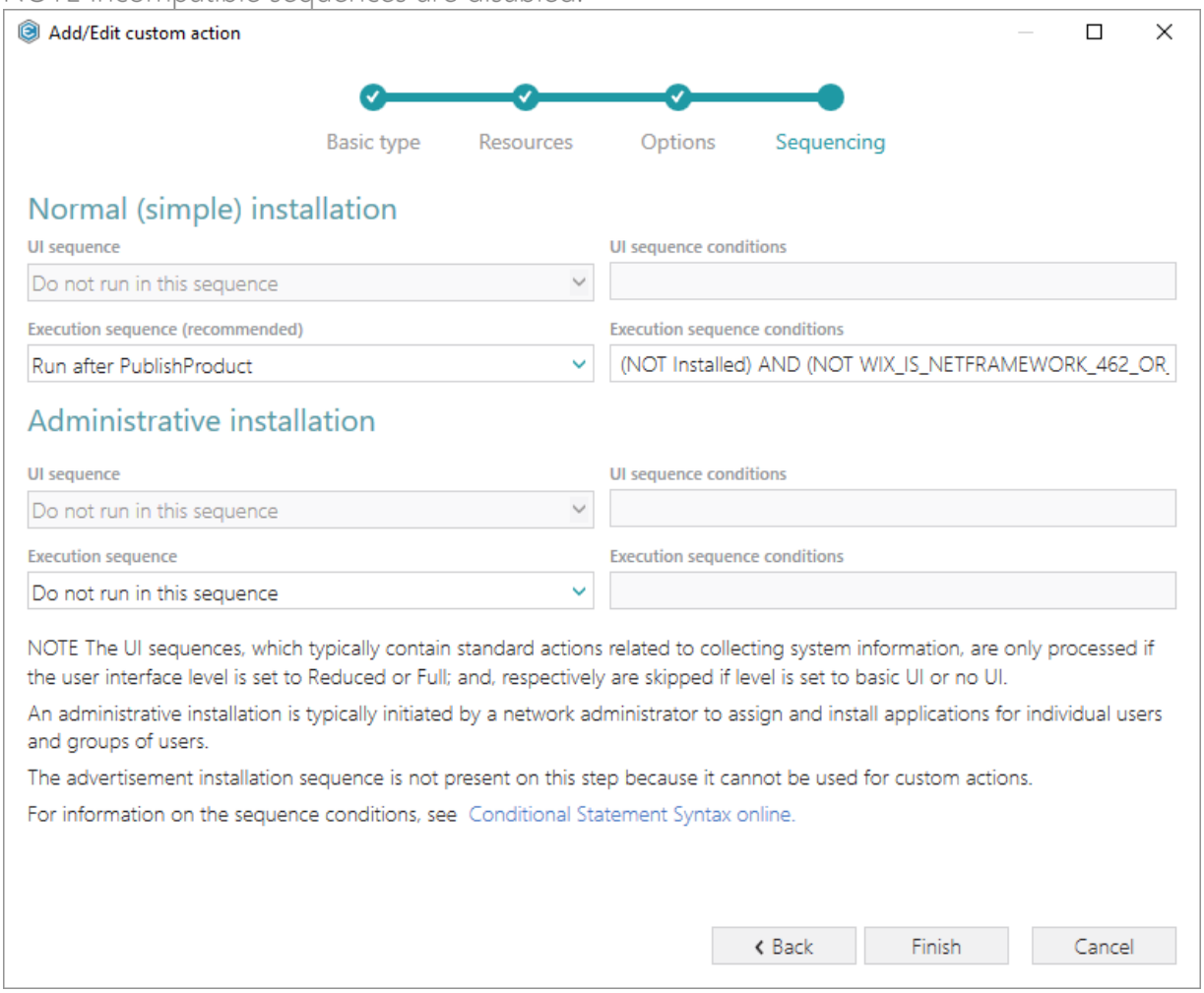

- At the Options step you can set an Action name, select Execution mode (Immediate, Deferred, Rollback and Commit), execution context, scheduling, return processing and other options. Then, click Next.

NOTE All options that are incompatiable with the previously selected settings are disabled.

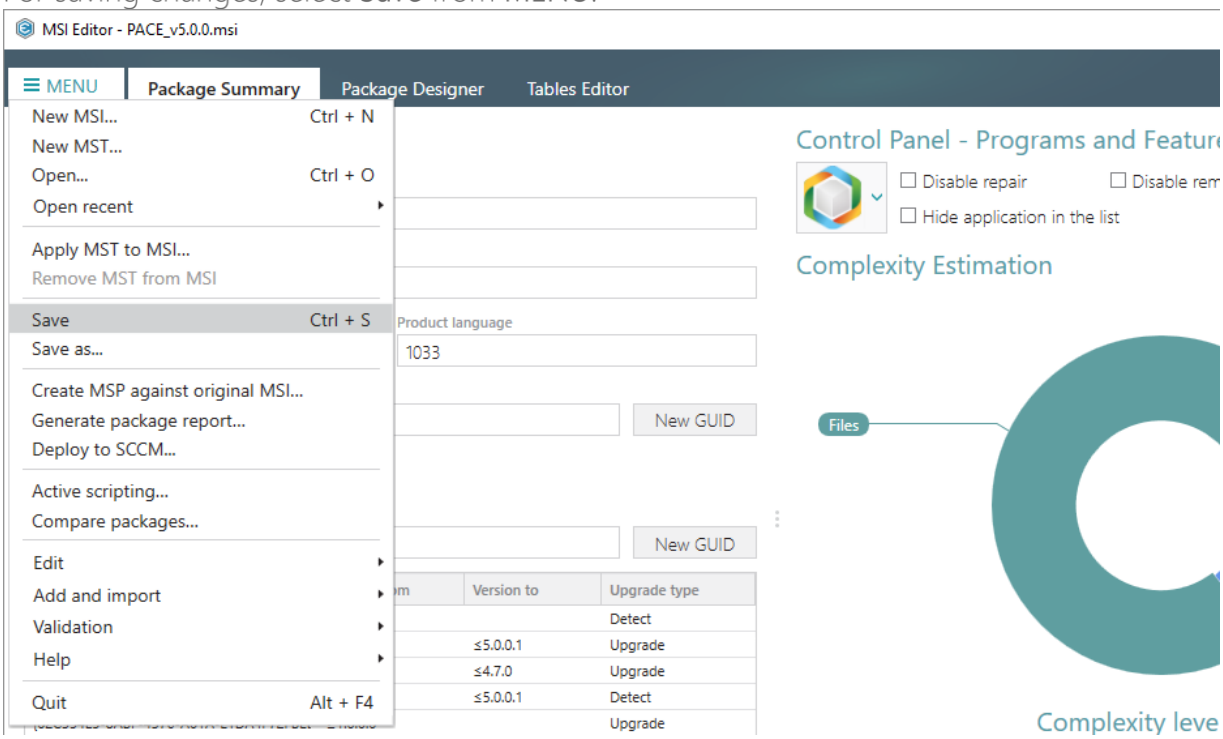



- At the last step, you can change a position, where you would like to run your custom action. The sequence drop-down list allows you to select only compatible positions, taking into account your previously selected options. Also, you free to update the execution conditions either by entering them manually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

NOTE Incompatible sequences are disabled.




- For saving changes, select Save from MENU.



## 3.7.6 Drivers

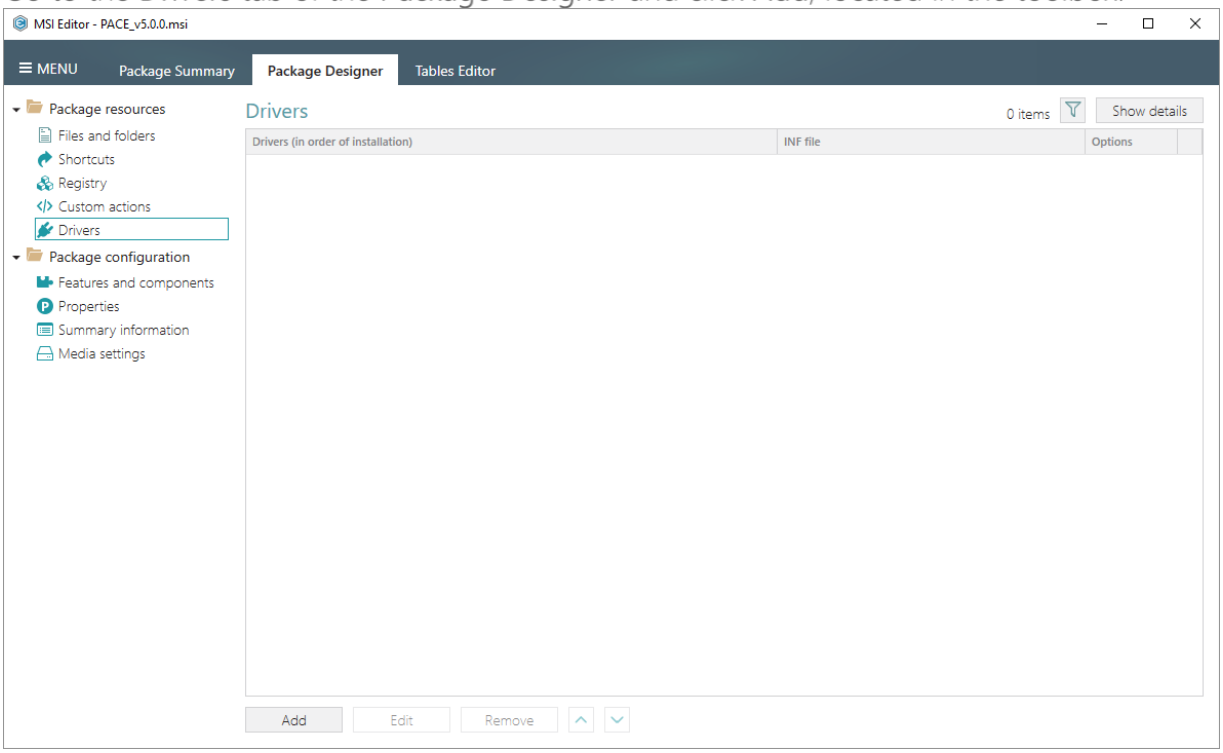
Choose a scenario that better suits your needs:

- Add Driver, described in section 3.7.6.1
  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 3.7.6.2
  Edit installation options, conditions and target architecture of the existing device driver.

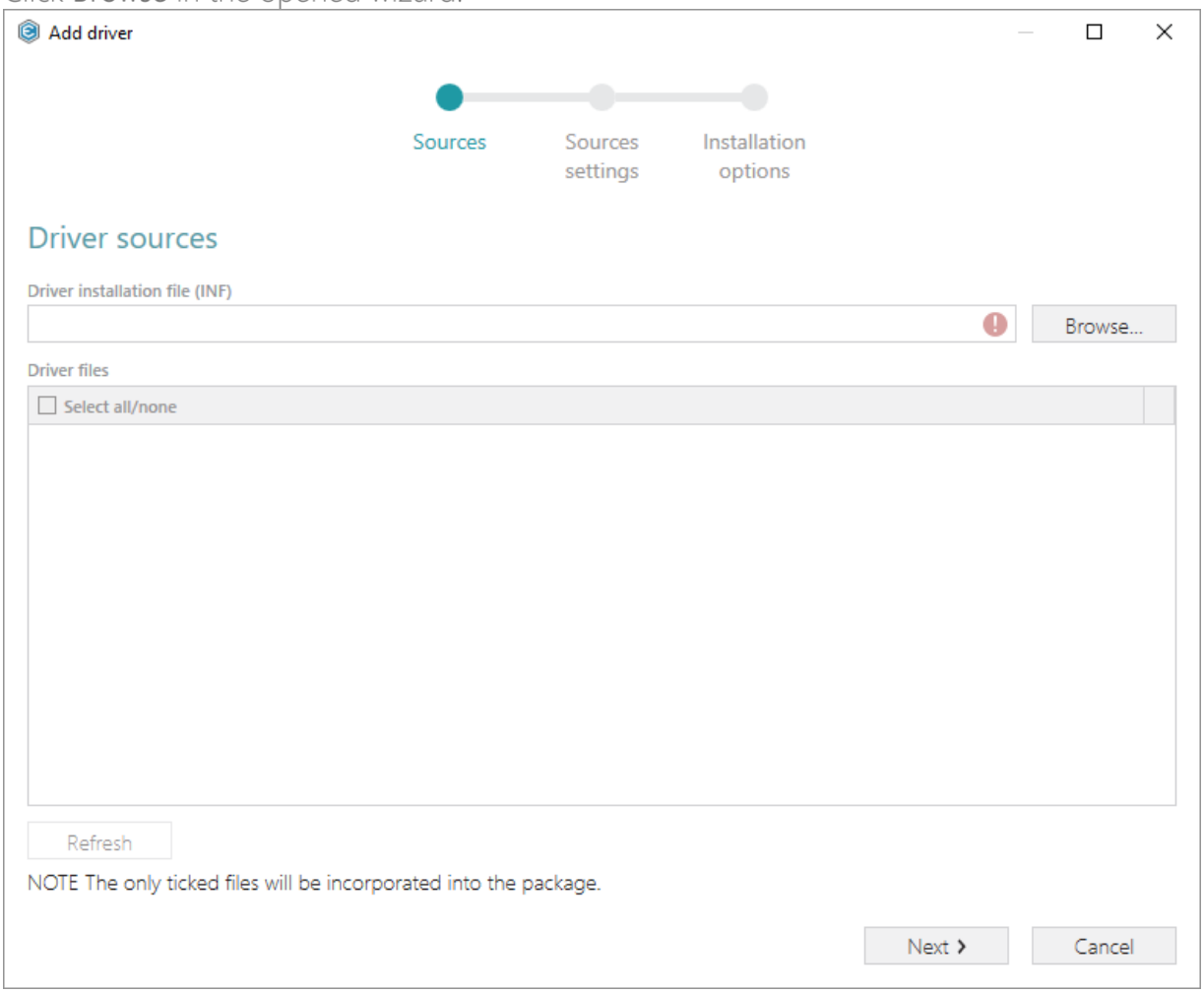
### 3.7.6.1 Add Driver

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

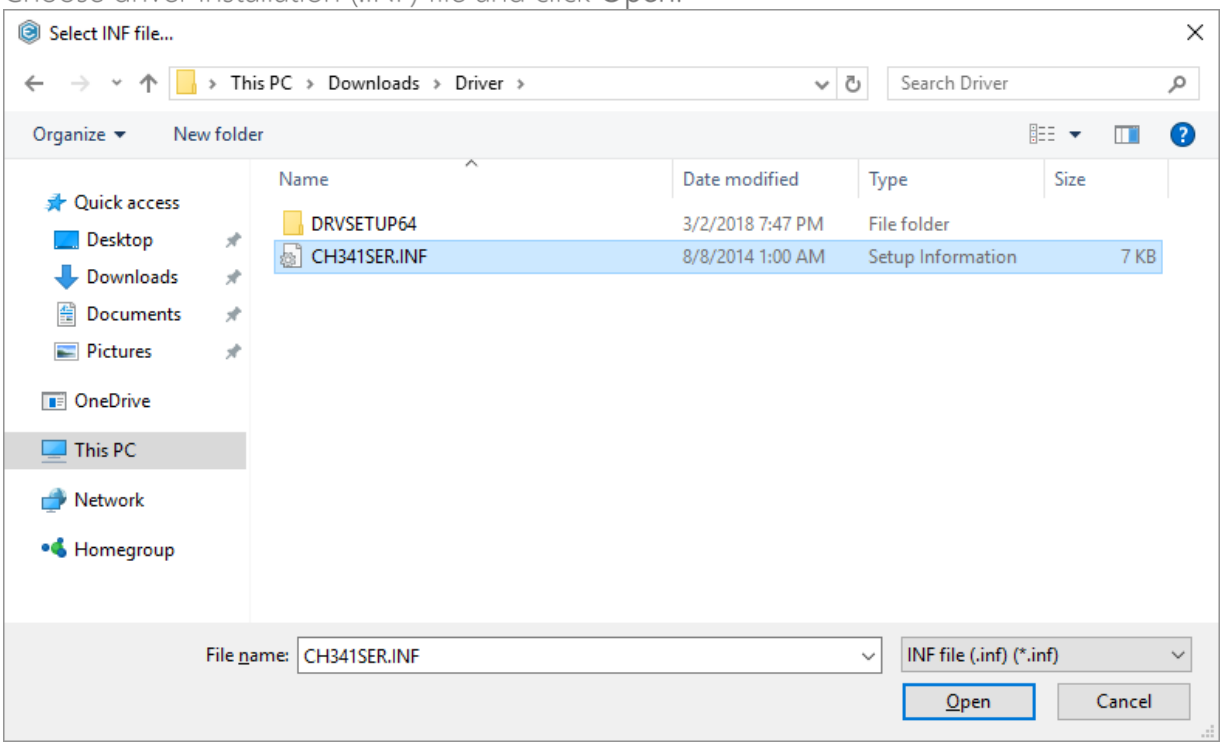
- Go to the Drivers tab of the Package Designer and click Add, located in the toolbox.



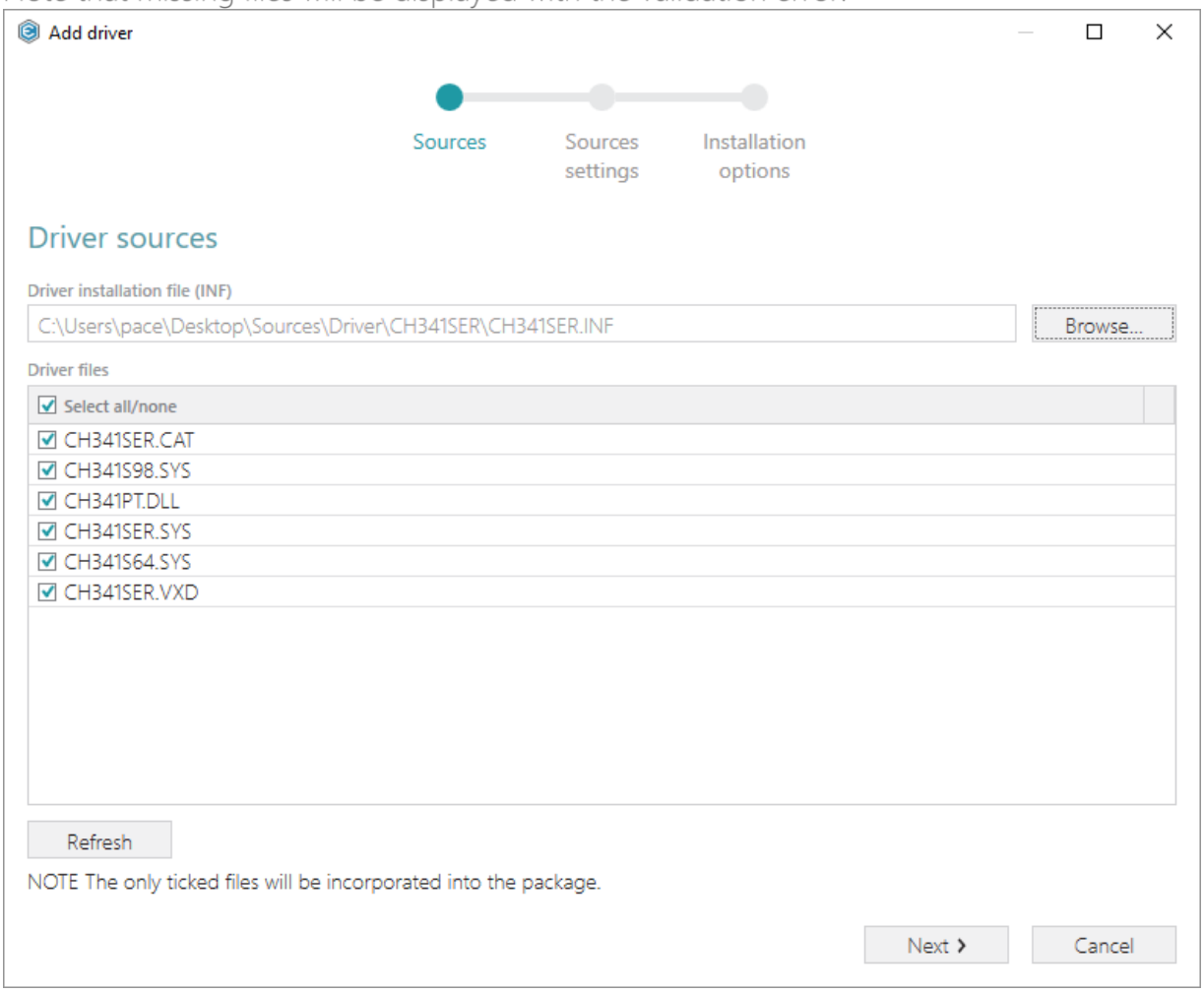
- Click Browse in the opened wizard.




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

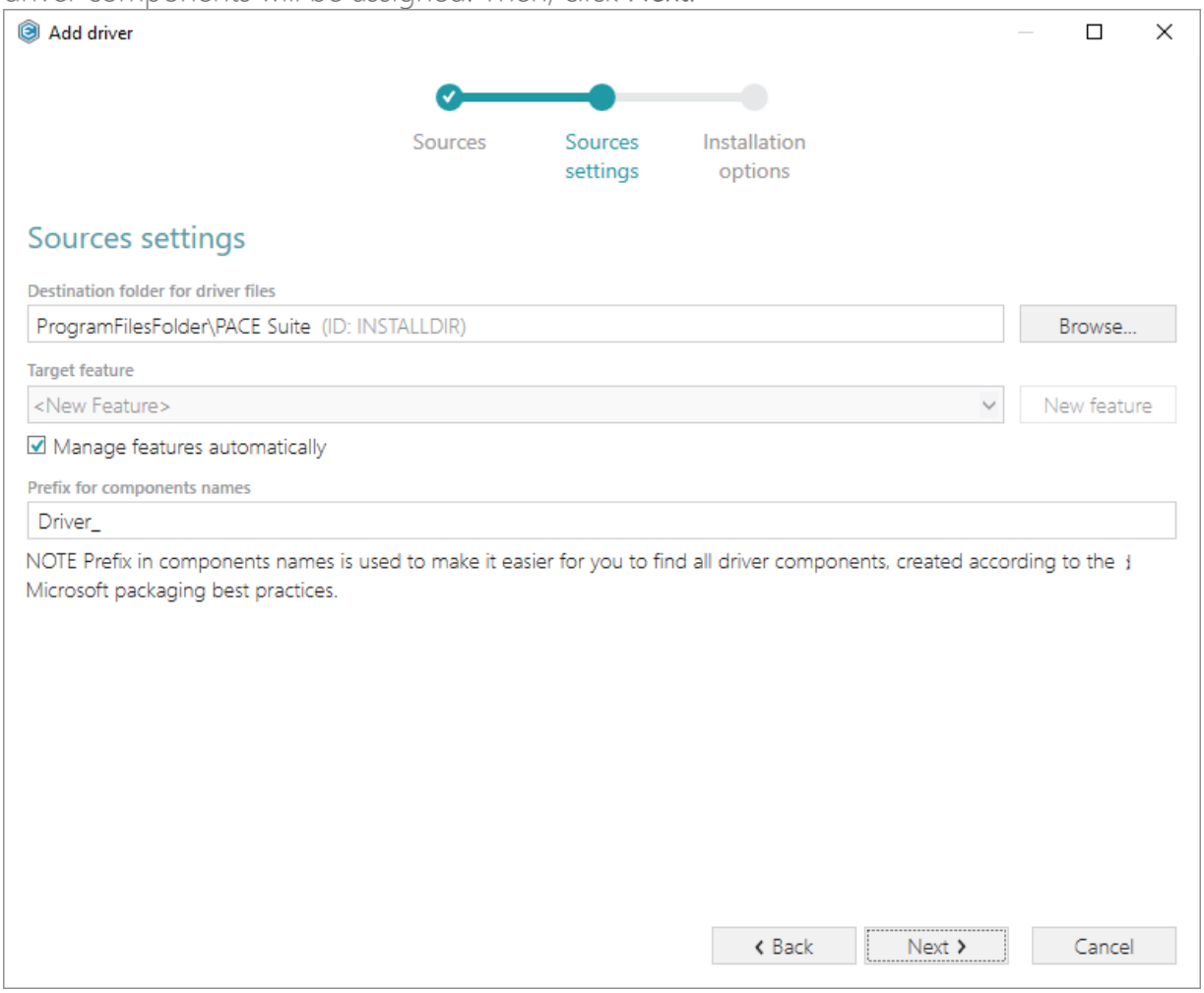


- Ensure that all files, required by driver, exist on the file system, and click **Next**.
  Note that missing files will be displayed with the validation error.



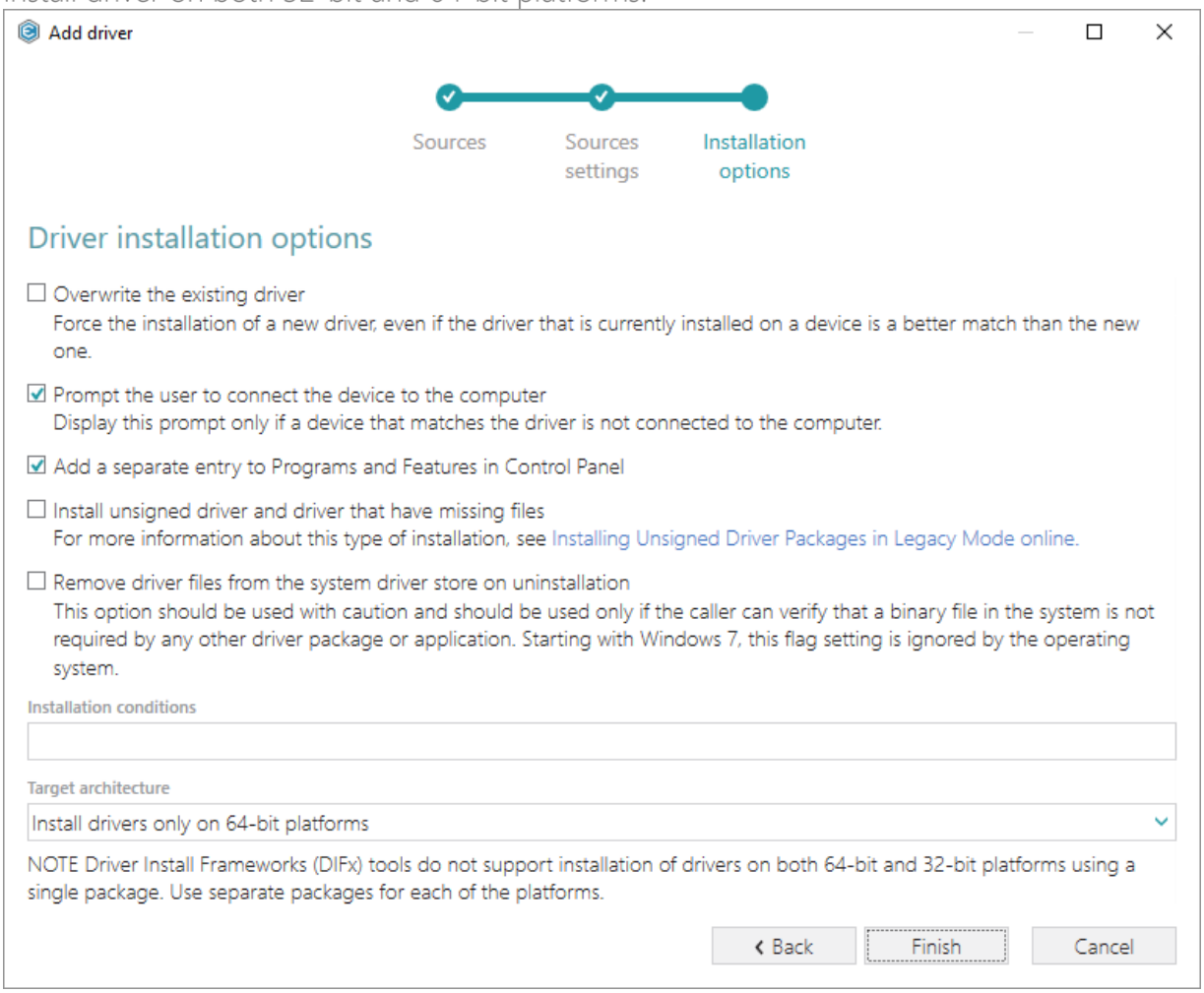

- Select a **Destination folder for the driver files** (the wizard additionally creates a subfolder in the selected folder). Also, you can manually specify a **Target feature**, to which the created

driver components will be assigned. Then, click **Next**.

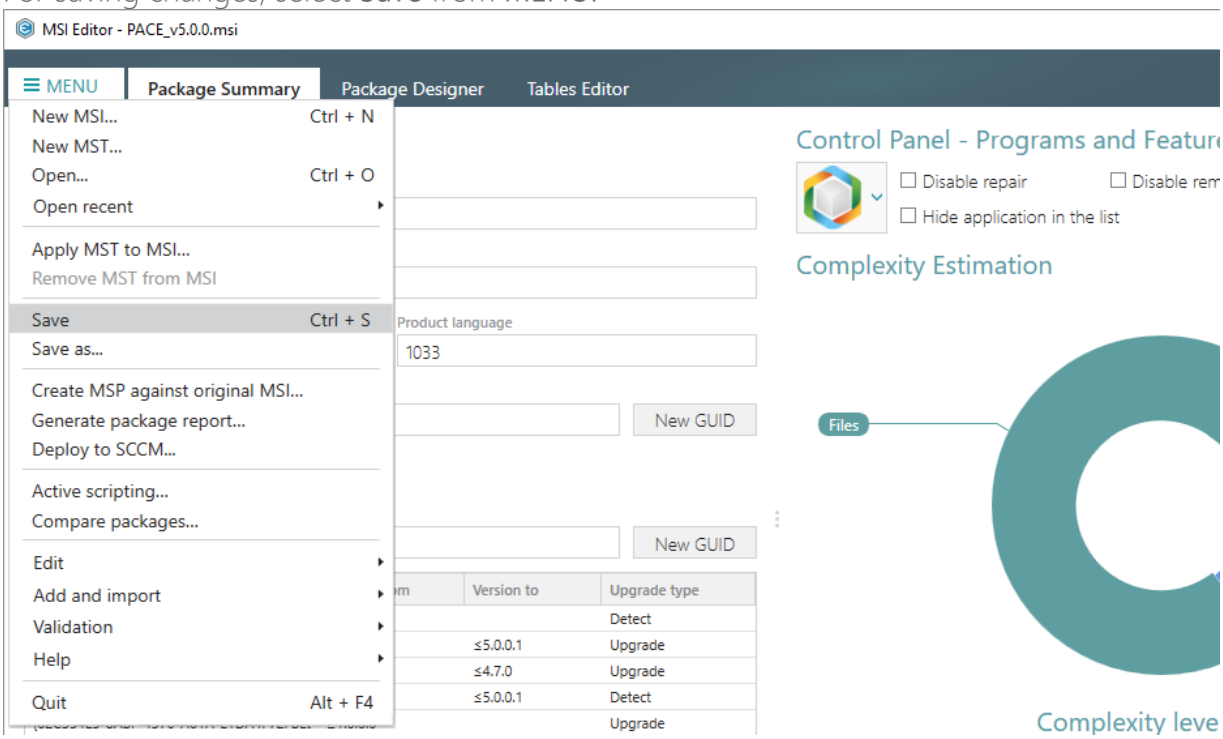



- At the last step, you can select **driver installation options**, used by Driver Install Frameworks (DIFx) tool, specify **installation conditions**, which will be used for the main driver component (see Conditional Statement Syntax), and select a **target architecture**, which defines whether a 32-bit 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.



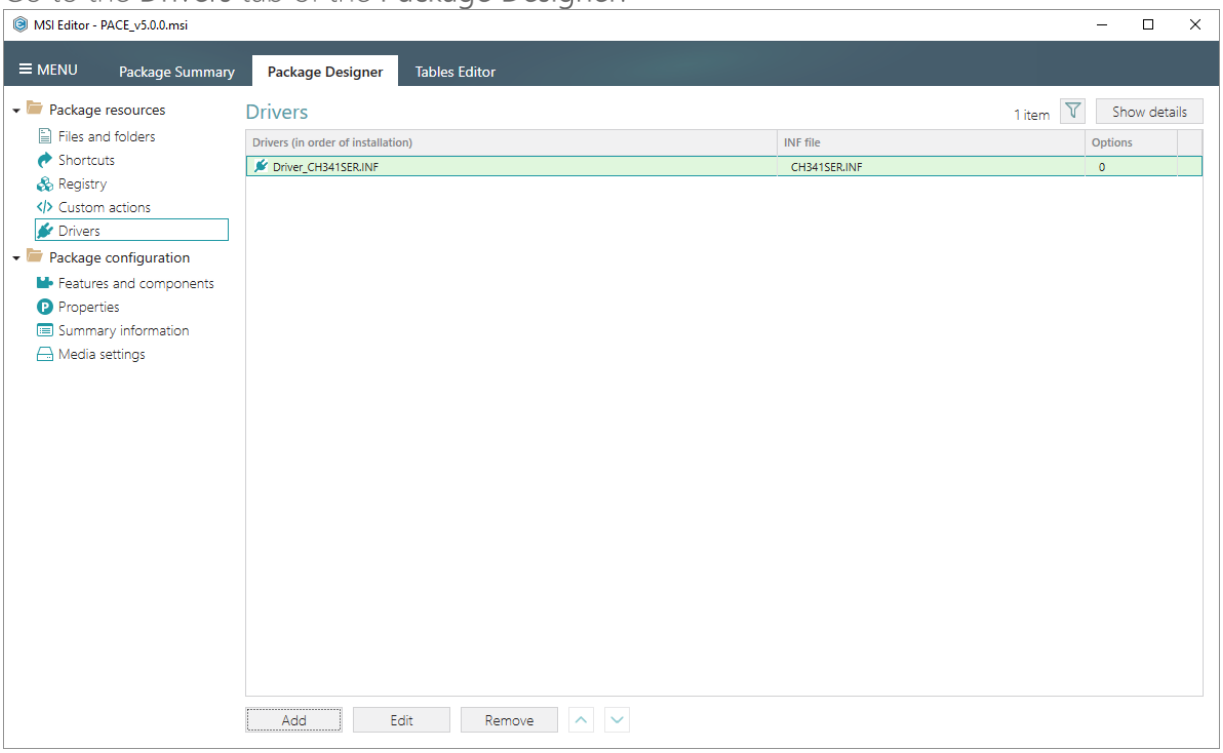

- For saving changes, select Save from MENU.



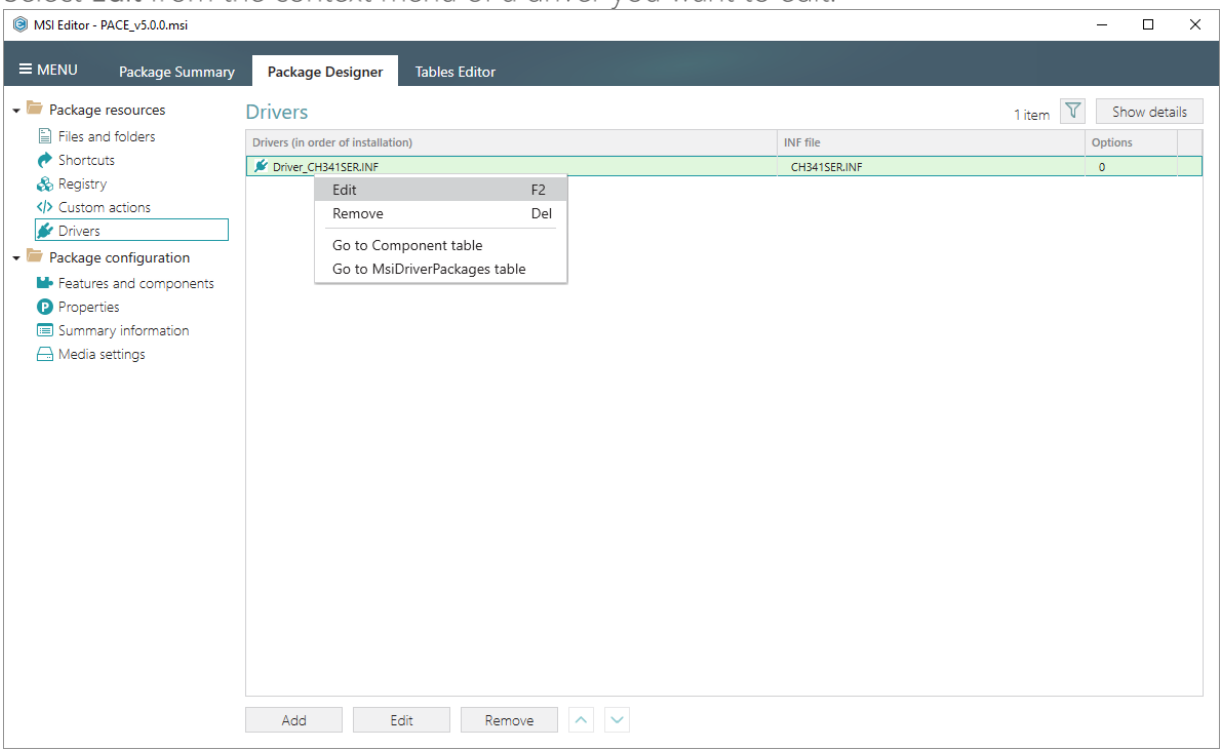
### 3.7.6.2 Edit Driver

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

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

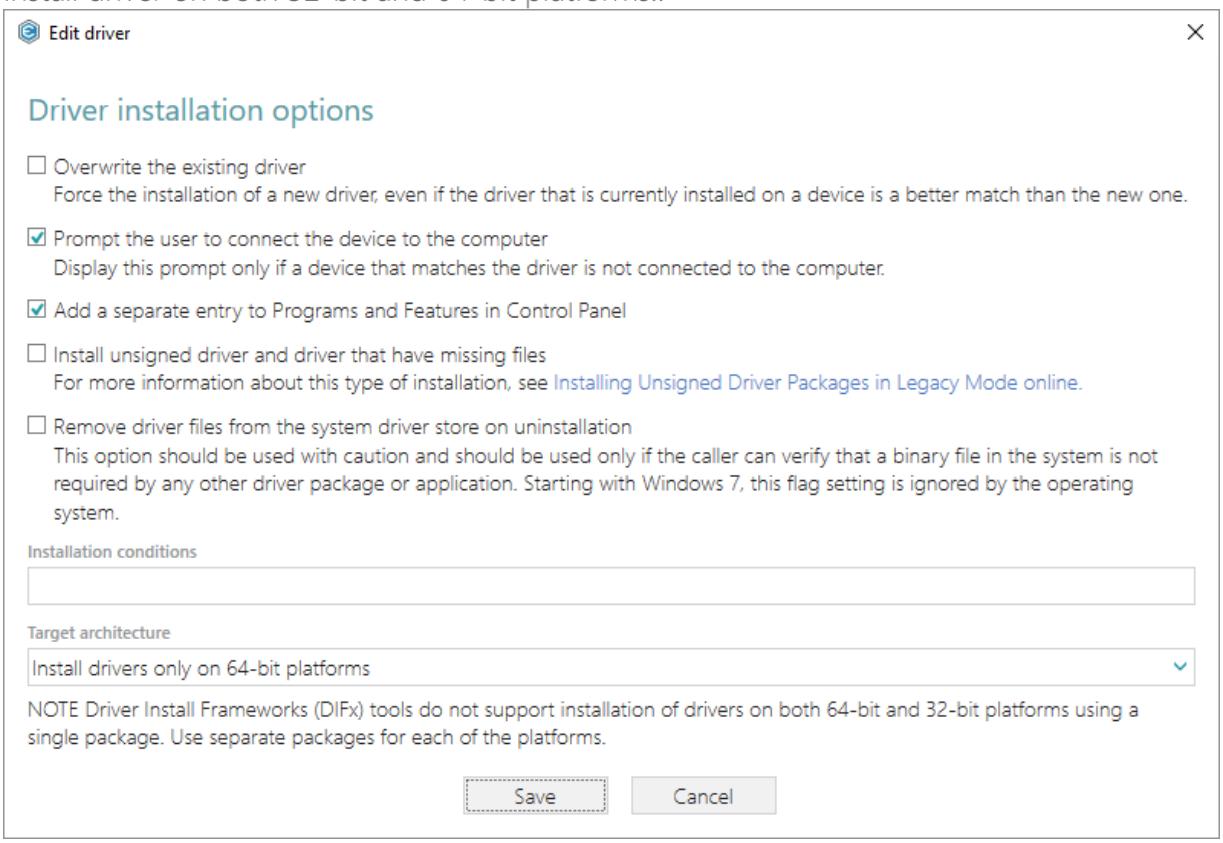



- Select **Edit** from the context menu of a driver you want to edit.

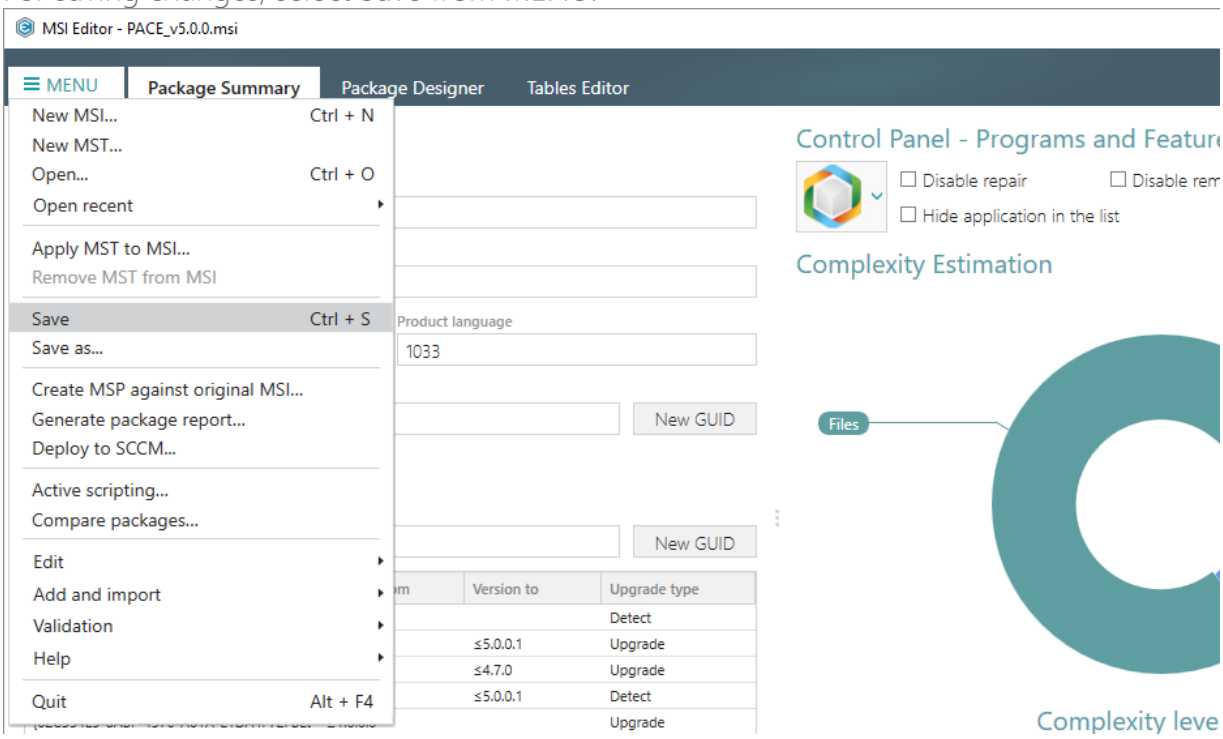



- In the opened window you can change **driver installation options**, used by Driver Install Frameworks (DIFx) tool, specify **installation conditions**, which will be used for the main driver component (see Conditional Statement Syntax), and change a **target architecture**, which defines whether a 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 changes, select Save from MENU.



### 3.7.7 Features and Components

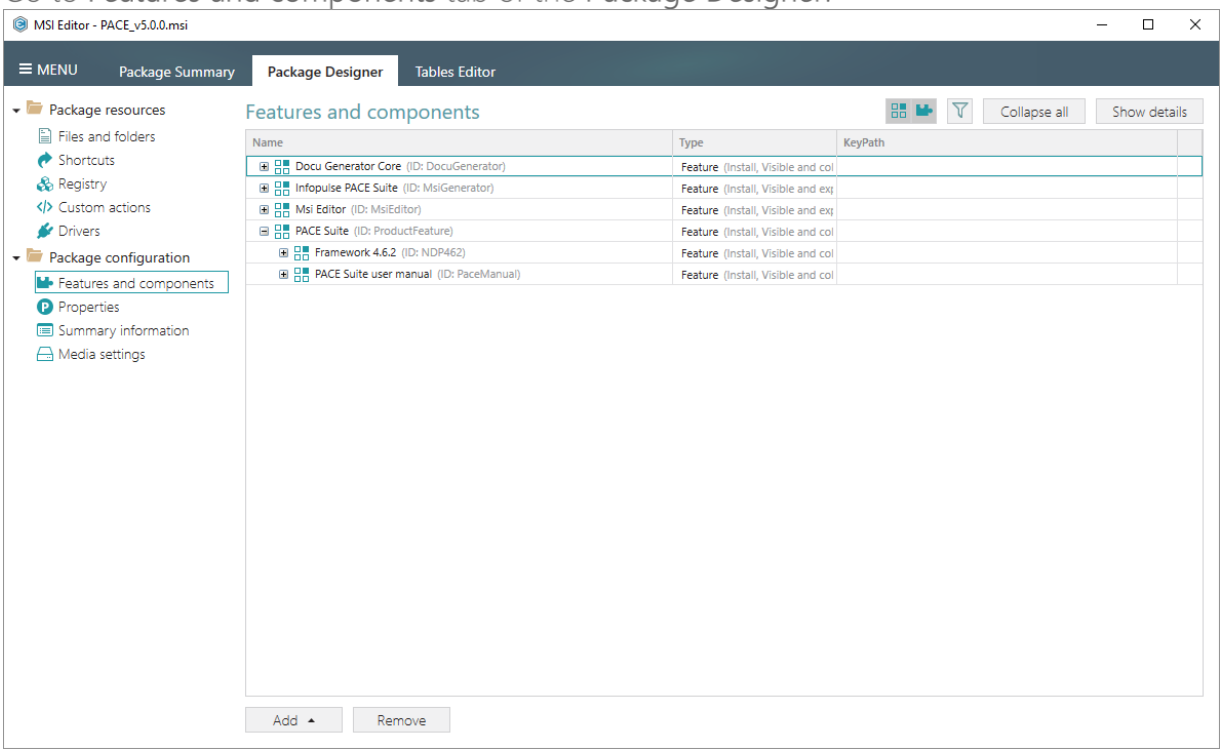
Choose a scenario that better suits your needs:

- Add Feature, described in section 3.7.7.1
  Add a new root feature or subfeature to the package.
- Add Component, described in section 3.7.7.2
  Add a new component to one or many features and set the component properties.
- Move Feature, Component, described in section 3.7.7.3
  Move a feature or a component to another feature.
- Edit Feature Properties, described in section 3.7.7.4
  Edit feature properties such as ID, title, folder, installation state, type, display options and advertisement.
- Edit Component Properties, described in section 3.7.7.5
  Edit component properties such as name, GUID, conditions, key path, attributes and relationship with features.
- Find Empty Components, described in section 3.7.7.6
  Find empty components with no linked resources in order to remove them.
- Review Component Resources, described in section 3.7.7.7
  Review what resources (files, registry, shortcuts, etc.) are linked with the component.

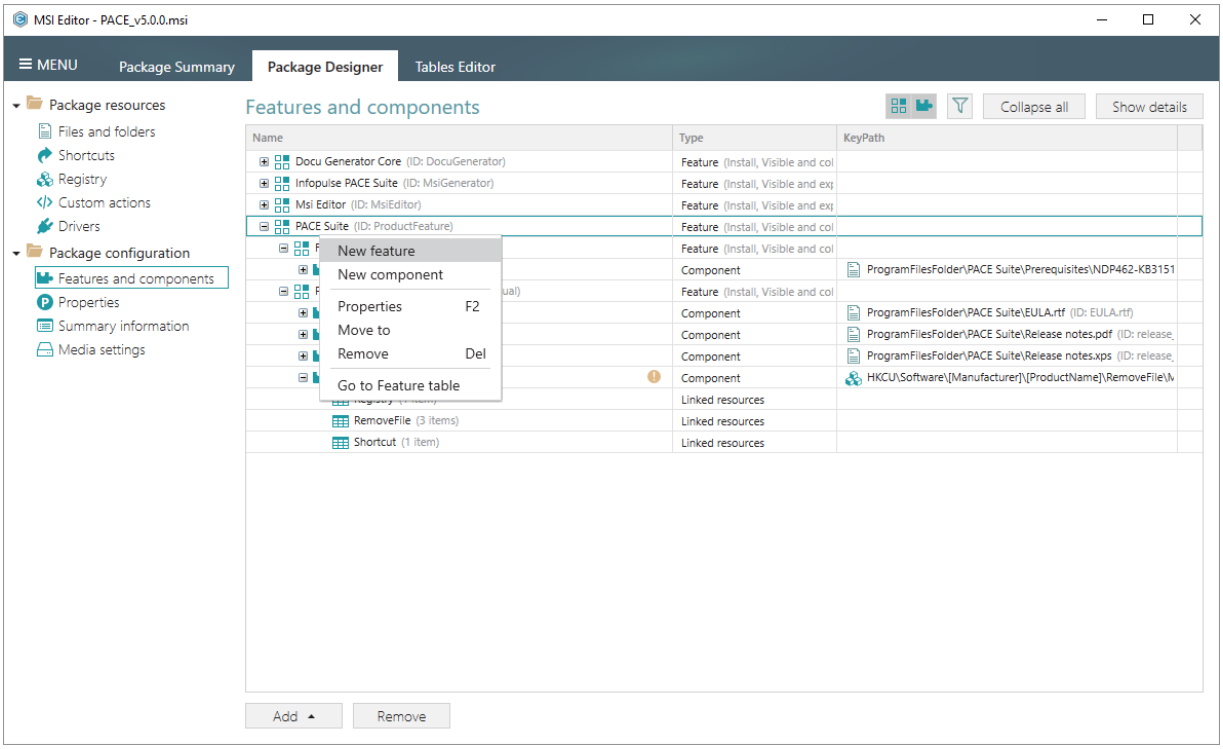
### 3.7.7.1 Add Feature

Add a new root feature or subfeature to the package.

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

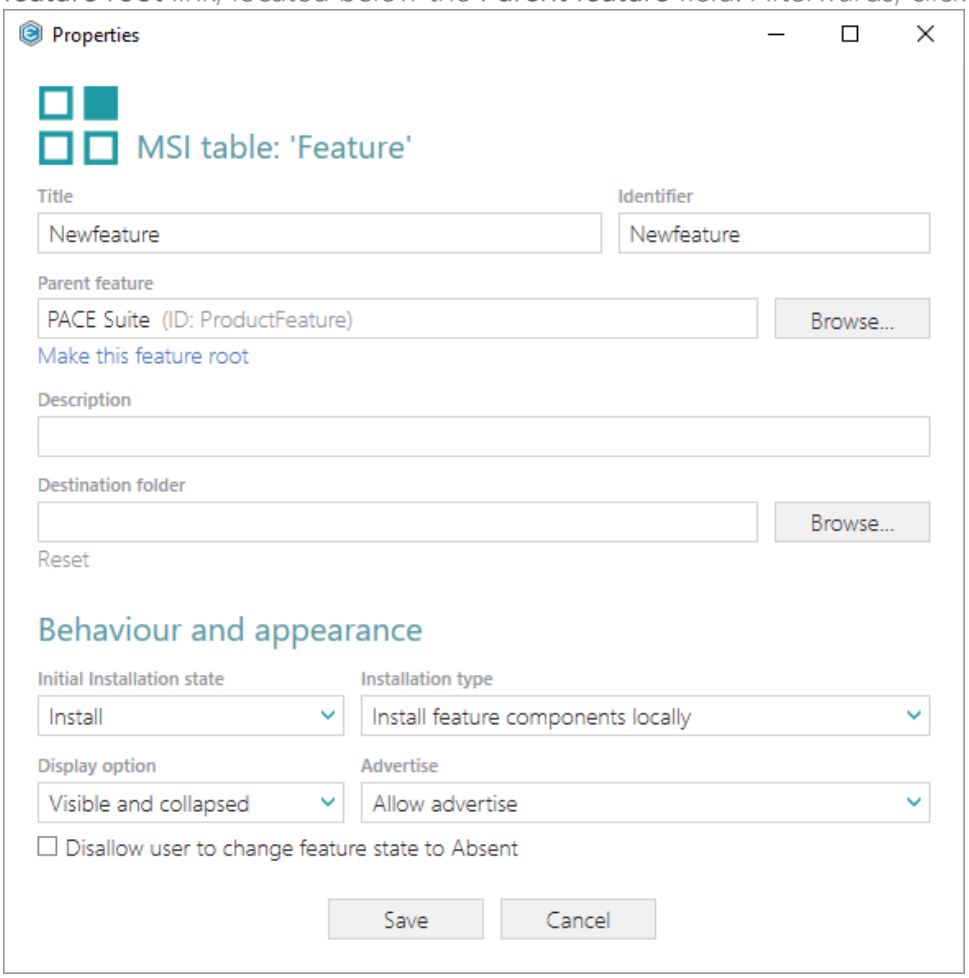



- Select New feature from the context menu of a feture, where you want to create a new feature. For creating a root feature, use any feature and later you can clean-up its parent feature.

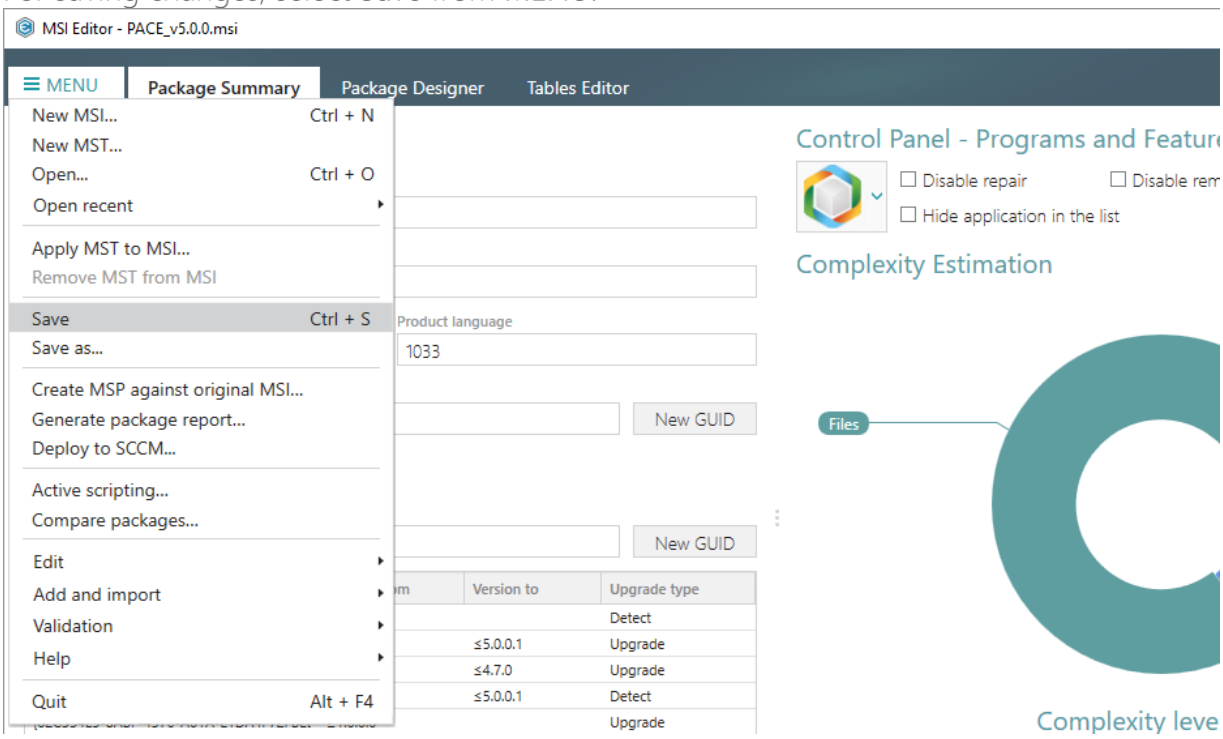



- Using the Properties window, update a feature title, identifier and select the necessary behavior and appearance settings. For creating a root feature, click on the Make this

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



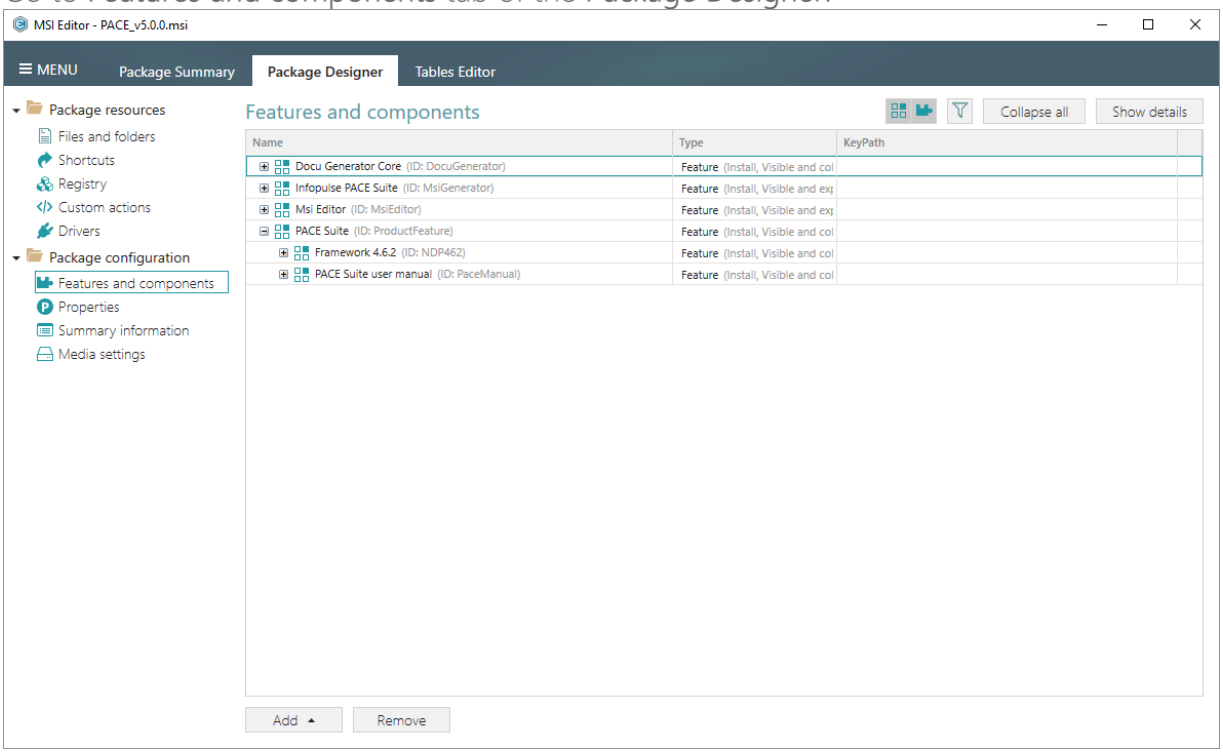

- For saving changes, select Save from MENU.



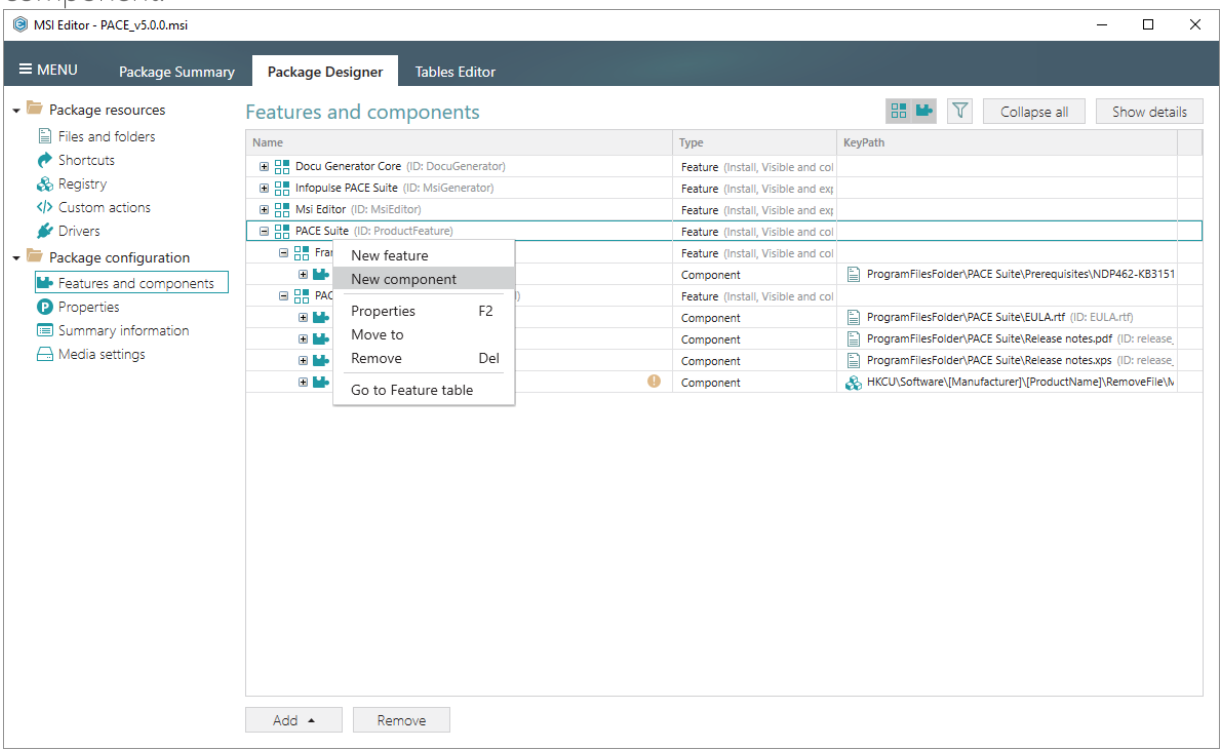
### 3.7.7.2 Add Component

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

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

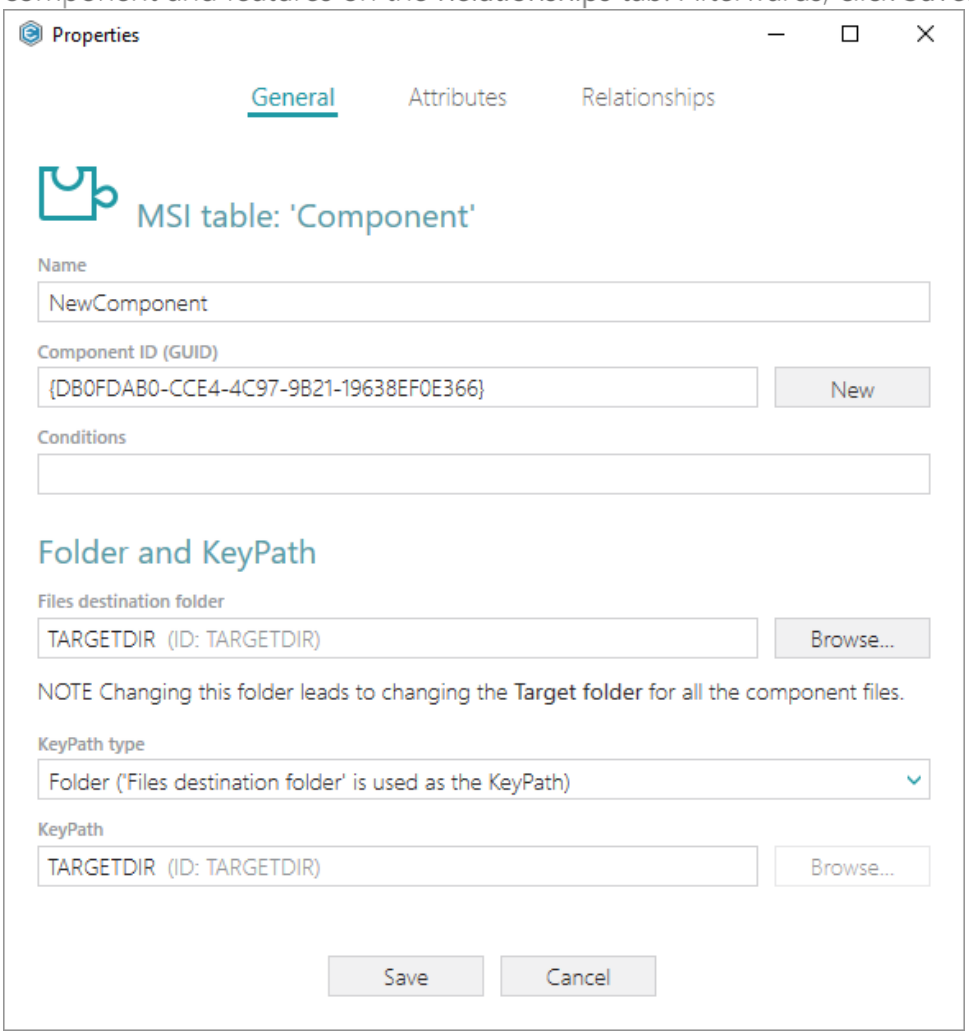



- Select New component from the context menu a feature, where you want to create a new component.

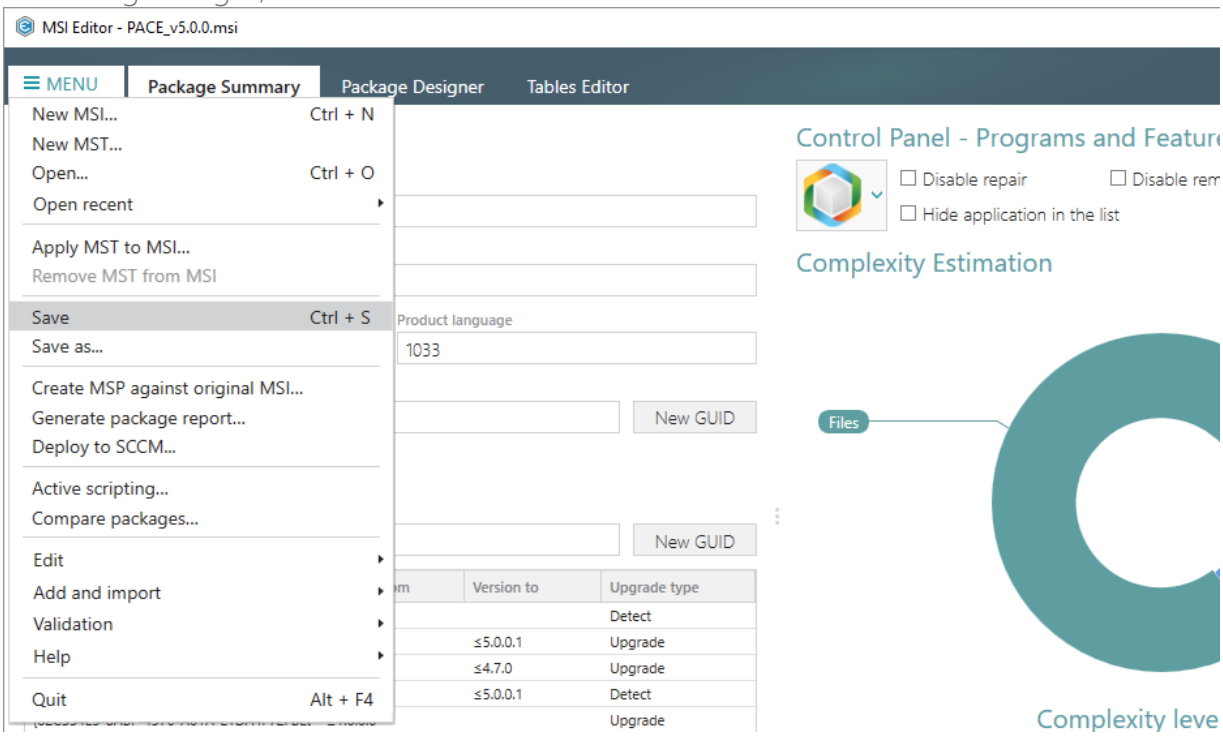



- Using the Properties window, update a component name, GUID and select the necessary installation conditions, target folder. Note that to select a key path from files or registry, this component should contain these resources. Switch to the Attributes tab to manage

component attributes and define new or change existing relationships between this component and features on the Relationships tab. Afterwards, click Save.



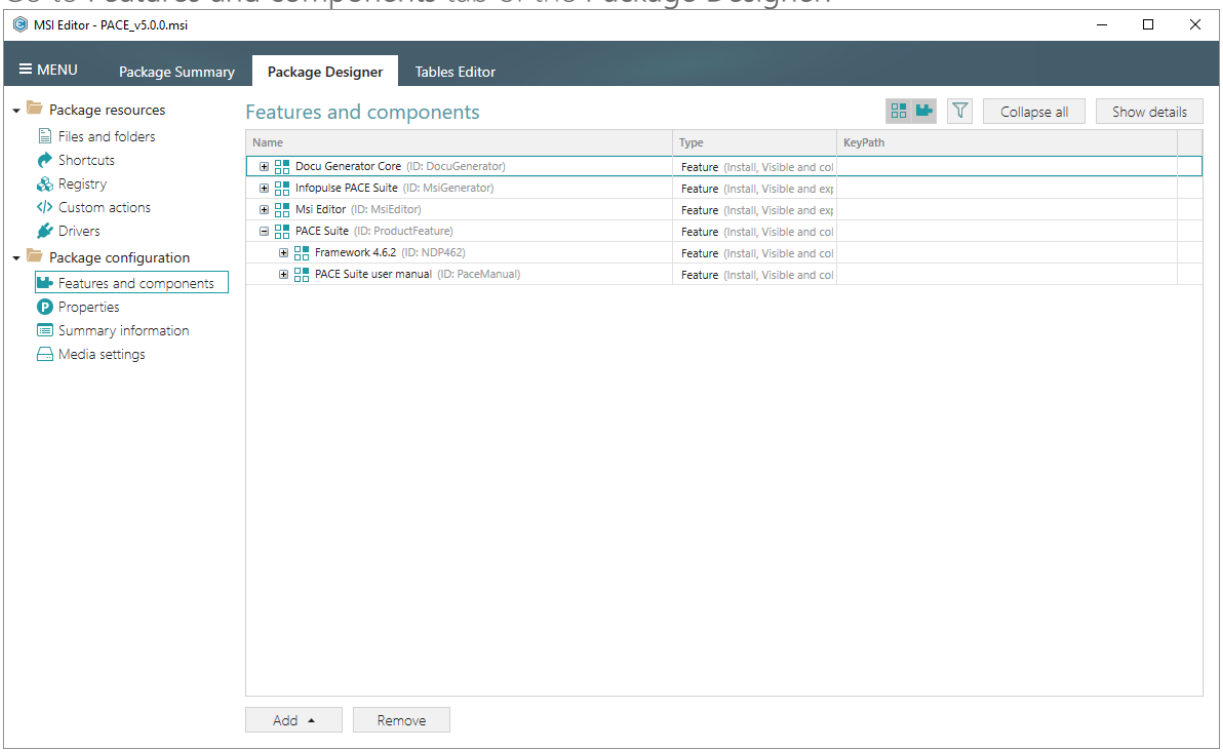

- For saving changes, select Save from MENU.



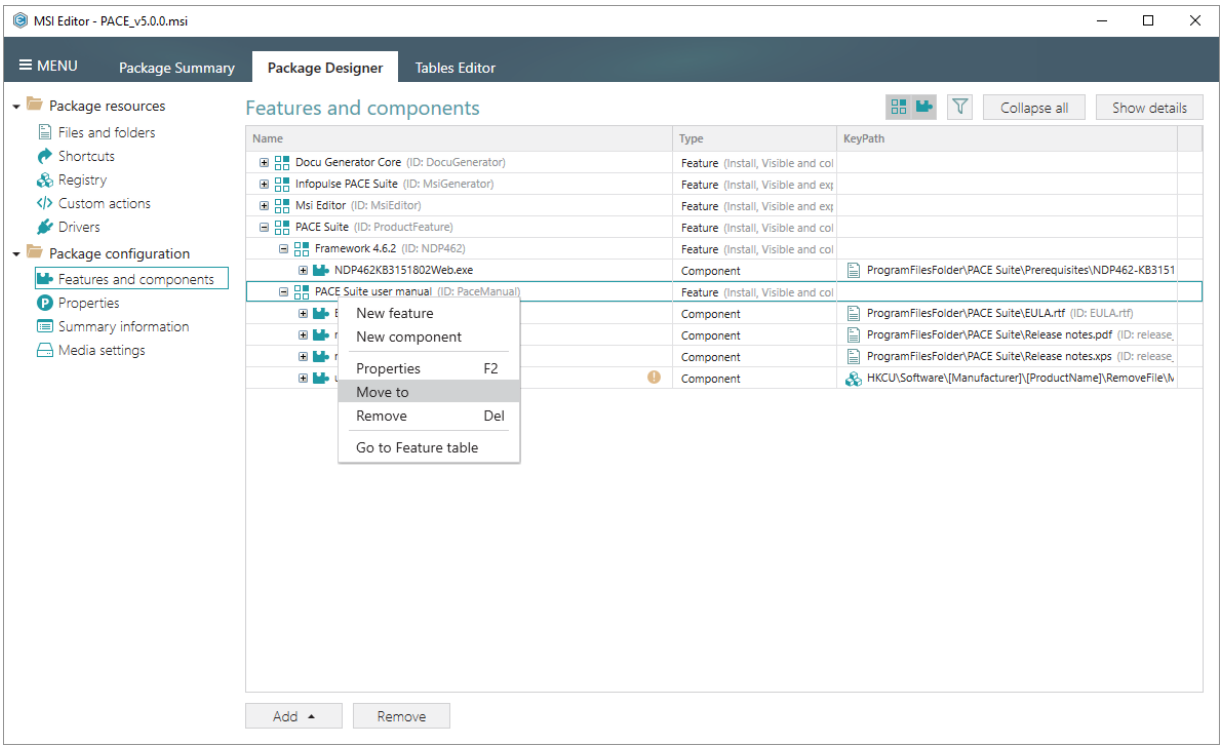
### 3.7.7.3 Move Feature, Component

Move a feature or a component to another feature.

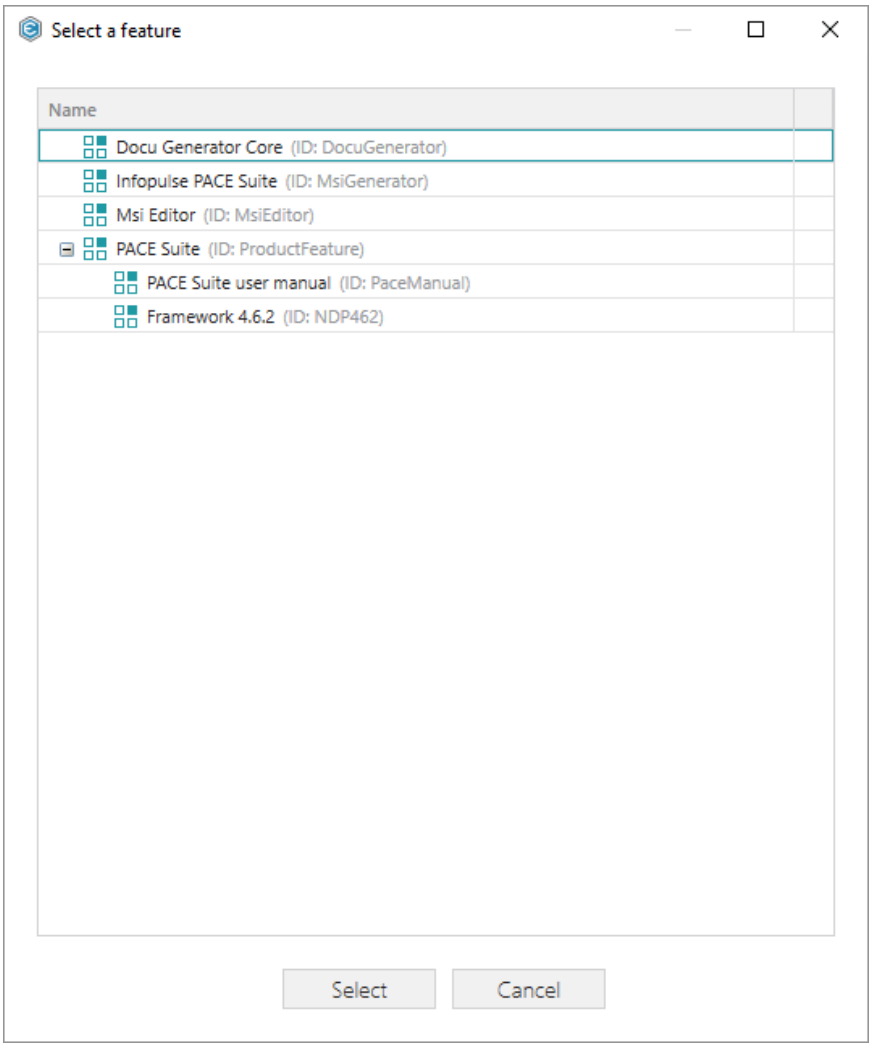
- Go to Features and components tab of the Package Designer.



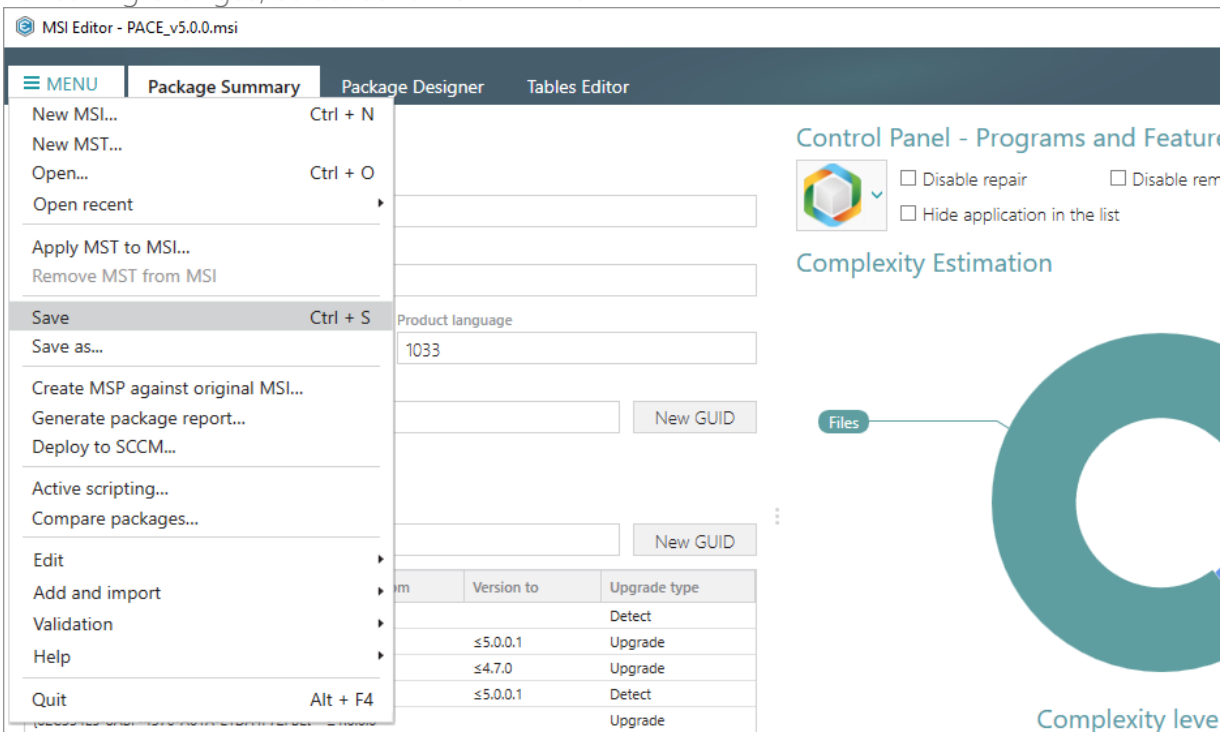

- Select Move to from the context menu of a feature or component, which you want to move to another feature.



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



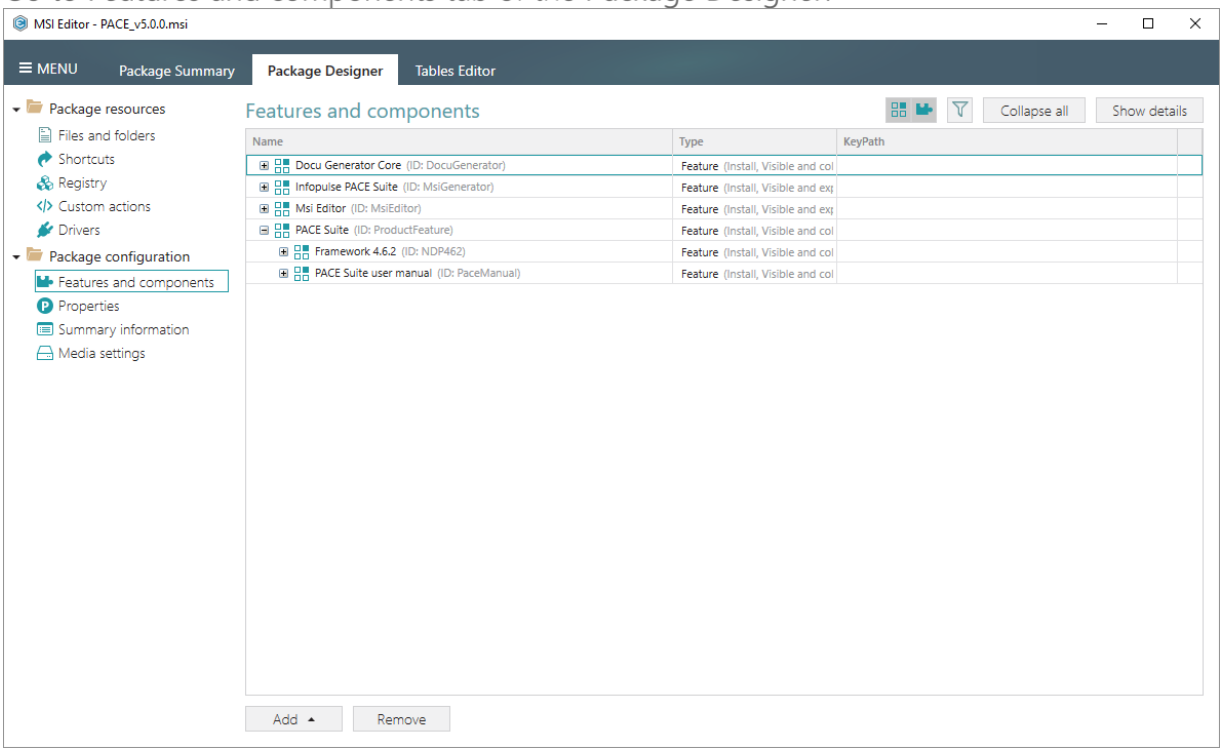

- For saving changes, select **Save** from **MENU**.



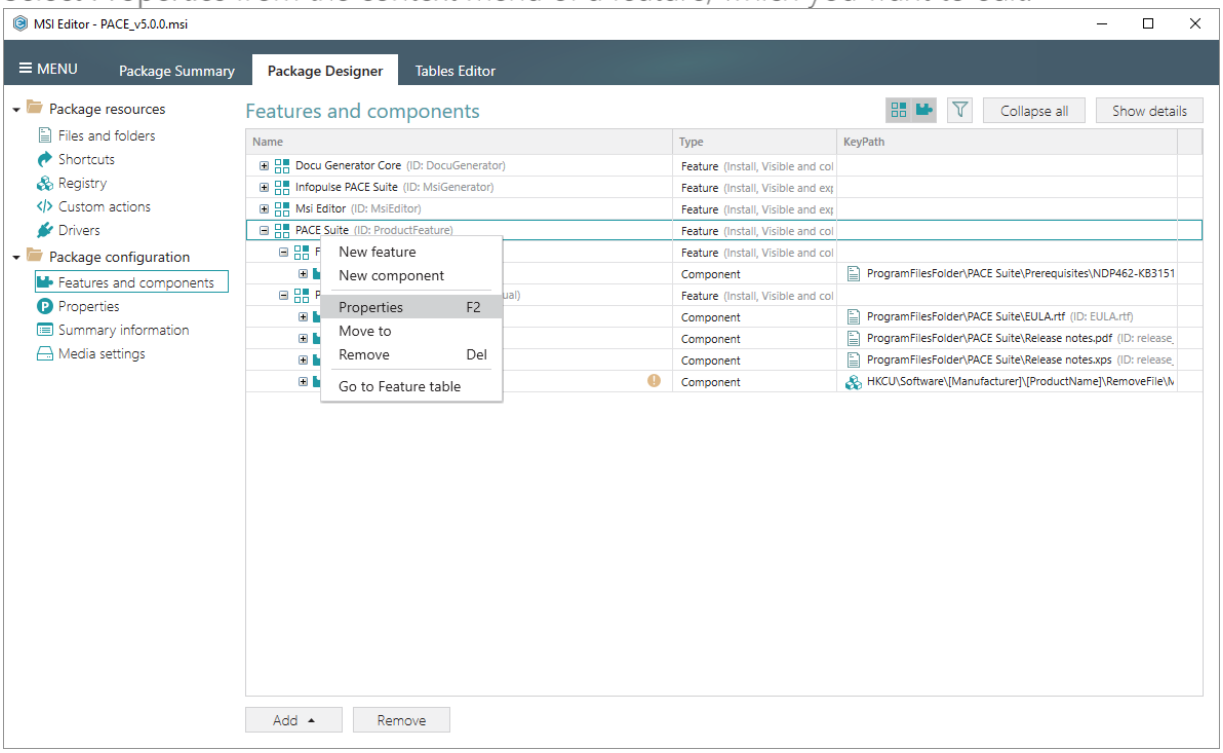
### 3.7.7.4 Edit Feature Properties

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

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

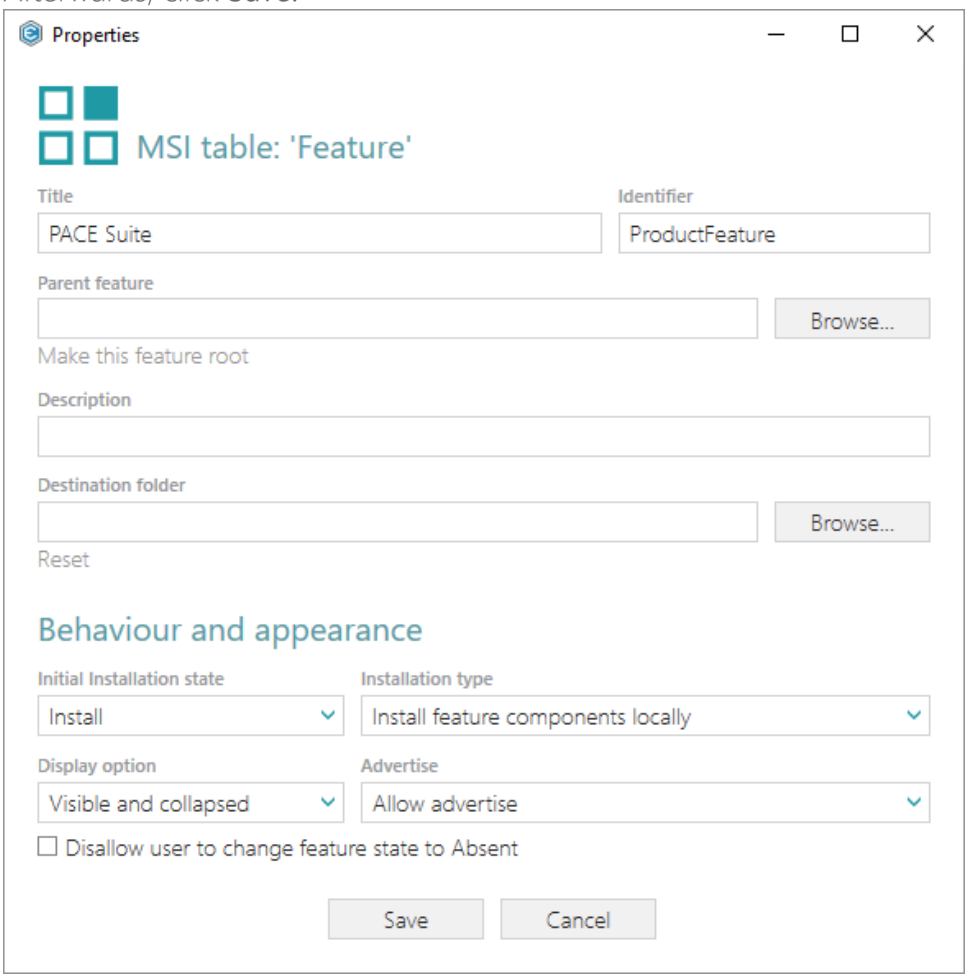



- Select Properties from the context menu of a feature, which you want to edit.

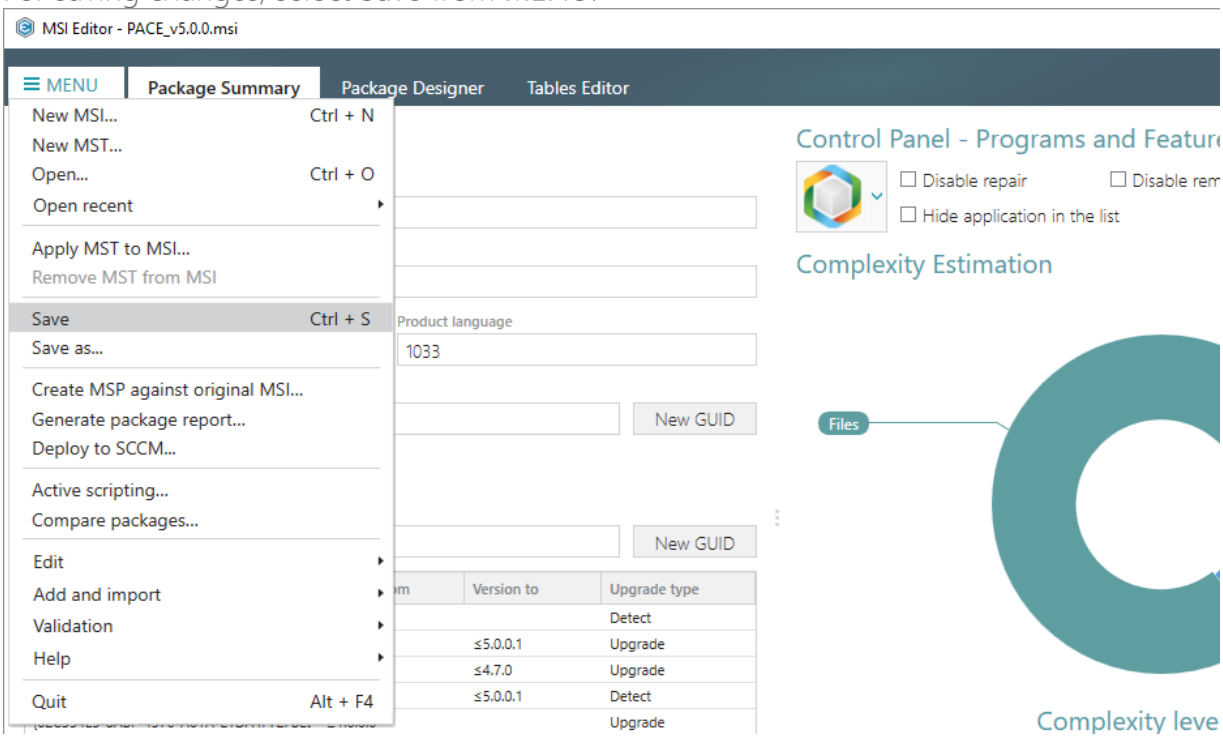



- Using the Properties window, update the necessary feature properties, such as a feature title, identifier, parent feature, installation state and type, display option and advertisement.

Afterwards, click Save.



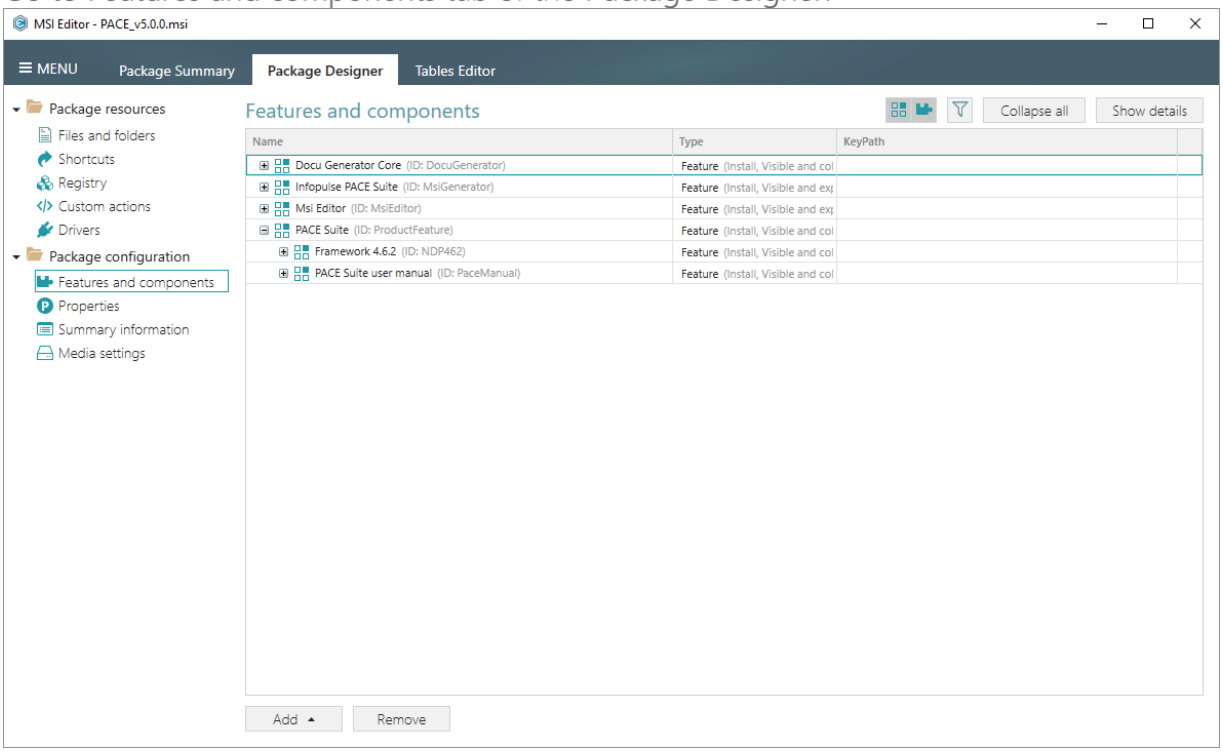

- For saving changes, select Save from MENU.



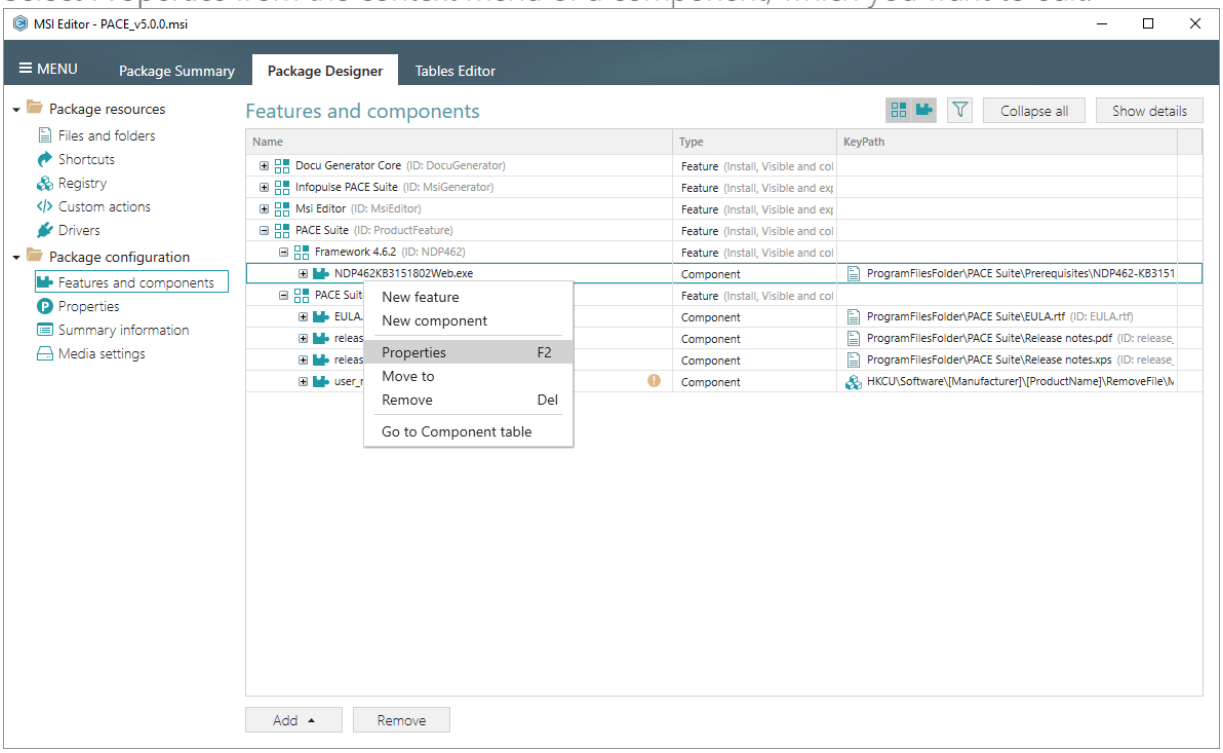
### 3.7.7.5 Edit Component Properties

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

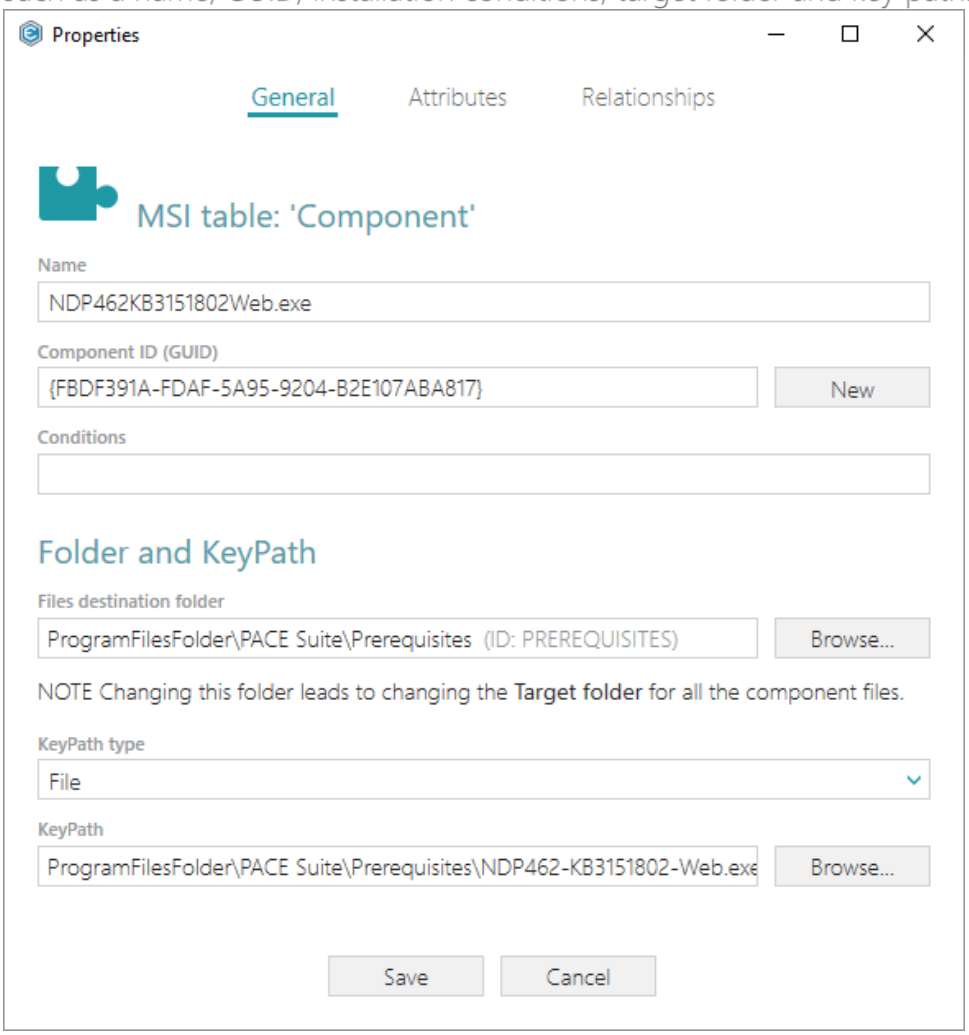
- Go to Features and components tab of the Package Designer.




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

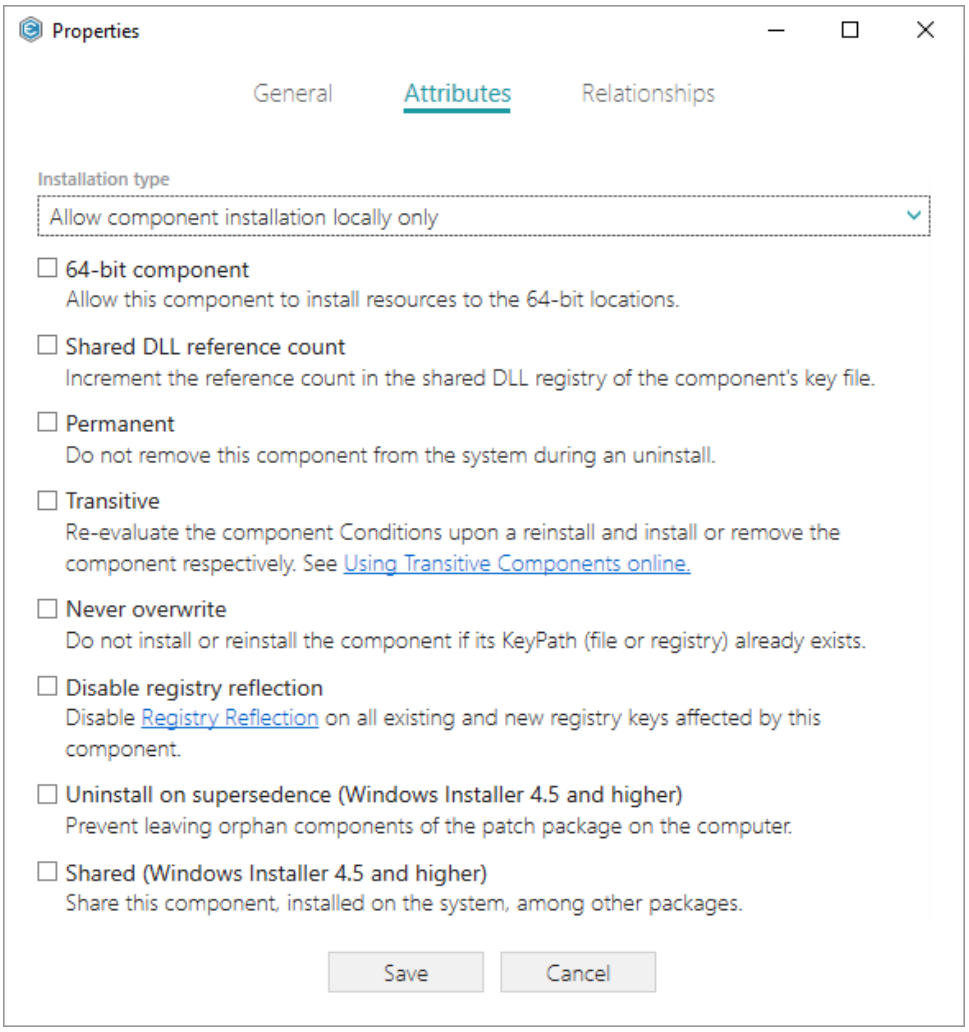


- On the General tab of the Properties window, update the necessary component properties, such as a name, GUID, installation conditions, target folder and key path.

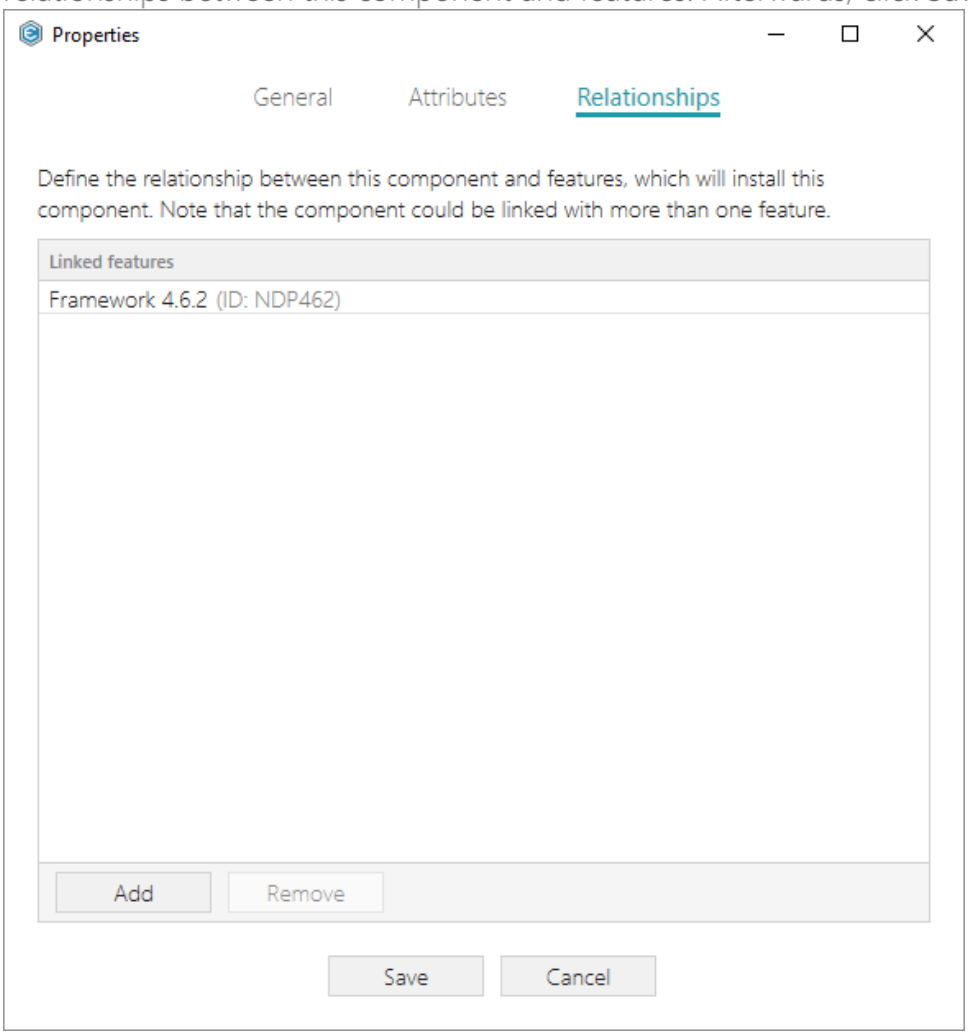



- On the Attributes tab of the Properties window you can manage component attributes such as component installation type, bitness (set the 64-bit component attribute if your component installs resoruces to the 64-bit locations), management of the shared resources

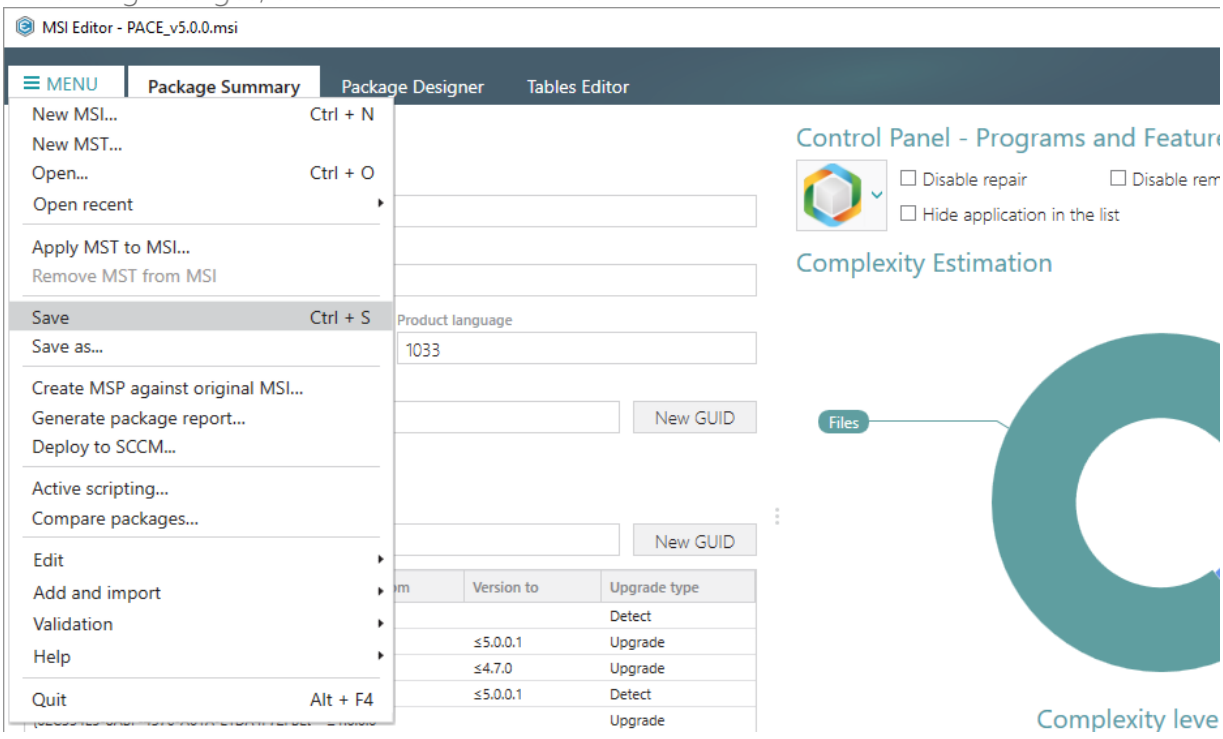
and so on.


Properties
General
Attributes
Relationships
Installation type
Allow component installation locally only
64-bit component
Allow this component to install resources to the 64-bit locations.
Shared DLL reference count
Increment the reference count in the shared DLL registry of the component's key file.
Permanent
Do not remove this component from the system during an uninstall.
Transitive
Re-evaluate the component Conditions upon a reinstall and install or remove the component respectively. See Using Transitive Components online.
Never overwrite
Do not install or reinstall the component if its KeyPath (file or registry) already exists.
Disable registry reflection
Disable Registry Reflection on all existing and new registry keys affected by this component.
Uninstall on supersedence (Windows Installer 4.5 and higher)
Prevent leaving orphan components of the patch package on the computer.
Shared (Windows Installer 4.5 and higher)
Share this component, installed on the system, among other packages.
Save
Cancel

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



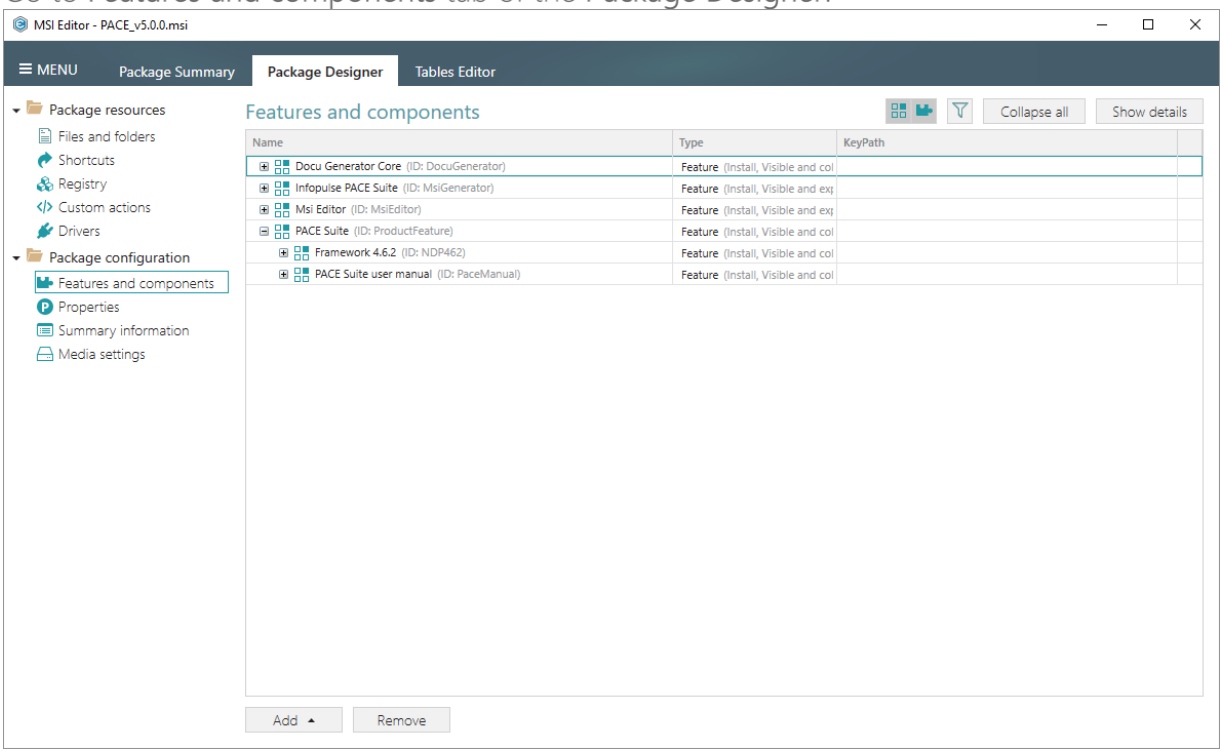

- For saving changes, select **Save** from **MENU**.



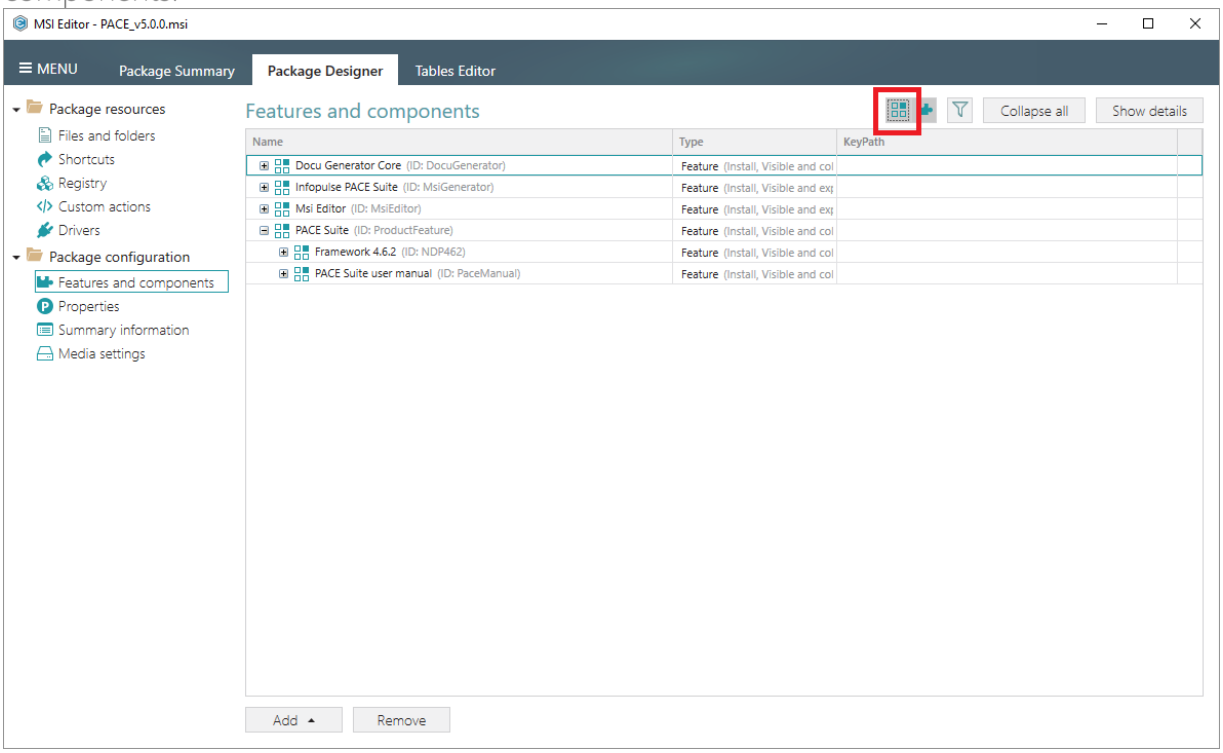
### 3.7.7.6 Find Empty Components

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

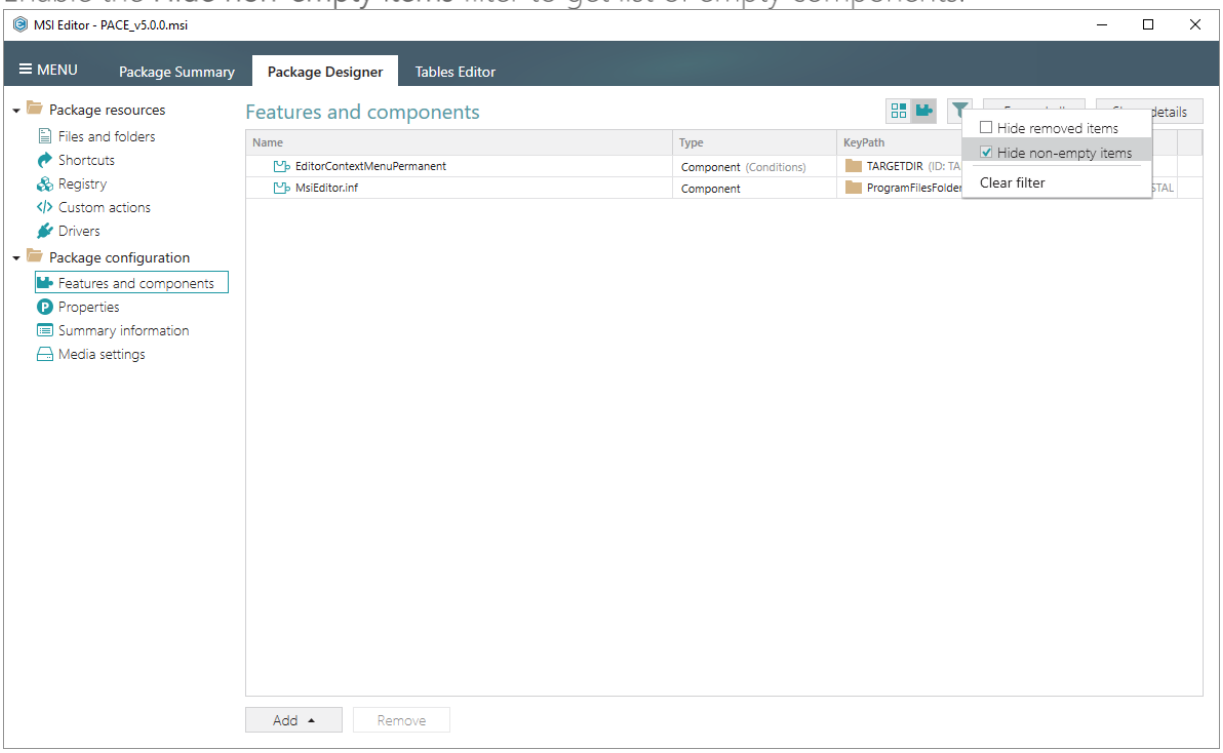
- Go to Features and components tab of the Package Designer.



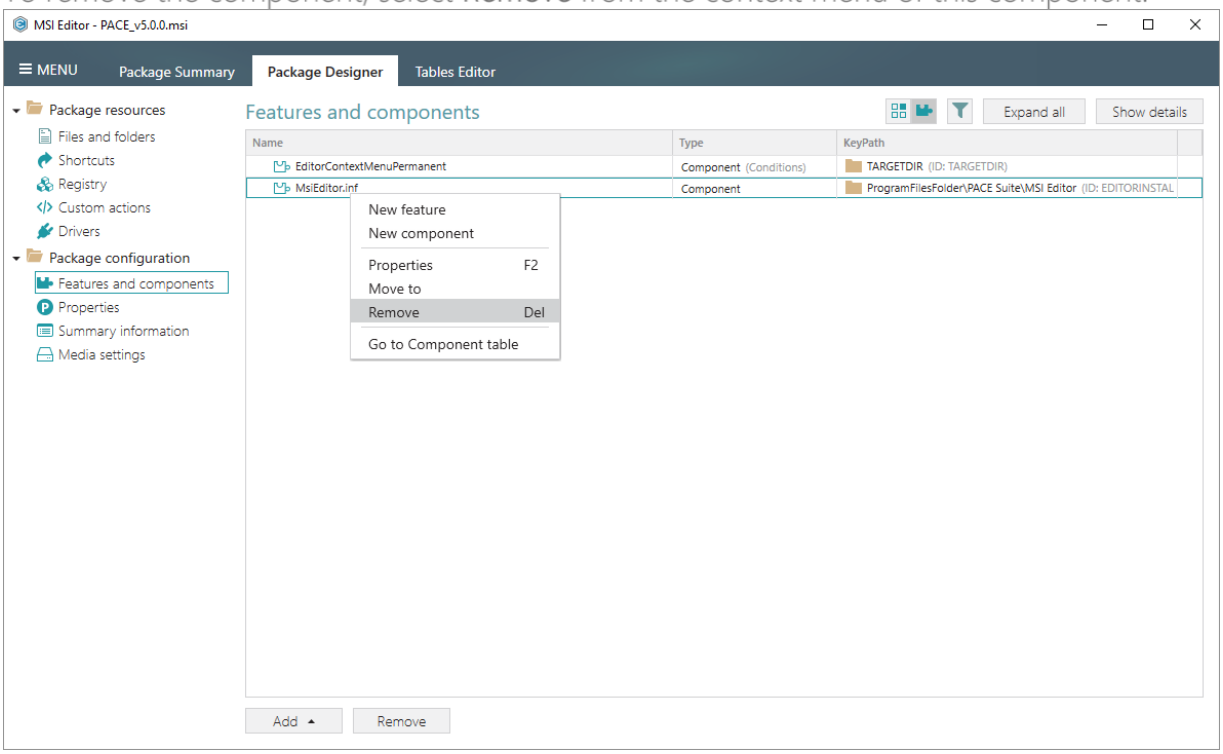

- Click on the 'feature' icon , located near the Filters, to change view for displaying only components.



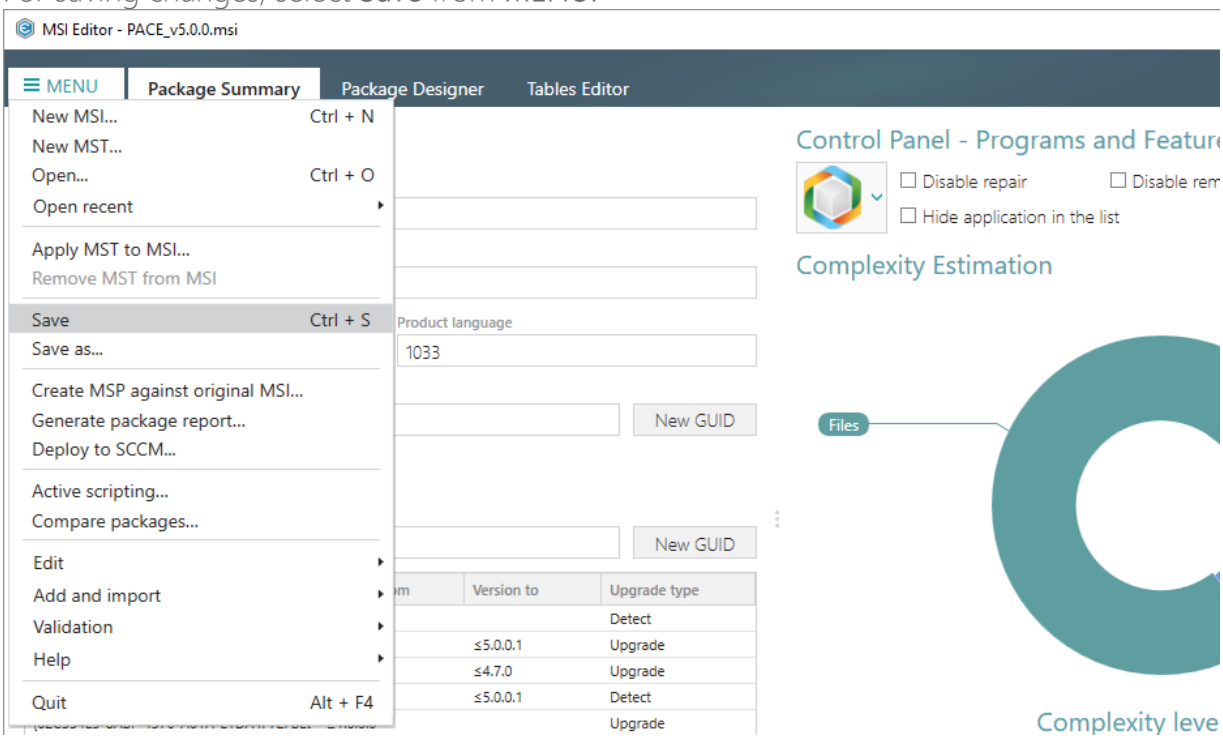
- Enable the **Hide non-empty items** filter to get list of empty components.




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



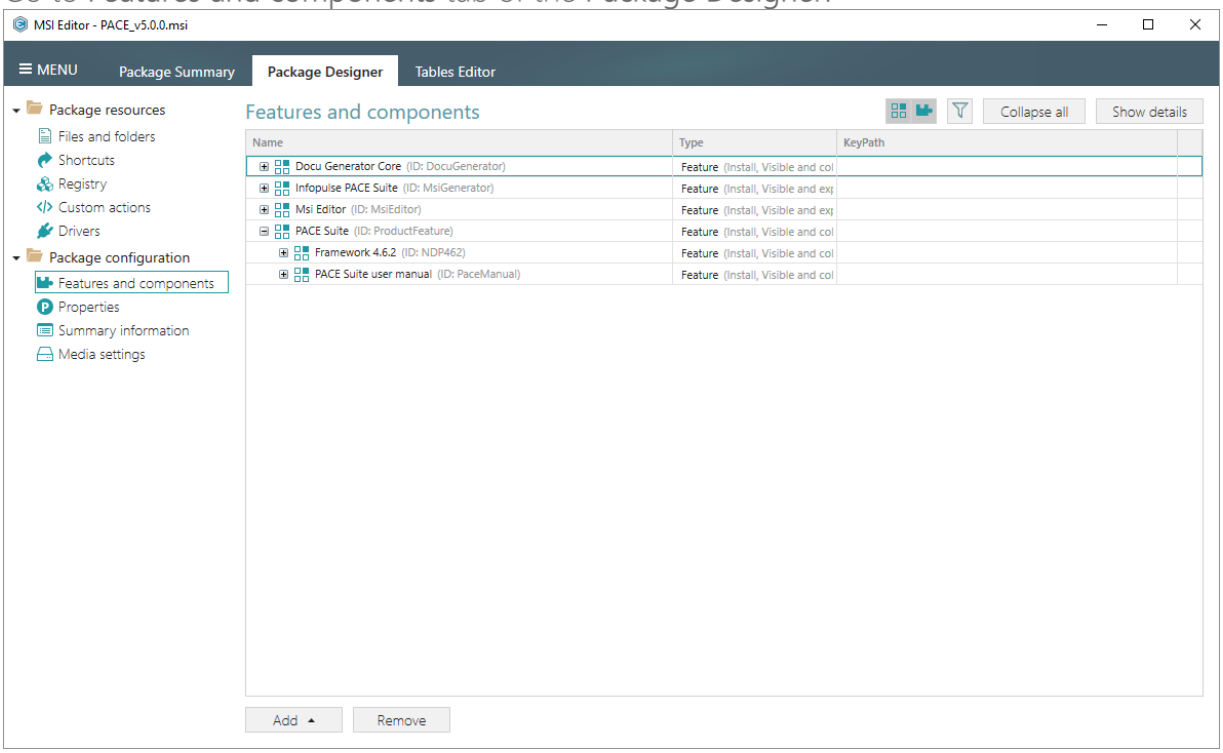
- For saving changes, select Save from MENU.




### 3.7.7.7 Review Component Resources

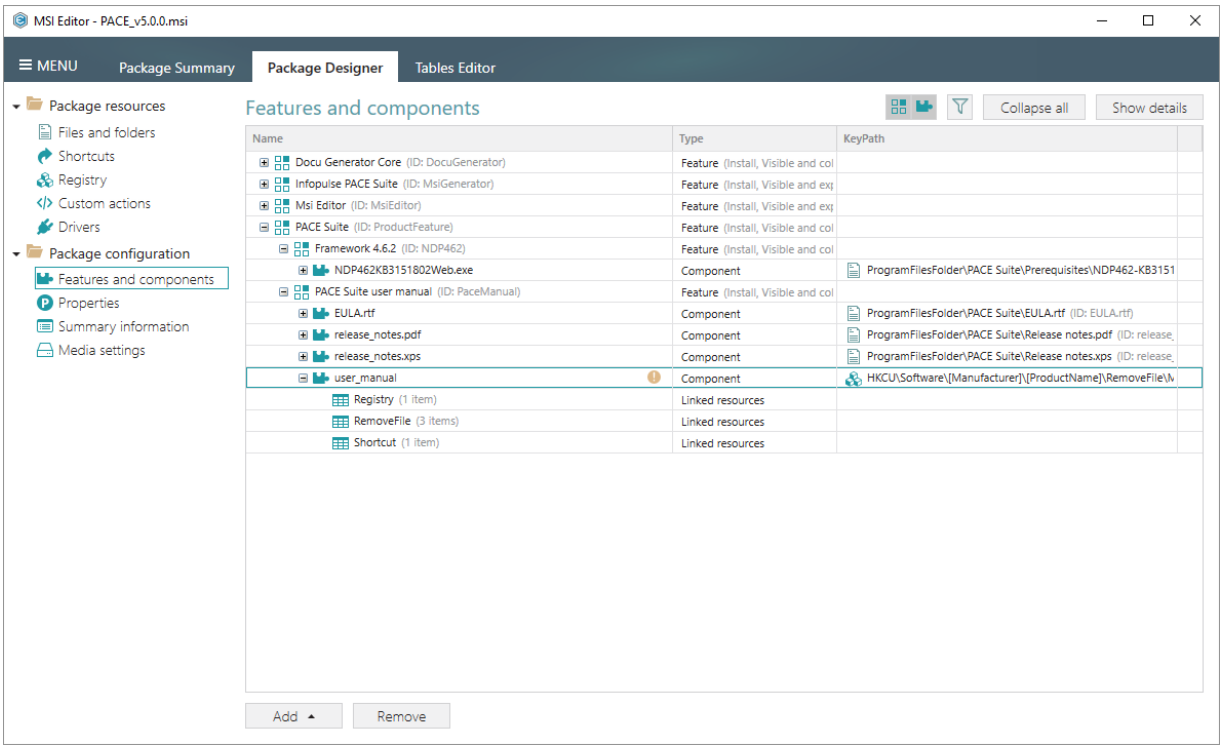
Review what resources (files, registry, shortcuts, etc.) are linked with the component.

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

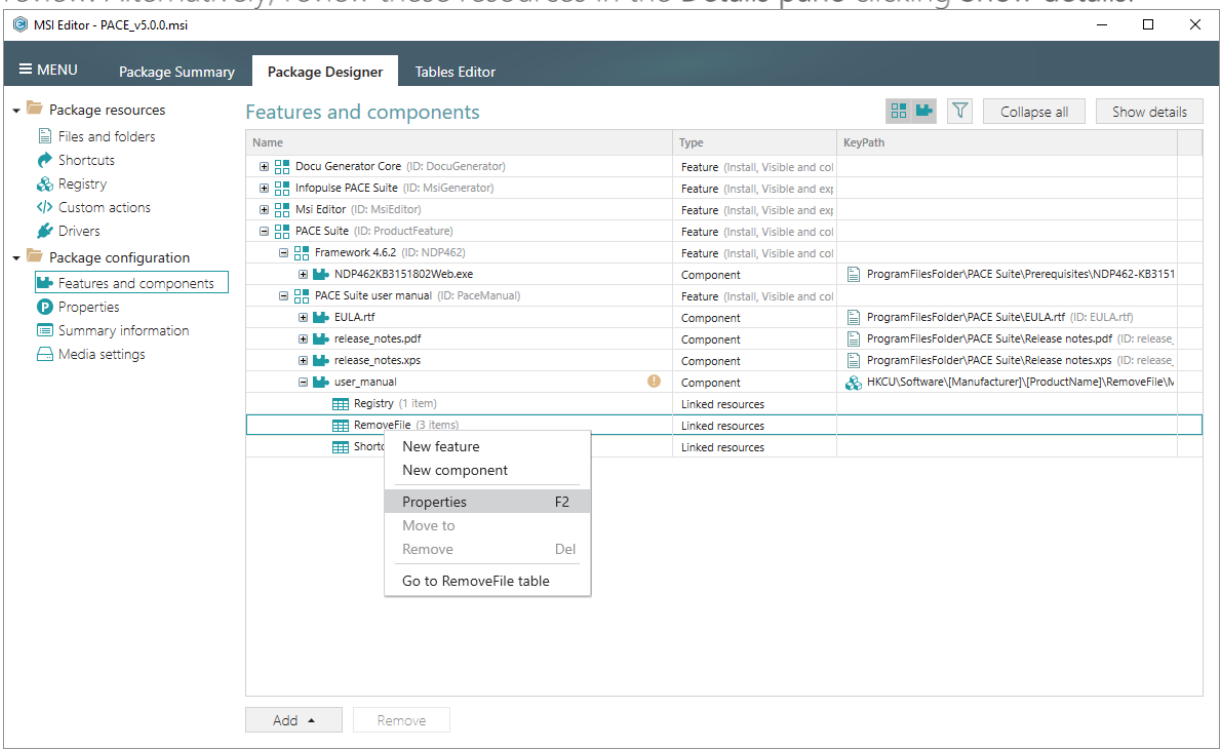



- Find the necessary component and click ⊞ to expand the component node and to see resources, linked with this component. Note that all resources are grouped by tables, from

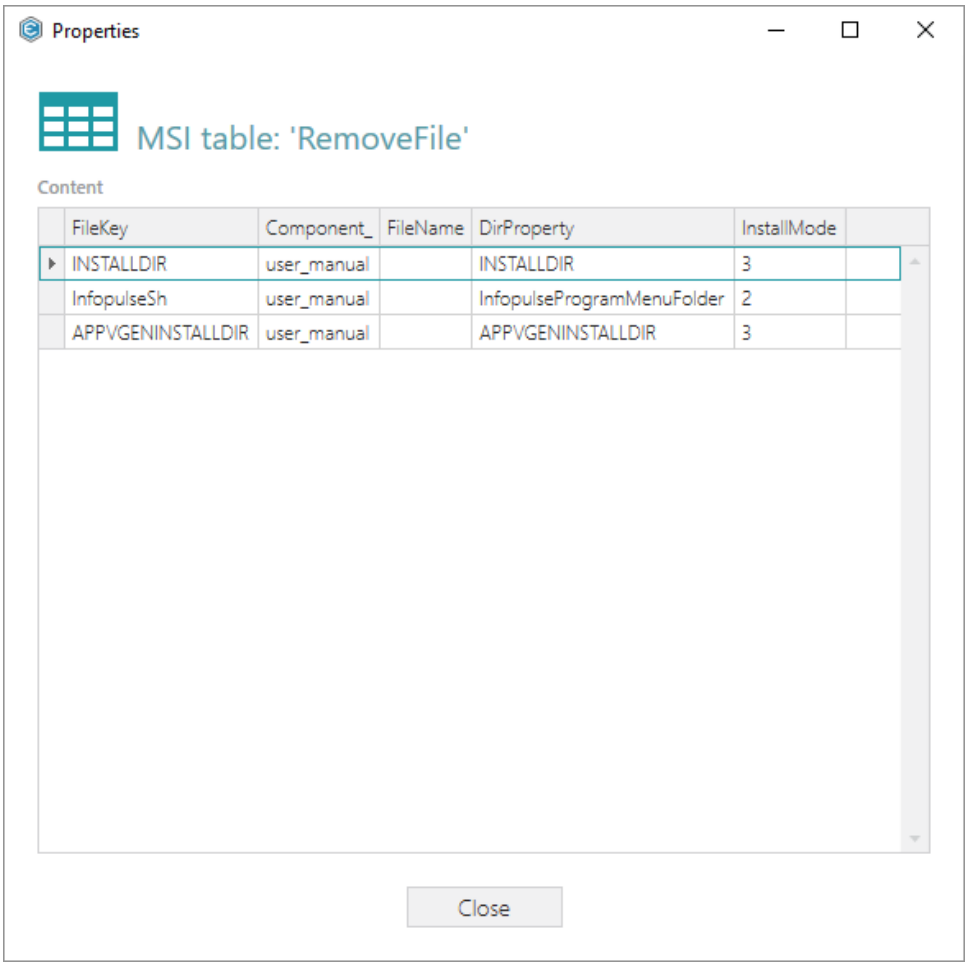
which these resources were read.




- 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.




## 3.7.8 Properties

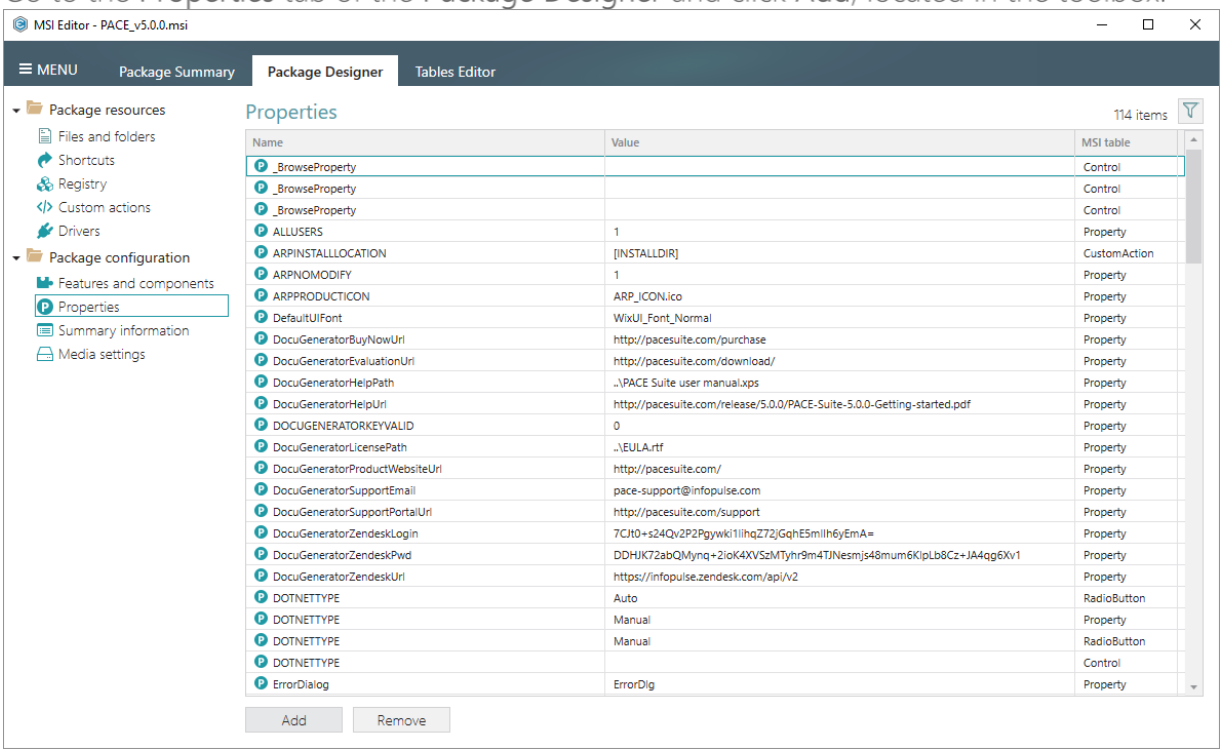
Choose a scenario that better suits your needs:

- Add Property, described in section 3.7.8.1
  Add a new property to the package.
- Edit Property, described in section 3.7.8.2
  Edit a name and a value of the existing property.

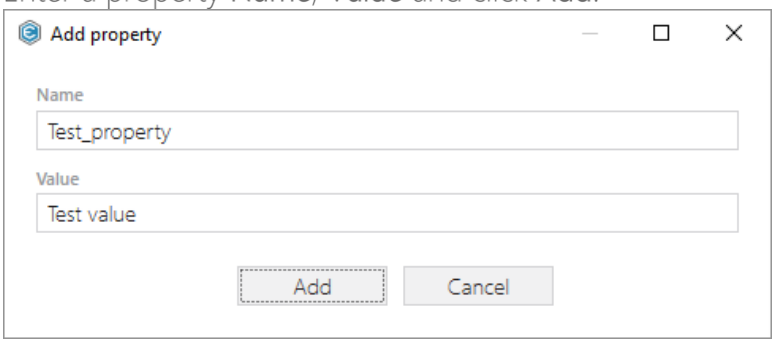
### 3.7.8.1 Add Property

Add a new property to the package.

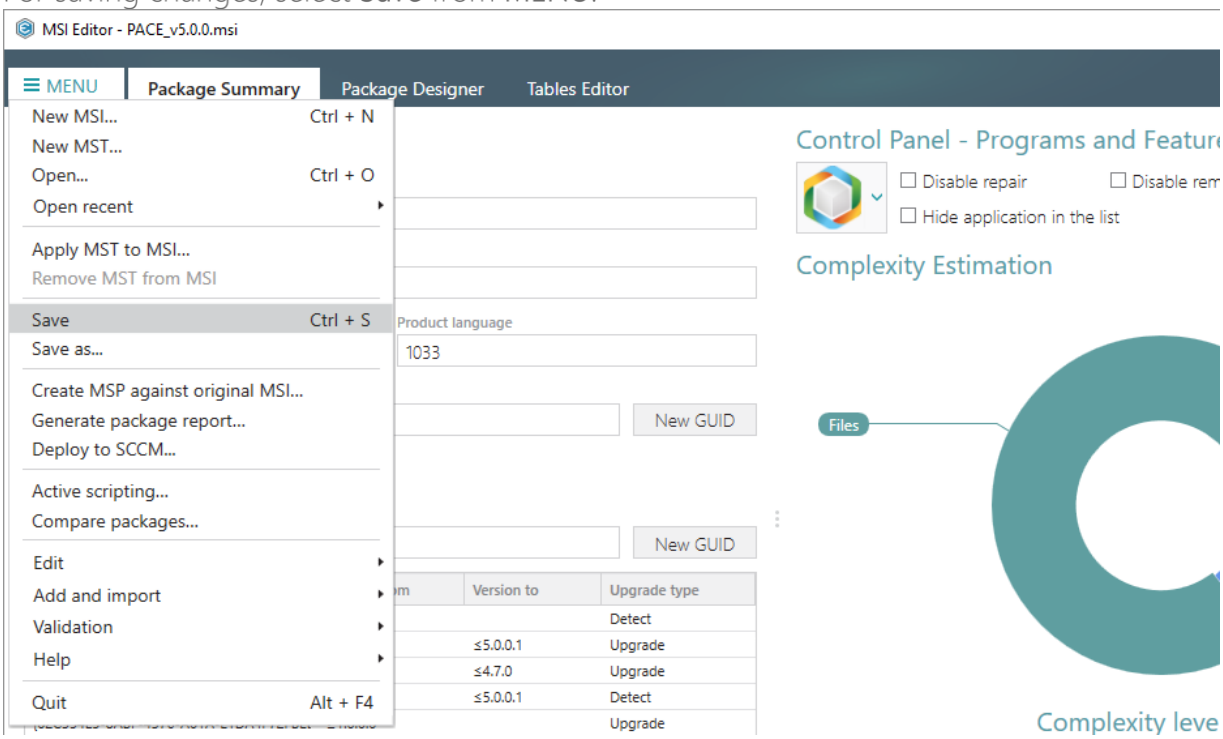
- Go to the **Properties** tab of the **Package Designer** and click **Add**, located in the toolbox.




- Enter a property **Name**, **Value** and click **Add**.



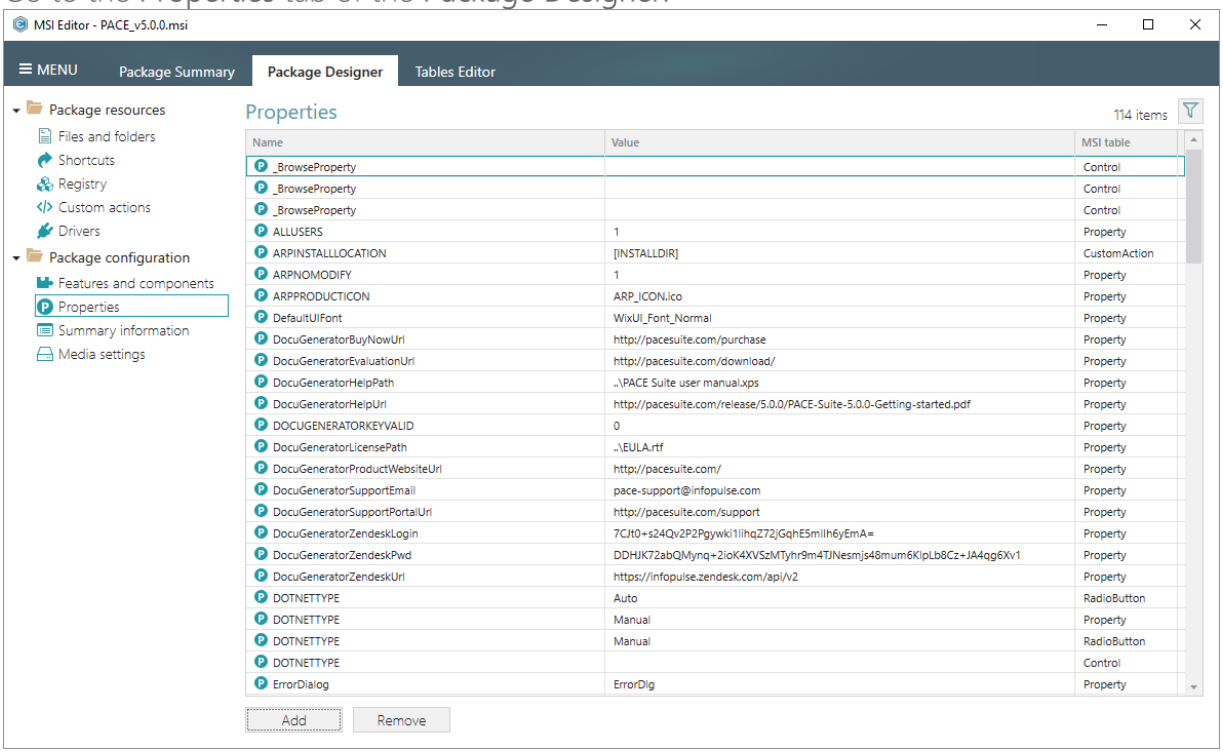
- For saving changes, select Save from MENU.



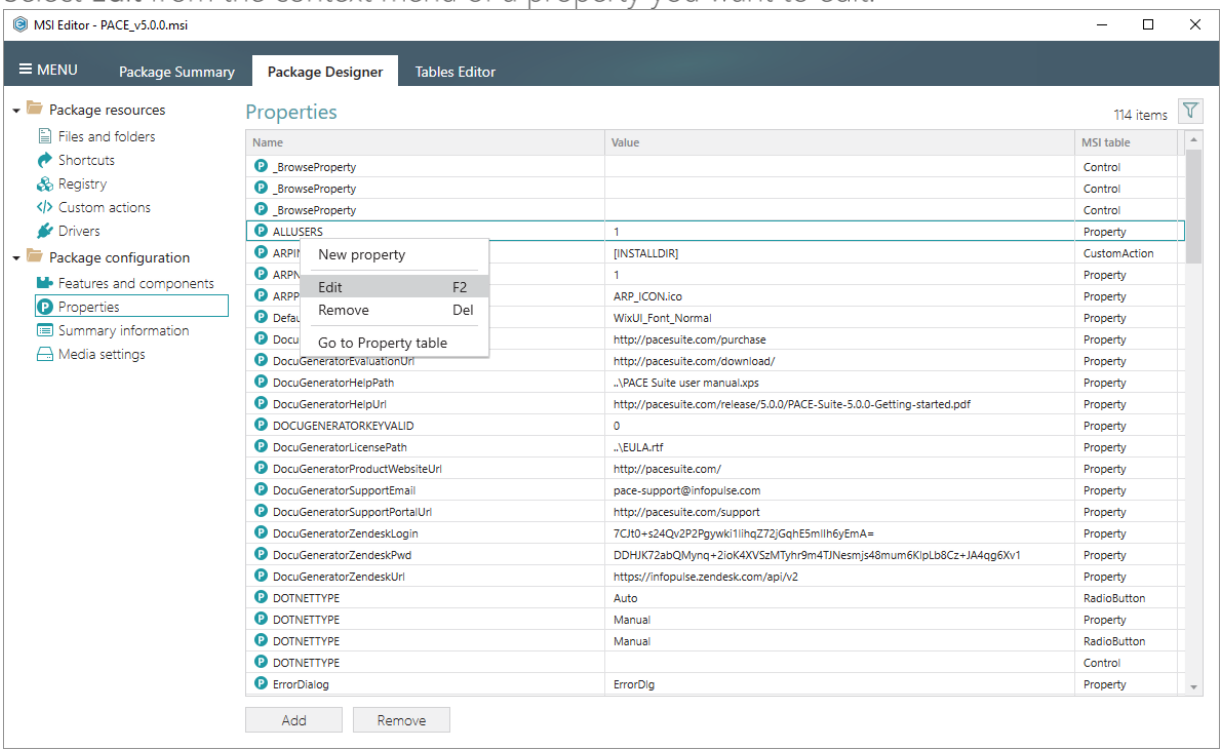

## 3.7.8.2 Edit Property

Edit a name and a value of the existing property.

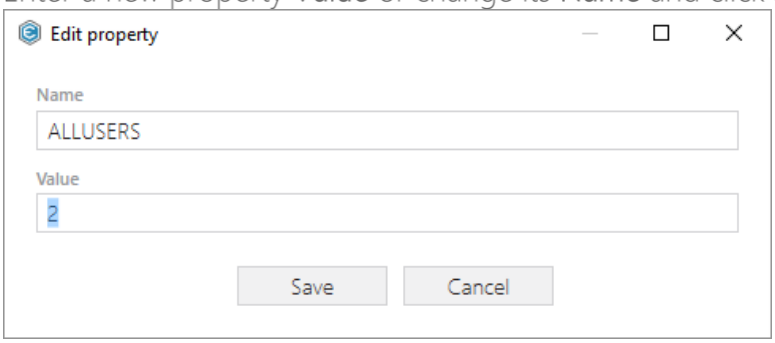
- Go to the Properties tab of the Package Designer.



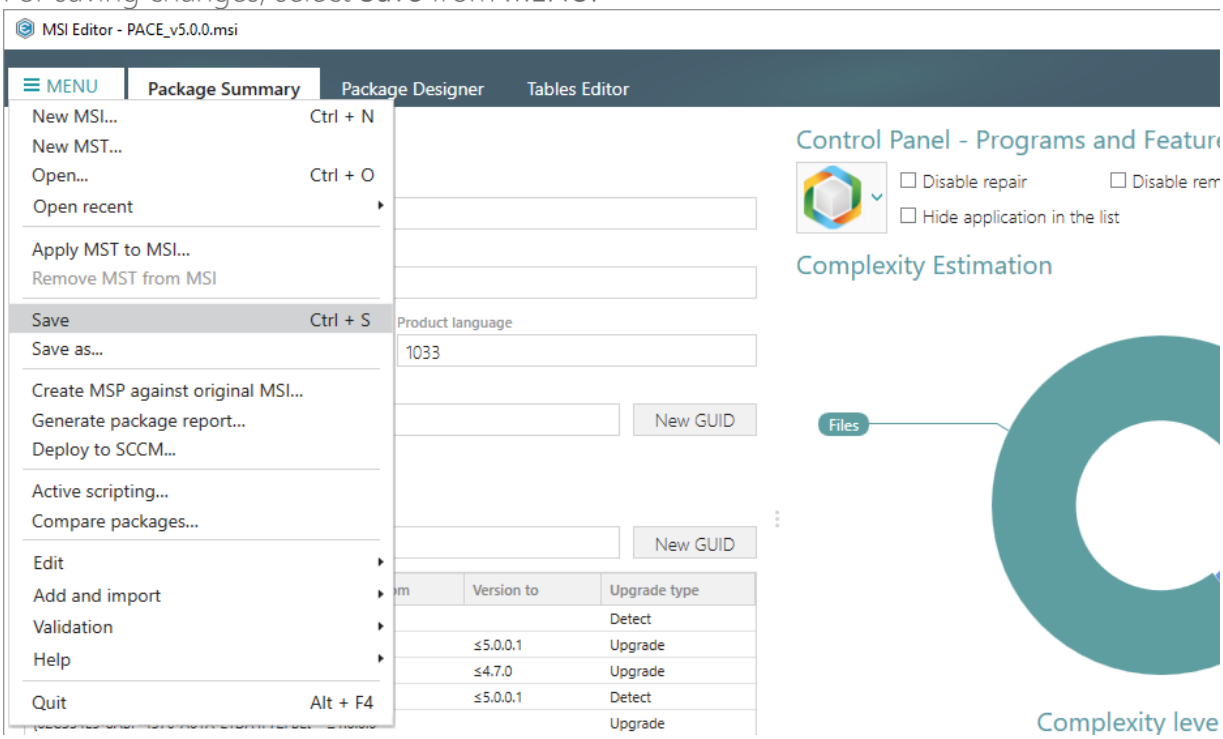
- Select **Edit** from the context menu of a property you want to edit.




- Enter a new property **Value** or change its **Name** and click **Save**.



- For saving changes, select Save from MENU.

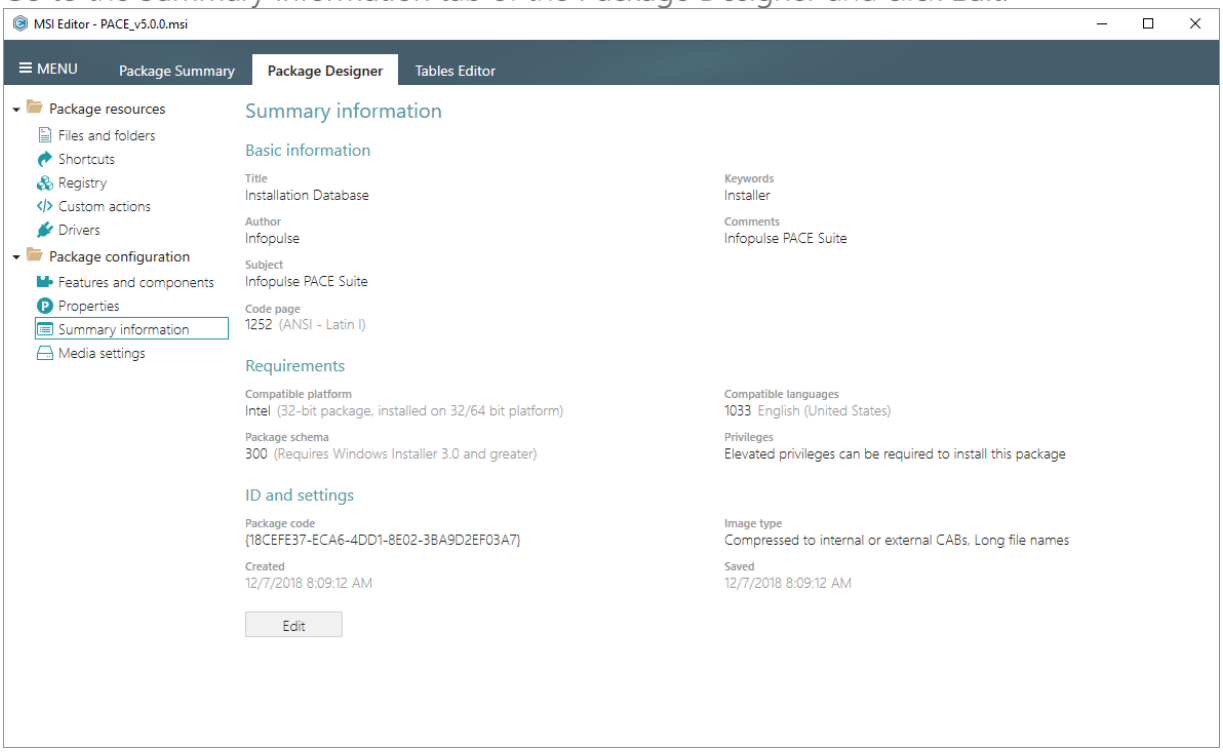



## 3.7.9 Summary Information

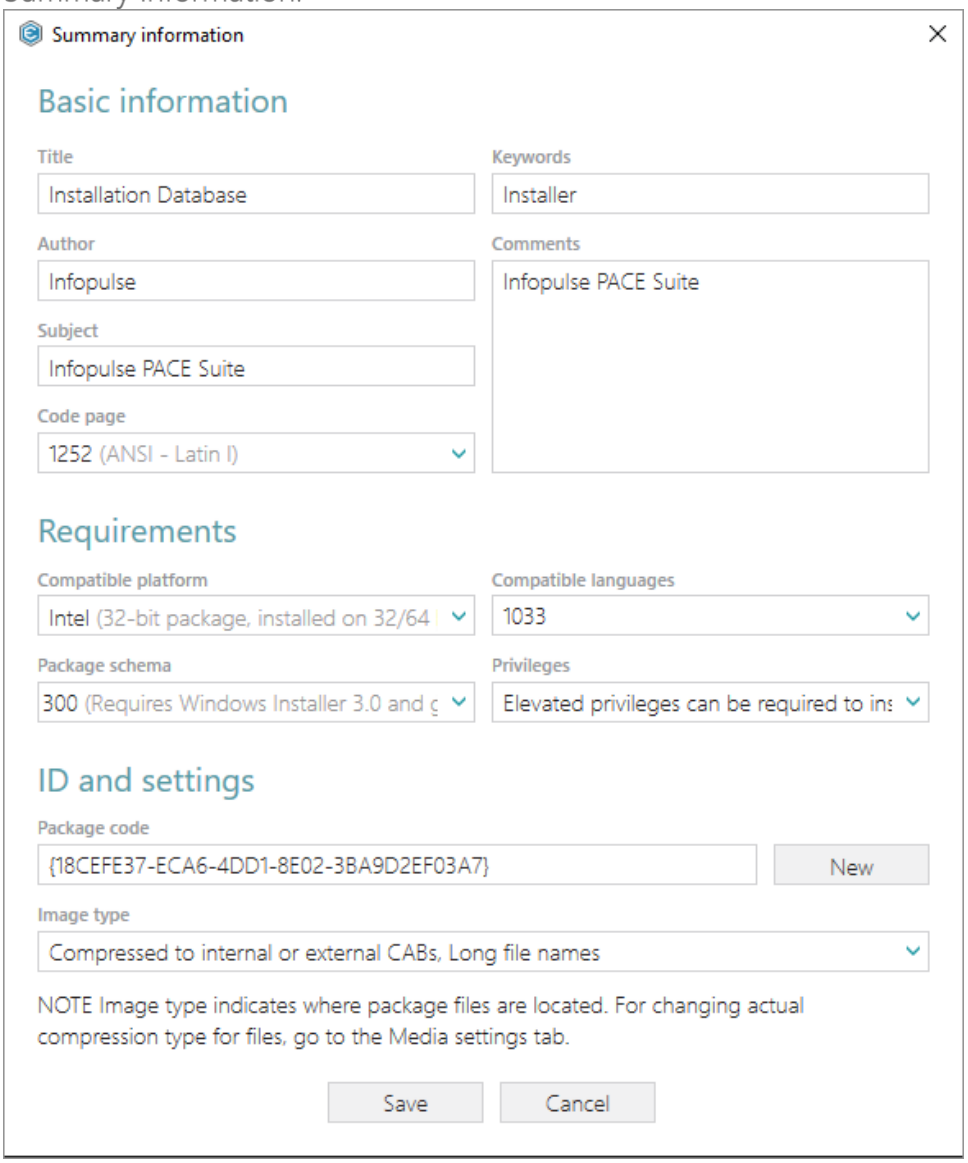
### 3.7.9.1 Edit MSI/MST Summary Information

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

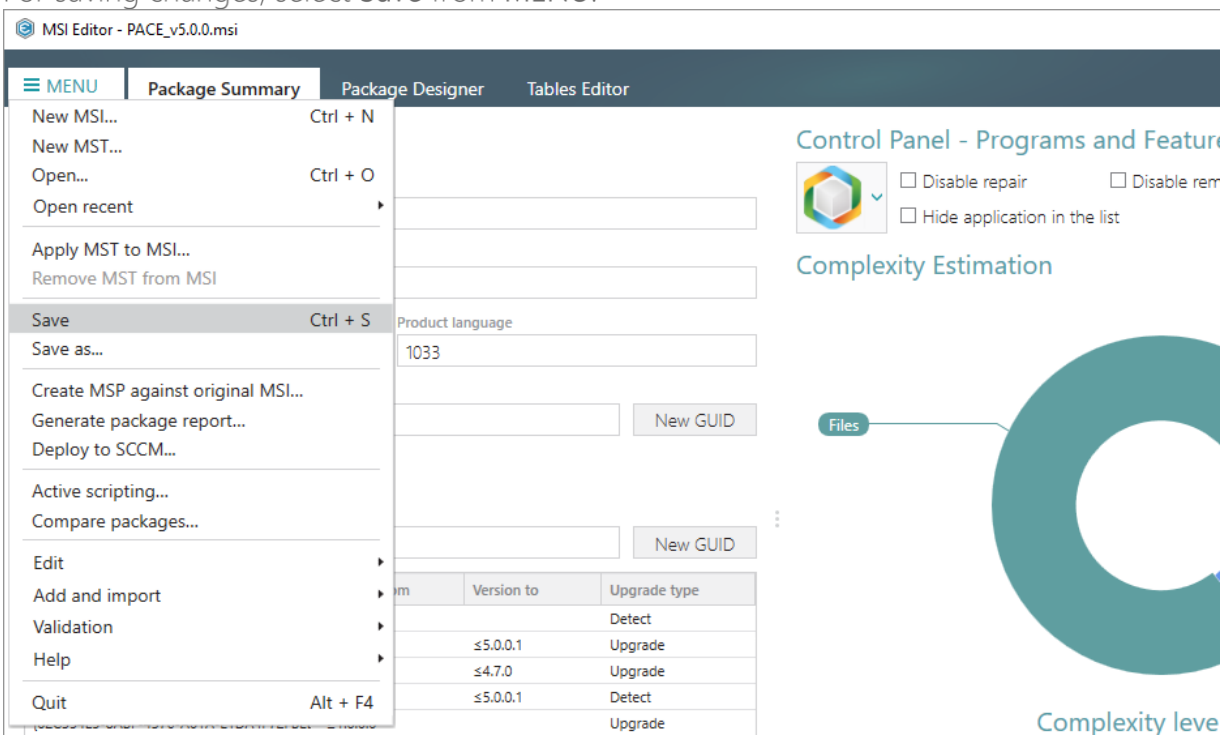
- Go to the Summary Information tab of the Package Designer and click Edit.



- Make the necessary changes to the Summary Information properties and click **Save**. Note that in order to change the files compression type, go to the **Media settings** tab of the **Package Designer**, as described in section 3.7.2.4, instead of changing **Image type** of the Summary Information.



- For saving changes, select Save from MENU.

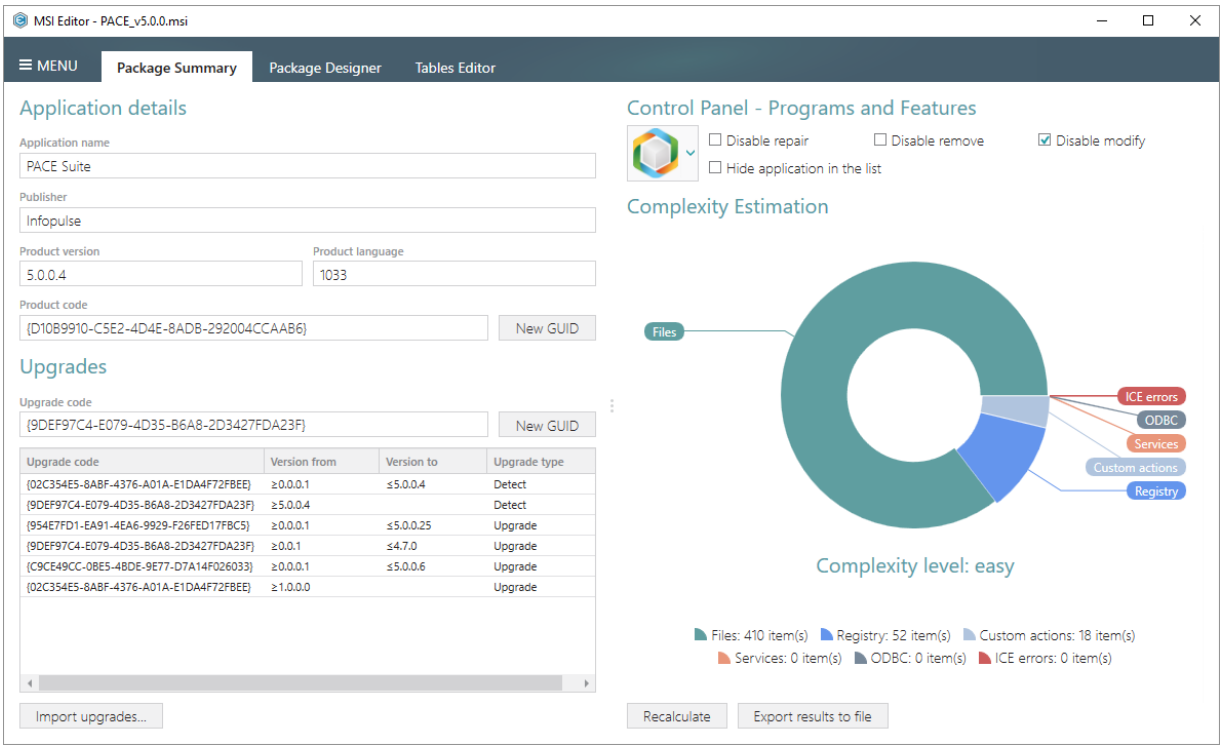



## 3.7.10 Upgrades

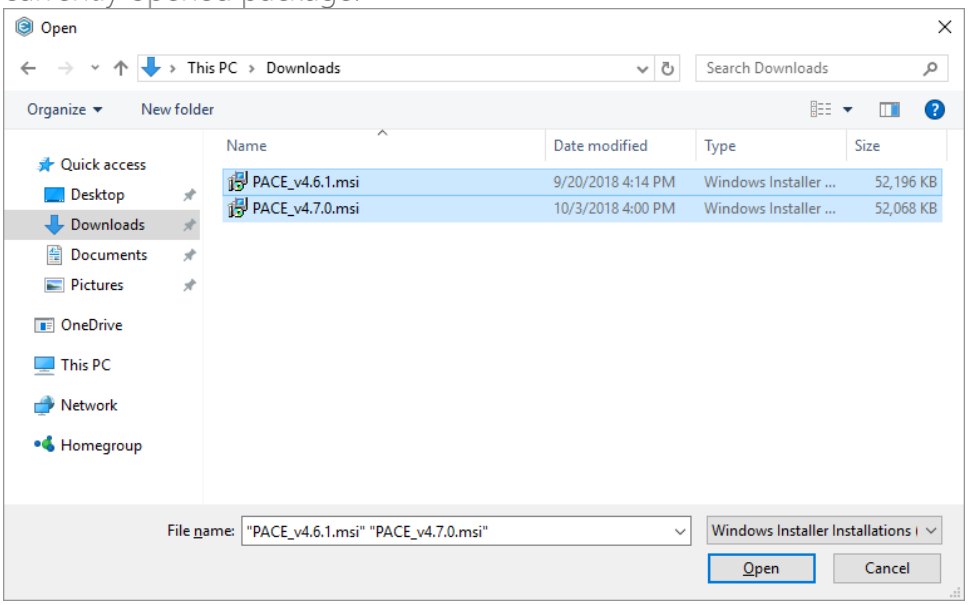
### 3.7.10.1 Integrate Upgrades

Integrate the automatic upgrade of packages of the previous versions.

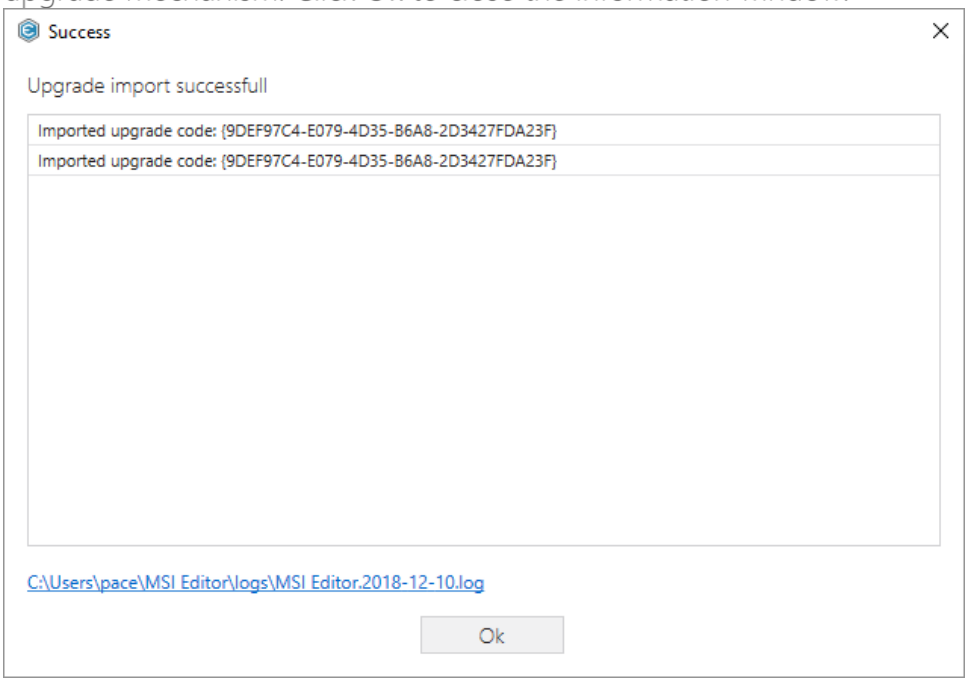
- Go to the Package Summary tab and click Import upgrades, located under the Upgrades section.



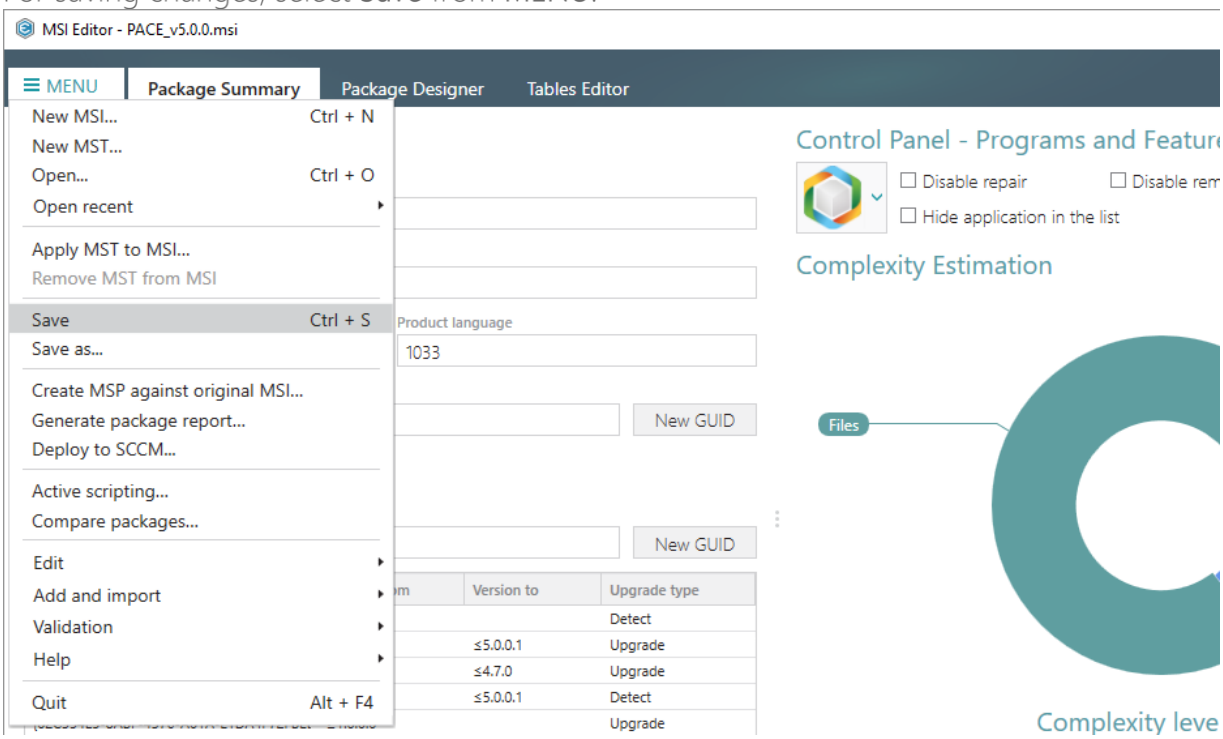
- Select MSI packages, which you want to upgrade and click Open. The upgraded packages will be automatically removed from the target system, if exist, during the installation of the currently opened package.




- The displayed window shows what upgrade codes were imported to the package. Also, MSI Editor imports product version values from the selected pacakges, adds the standard actions and the necessary informations to the opened package in order to integrate the upgrade mechanism. Click Ok to close the information window.



- For saving changes, select Save from MENU.




## 3.7.11 Tables

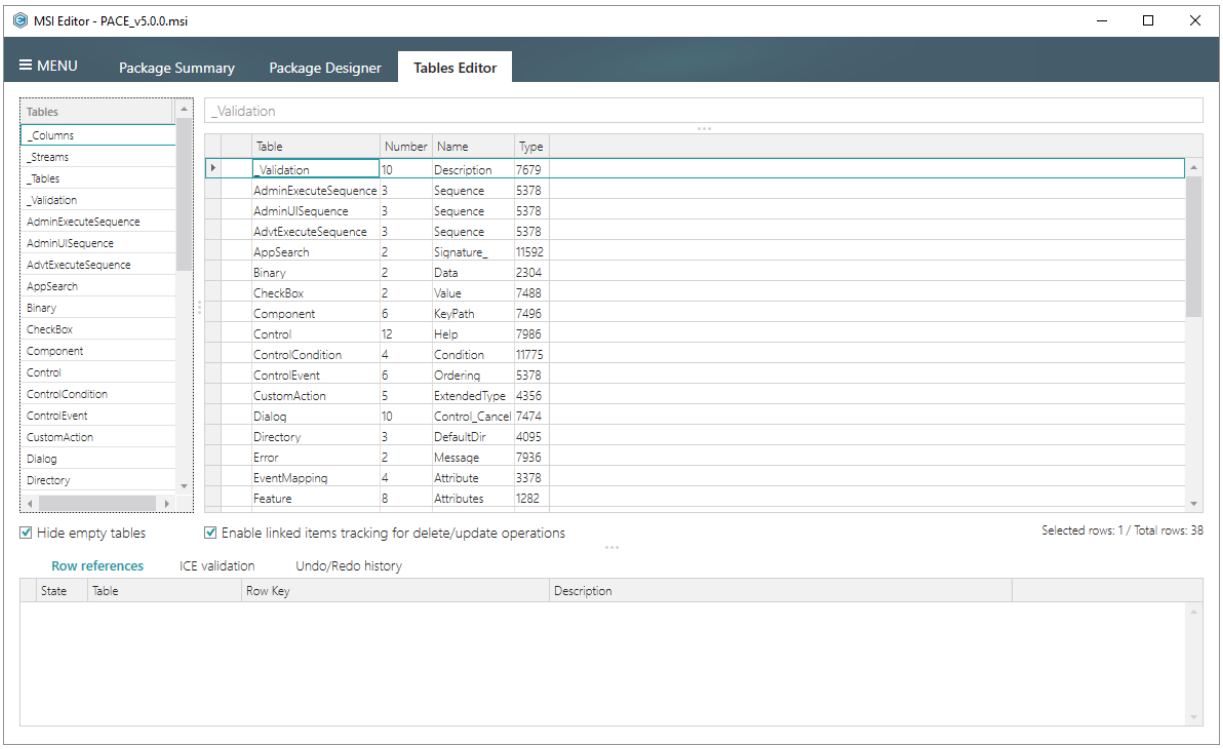
Choose a scenario that better suits your needs:

- Add Tables, described in section 3.7.11.1
  Add the standard MSI tables to the package.
- Drop Tables, described in section 3.7.11.2
  Drop tables from the package controling which of linked resources also will be removed.
- Import Tables (IDT), described in section 3.7.11.3
  Import standard or custom tables with or without content from the IDT files.

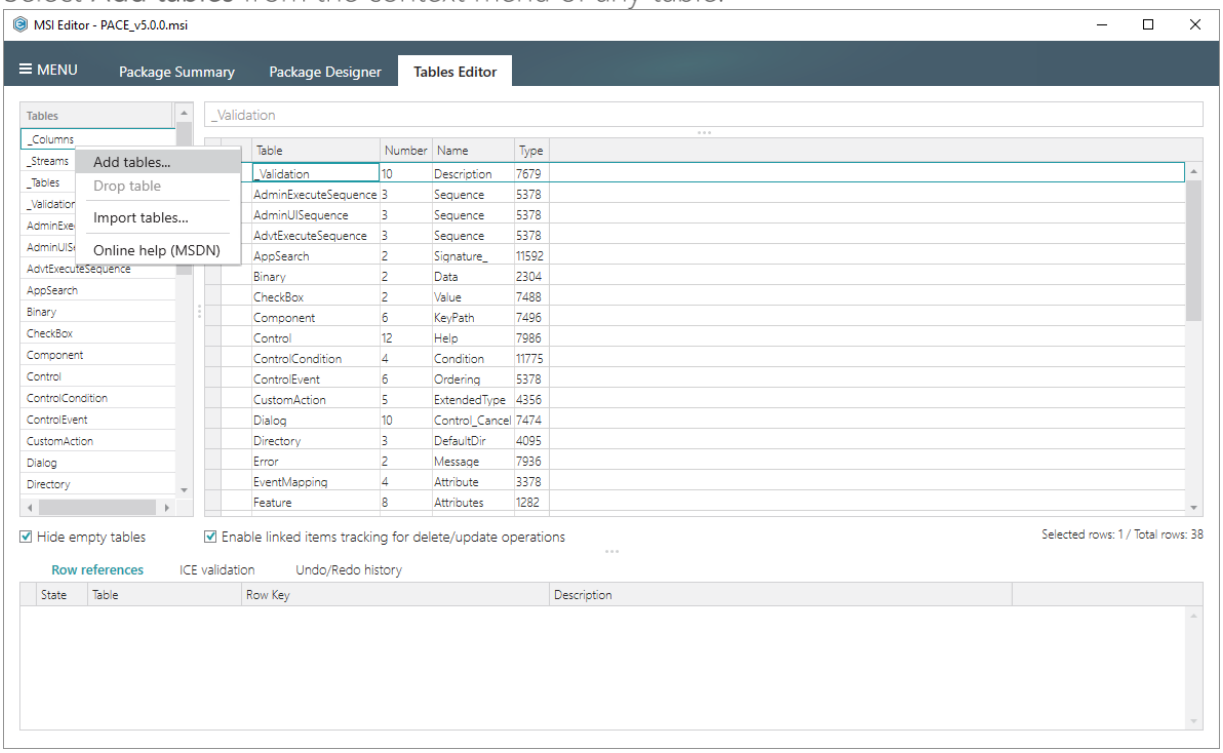
### 3.7.11.1 Add Tables

Add the standard MSI tables to the package.

- Go to the **Tables Editor** tab.

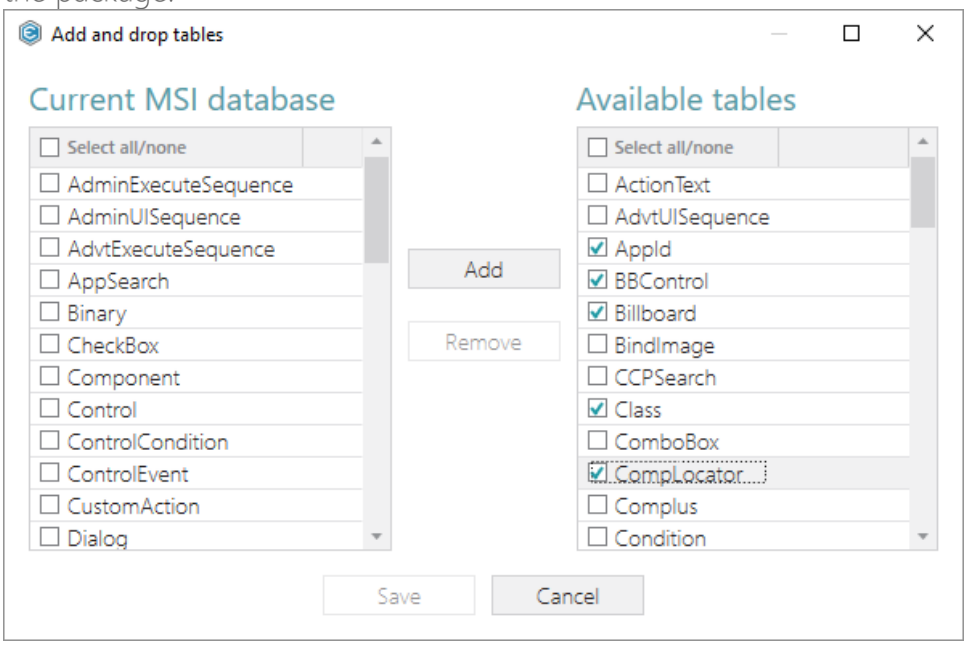



- Select **Add tables** from the context menu of any table.

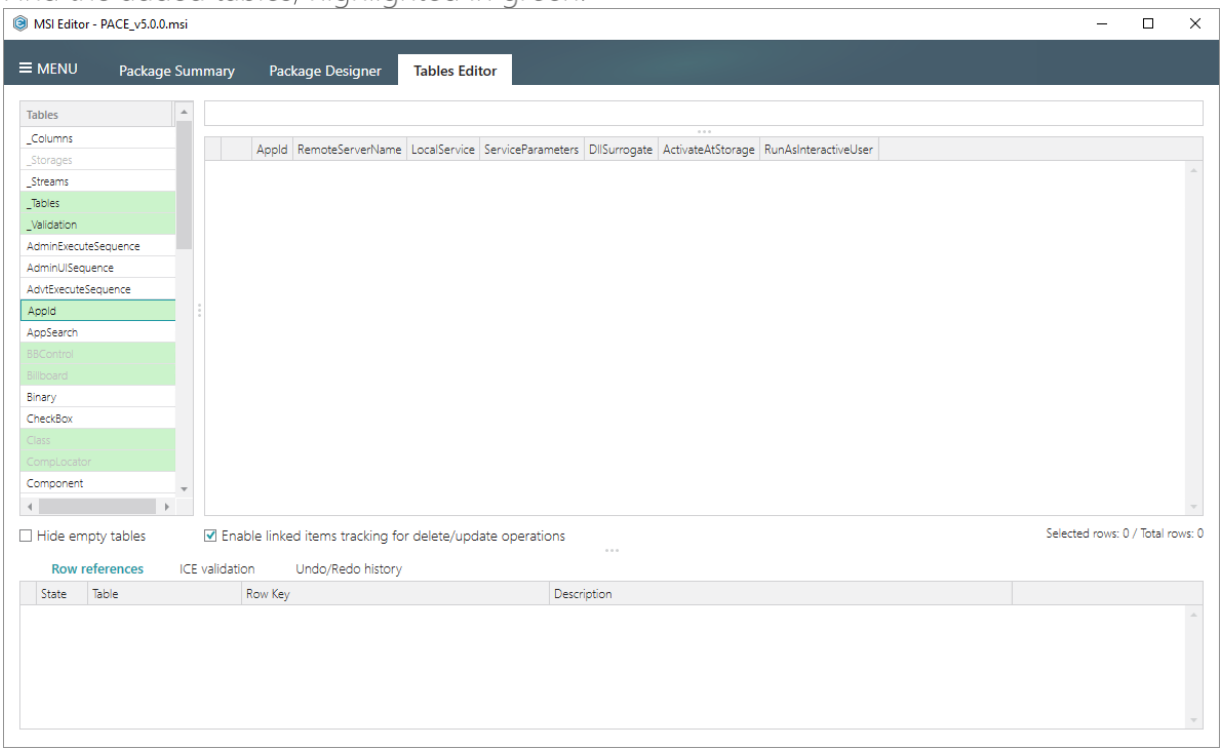



- Tick tables from the **Available tables** list, which you want to add to the package, then click **Add** and **Save**. Note that the list diplays only those standard tables, which do not exist in

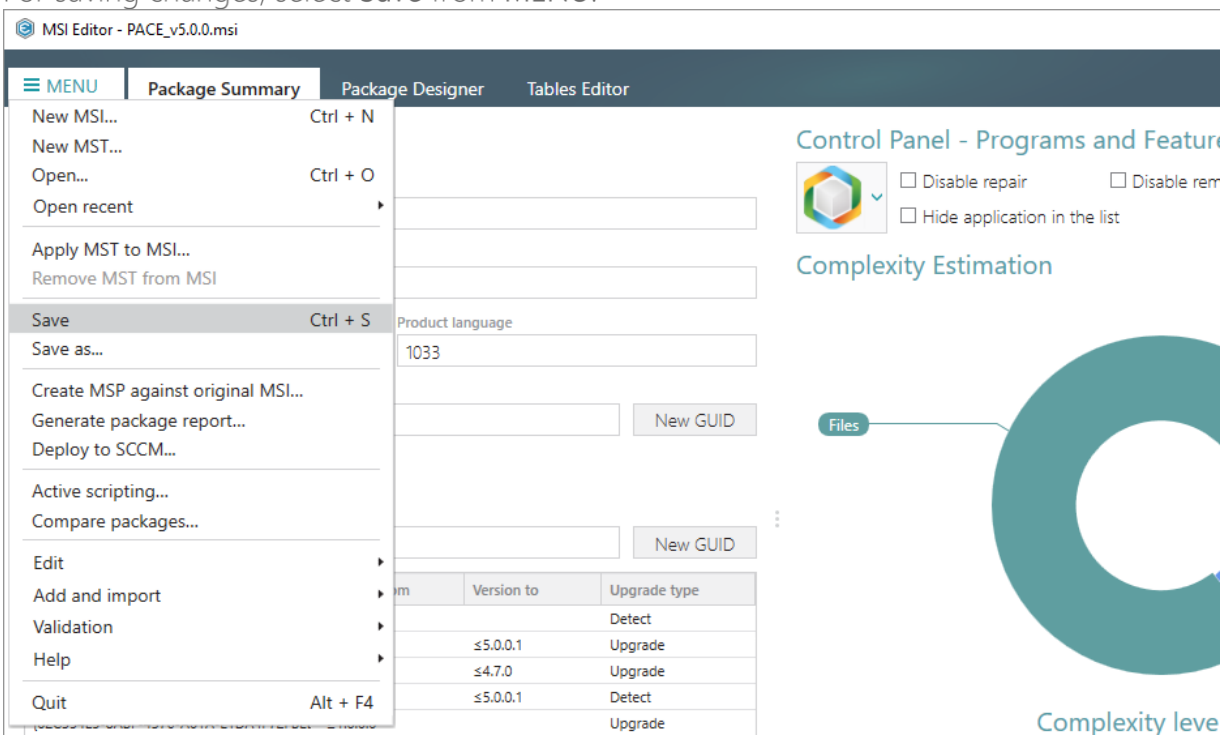
the package.




- Find the added tables, highlighted in green.



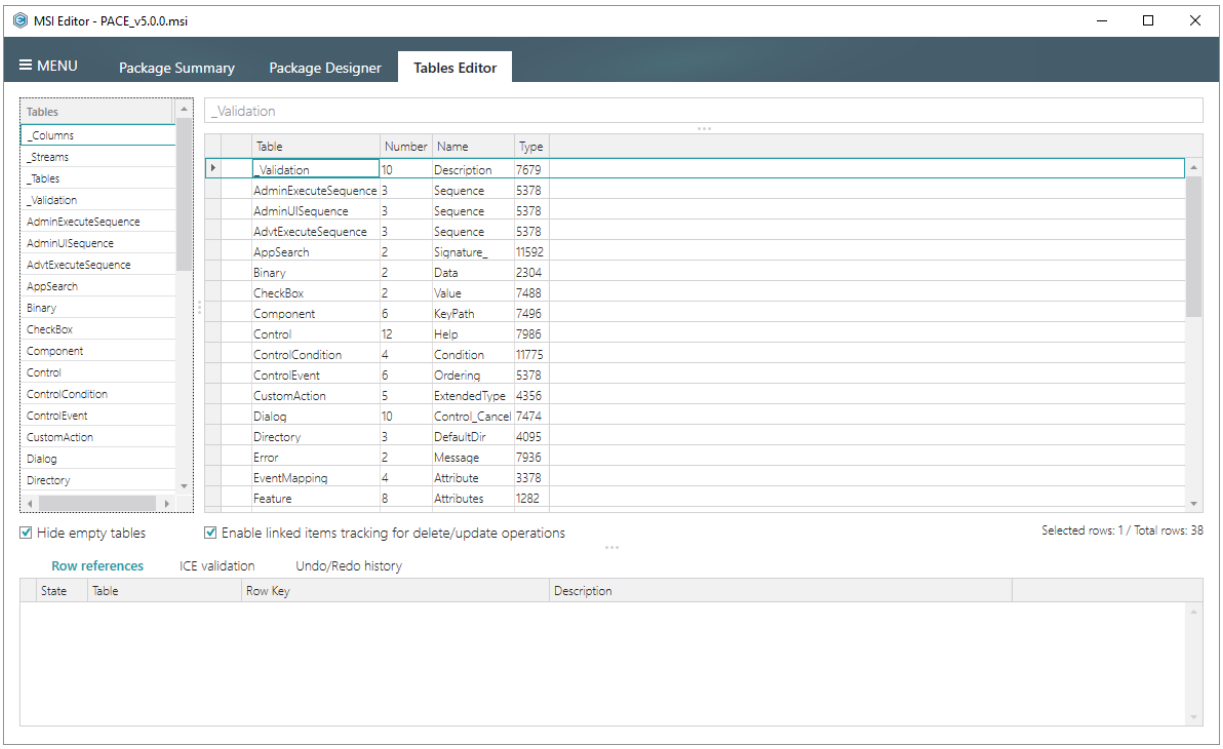
- For saving changes, select **Save** from **MENU**.



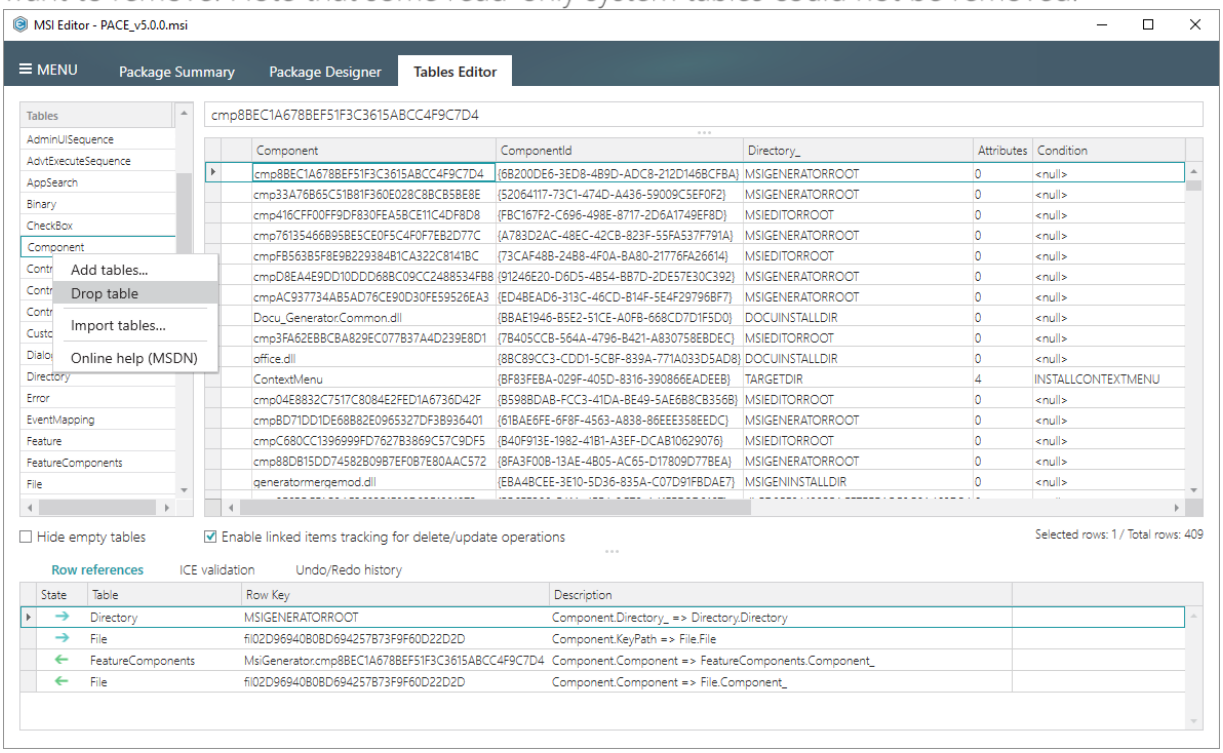

### 3.7.11.2 Drop Tables

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

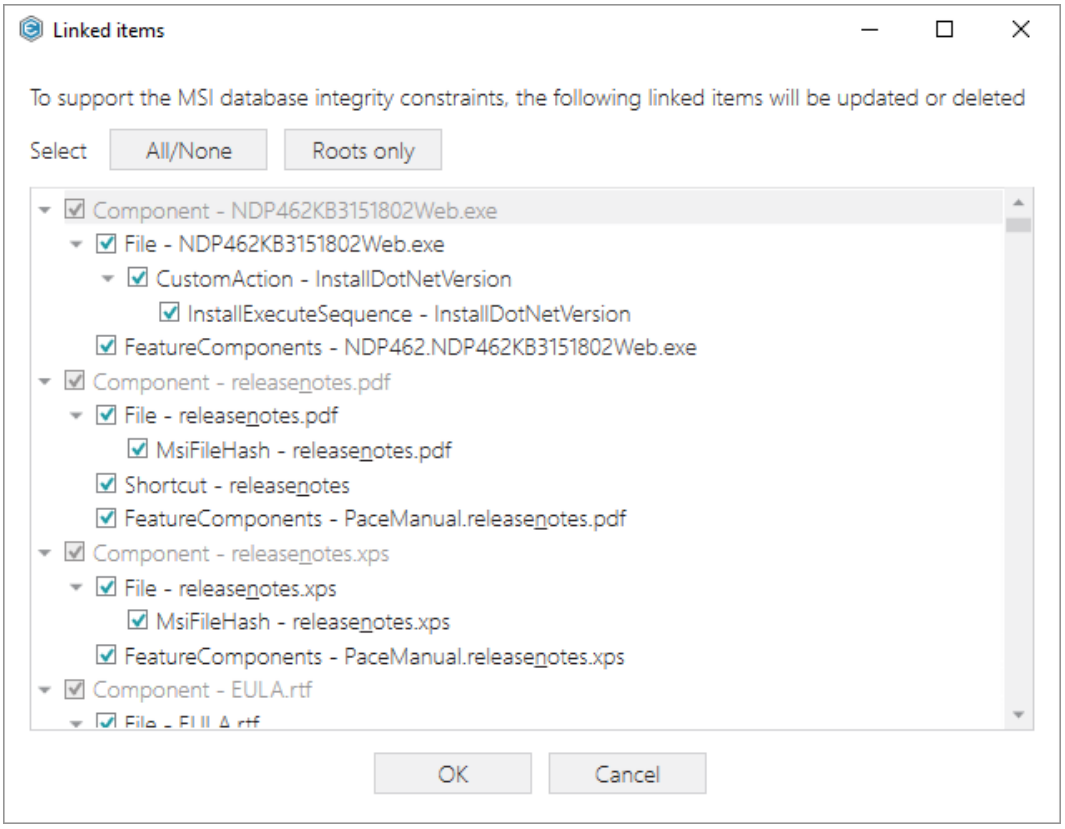
- Go to the **Tables Editor** tab.



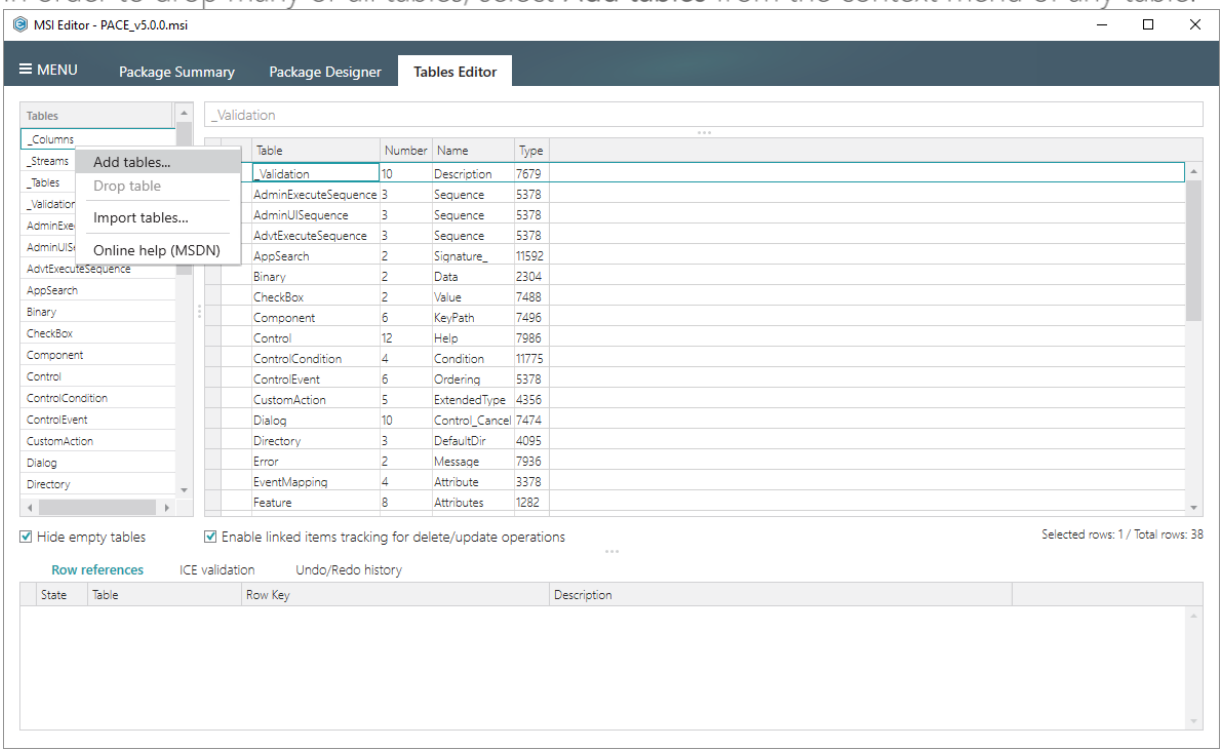
- In order to drop one or few tables, select Drop table from the context menu of tables you want to remove. Note that some read-only system tables could not be removed.



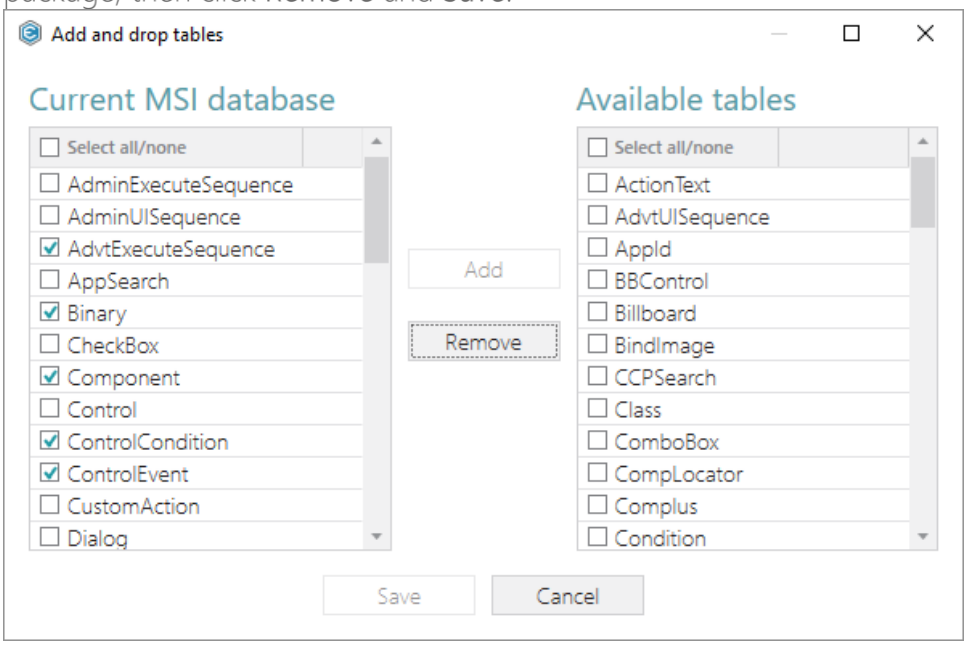

- If another MSI tables have row references to content of the dropping table, the Linked items window will display these references. The ticked items will be automatically removed from the package along with the dropping table in order to support the MSI database integrity constraints. You can untick those items, which do not want to remove. Click OK to continue.



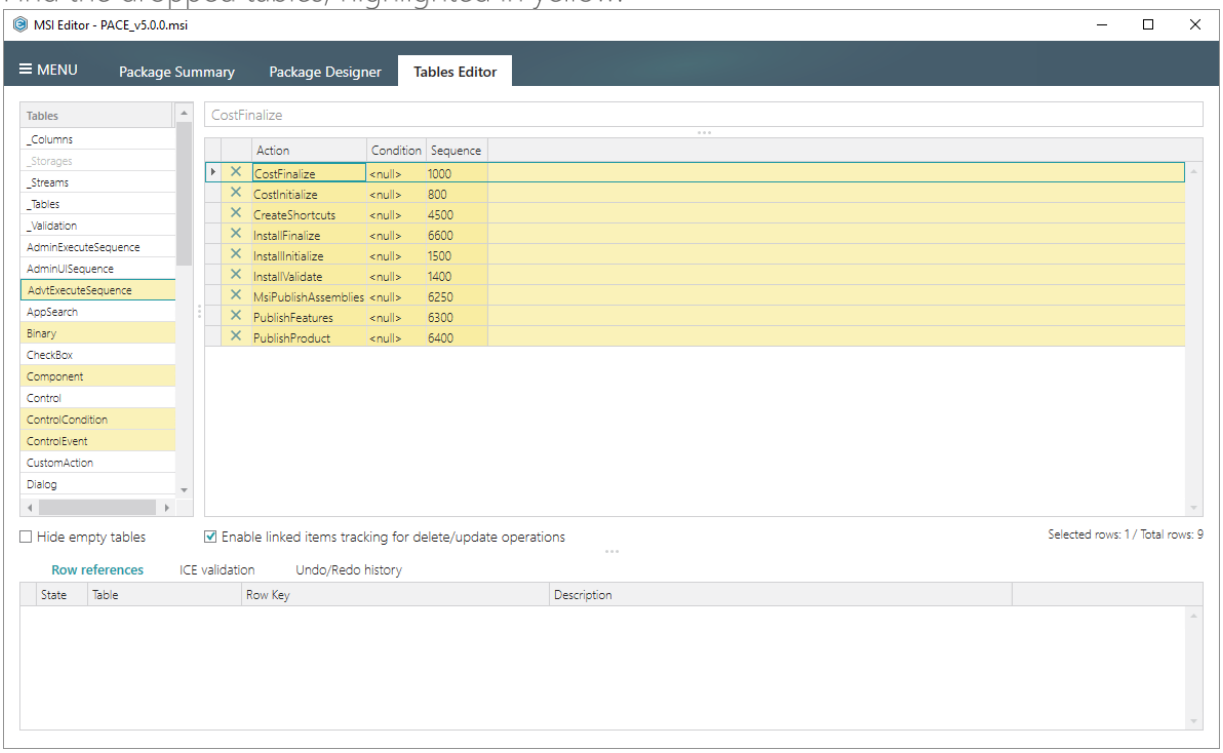
- In order to drop many or all tables, select **Add tables** from the context menu of any table.



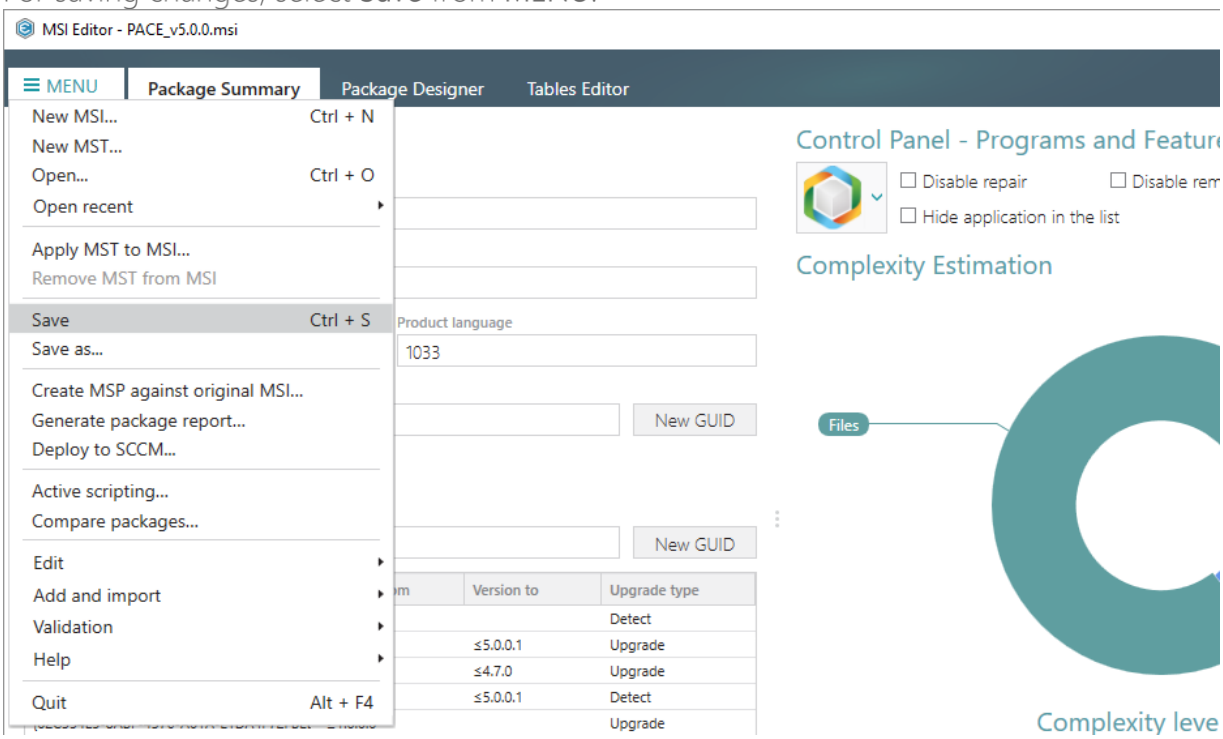

- Tick tables from the **Current MSI database** list, which you want to remove from the package, then click **Remove** and **Save**.



- Find the dropped tables, highlighted in yellow.



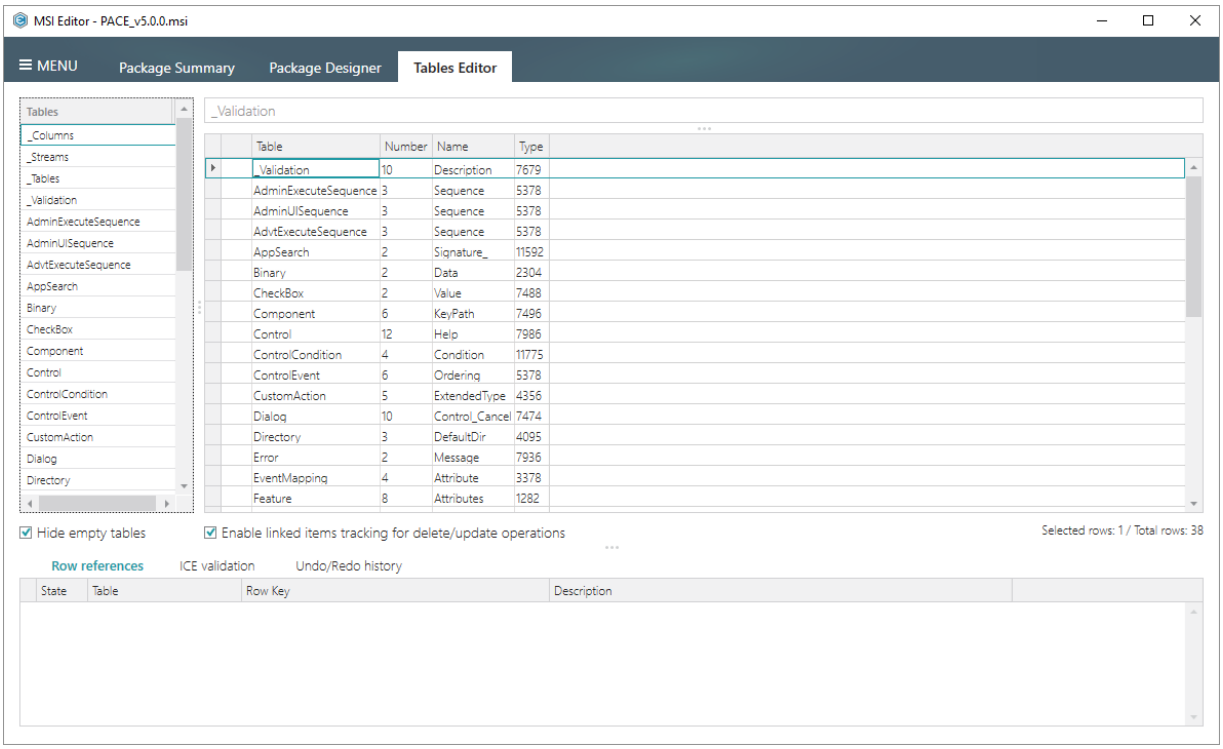

- For saving changes, select Save from MENU.



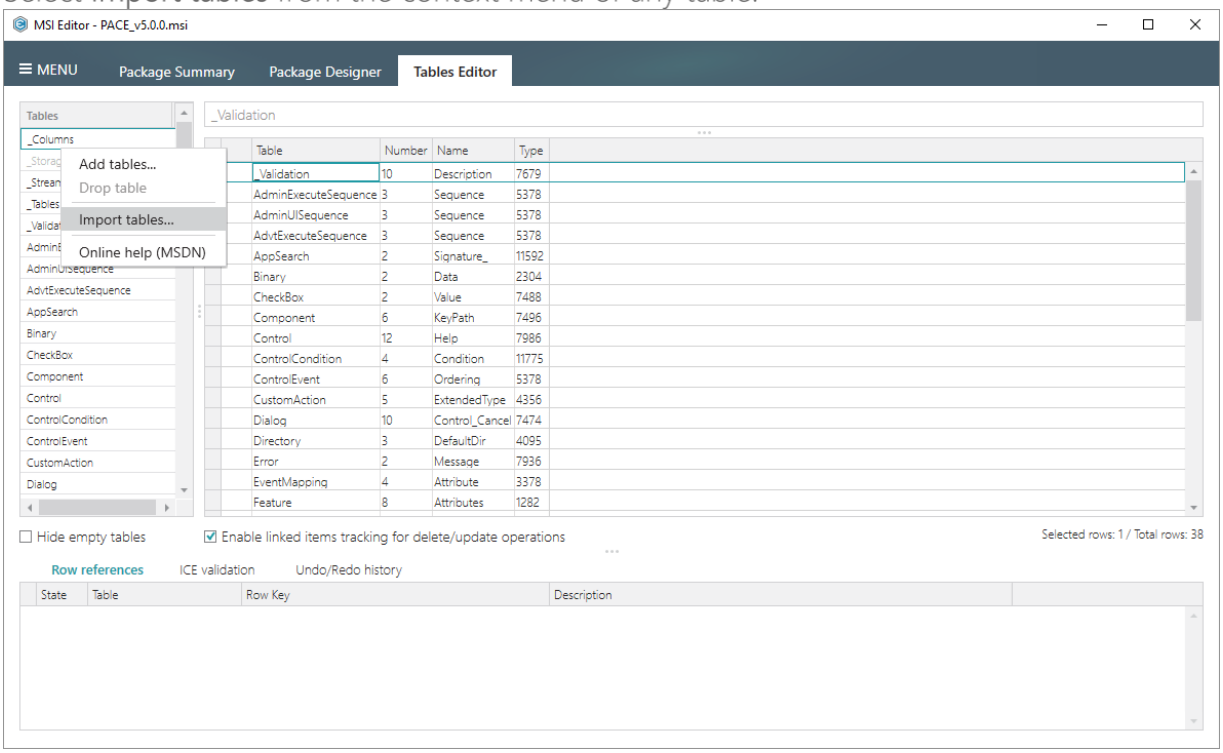
### 3.7.11.3 Import Tables (IDT)

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

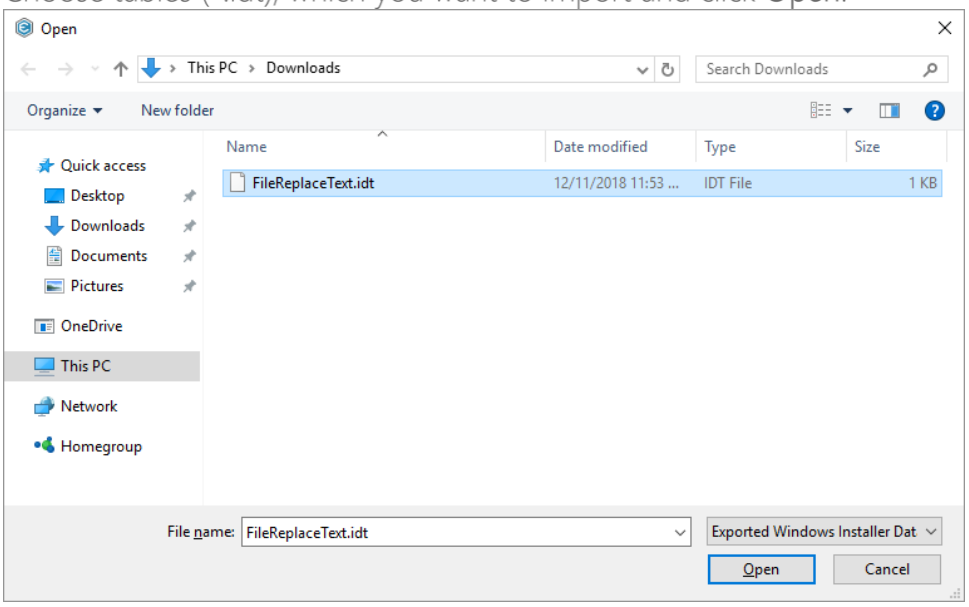
- Go to the **Tables Editor** tab.



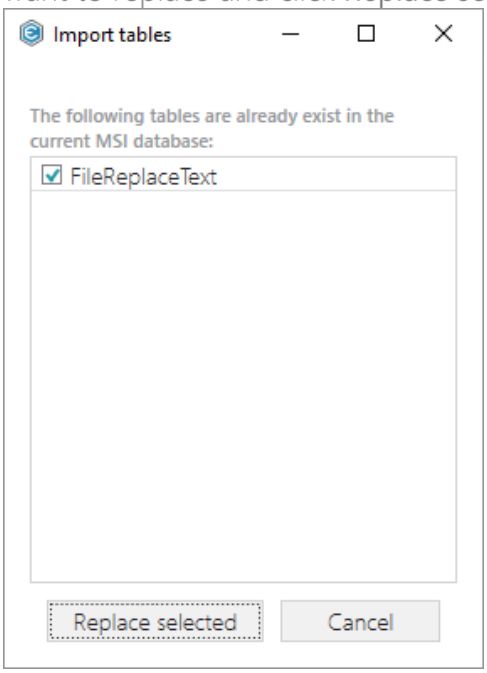

- Select **Import tables** from the context menu of any table.



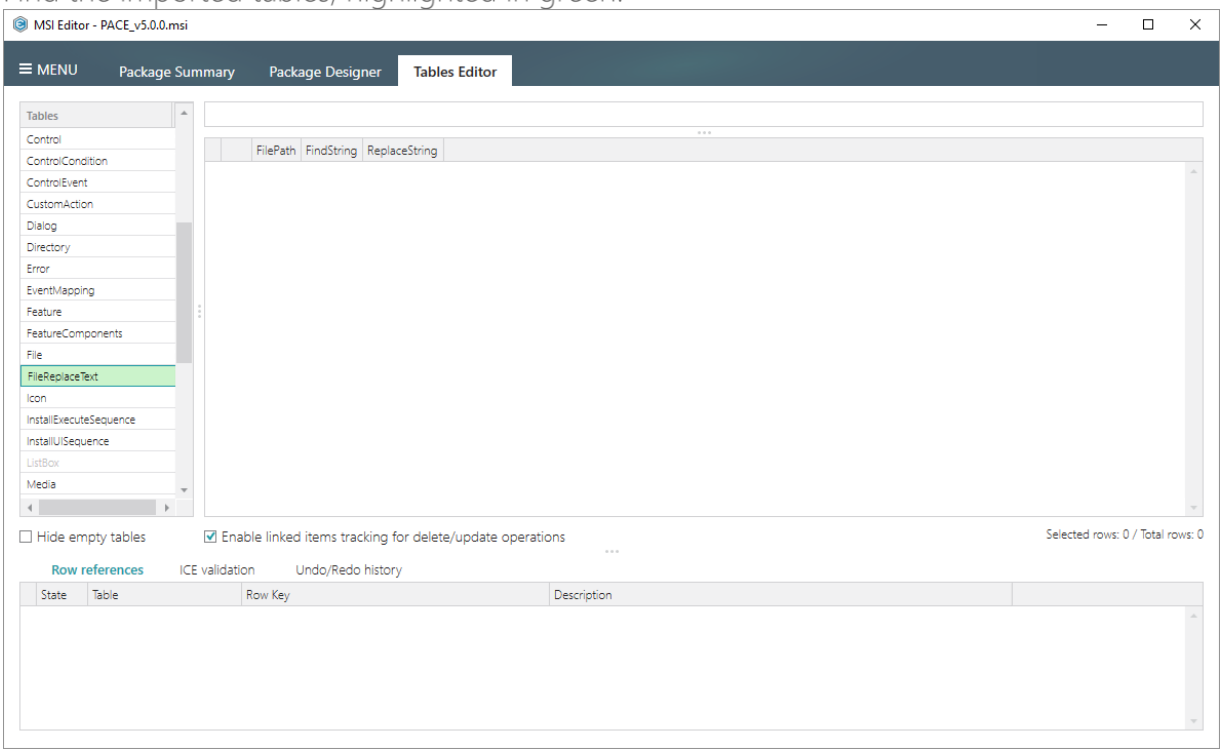
- Choose tables (*.idt), which you want to import and click Open.



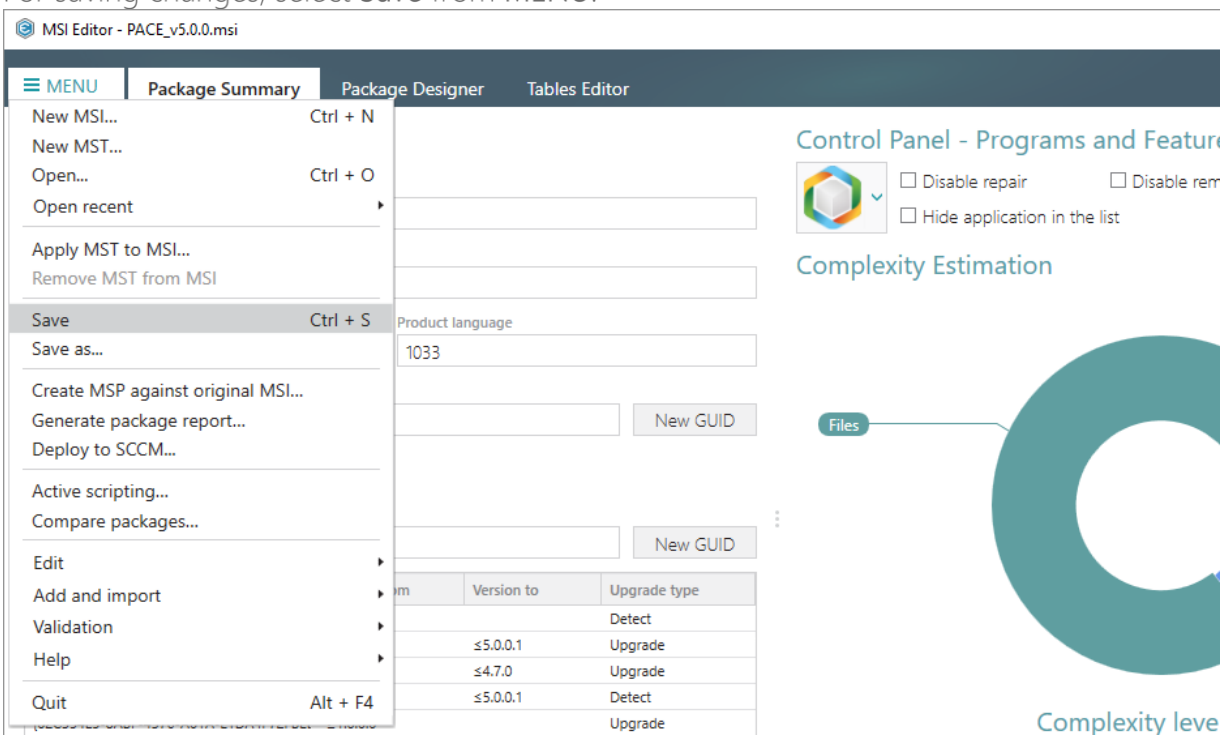

- If the package already contains a table with the same name, you will be informed about this by the Import tables window. To replace the existing tables, tick tables from the list you want to replace and click Replace selected.



- Find the imported tables, highlighted in green.




- For saving changes, select Save from MENU.

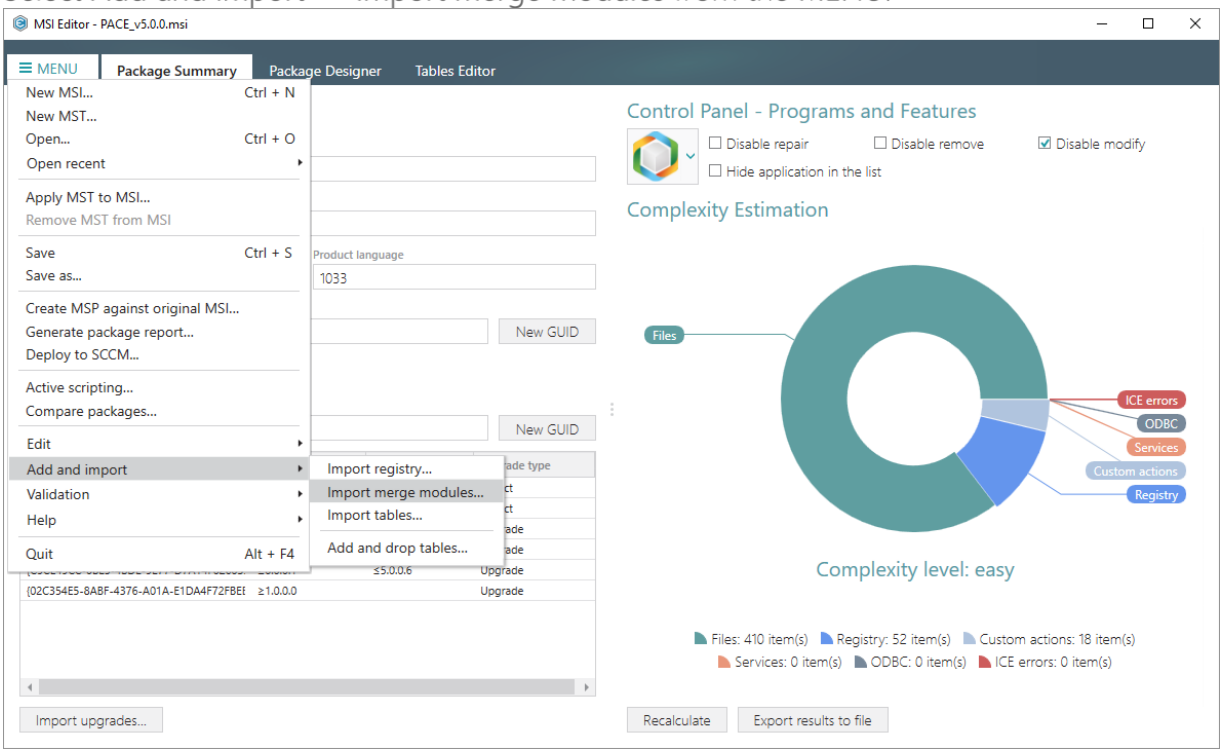


# 3.7.12 Merge Modules

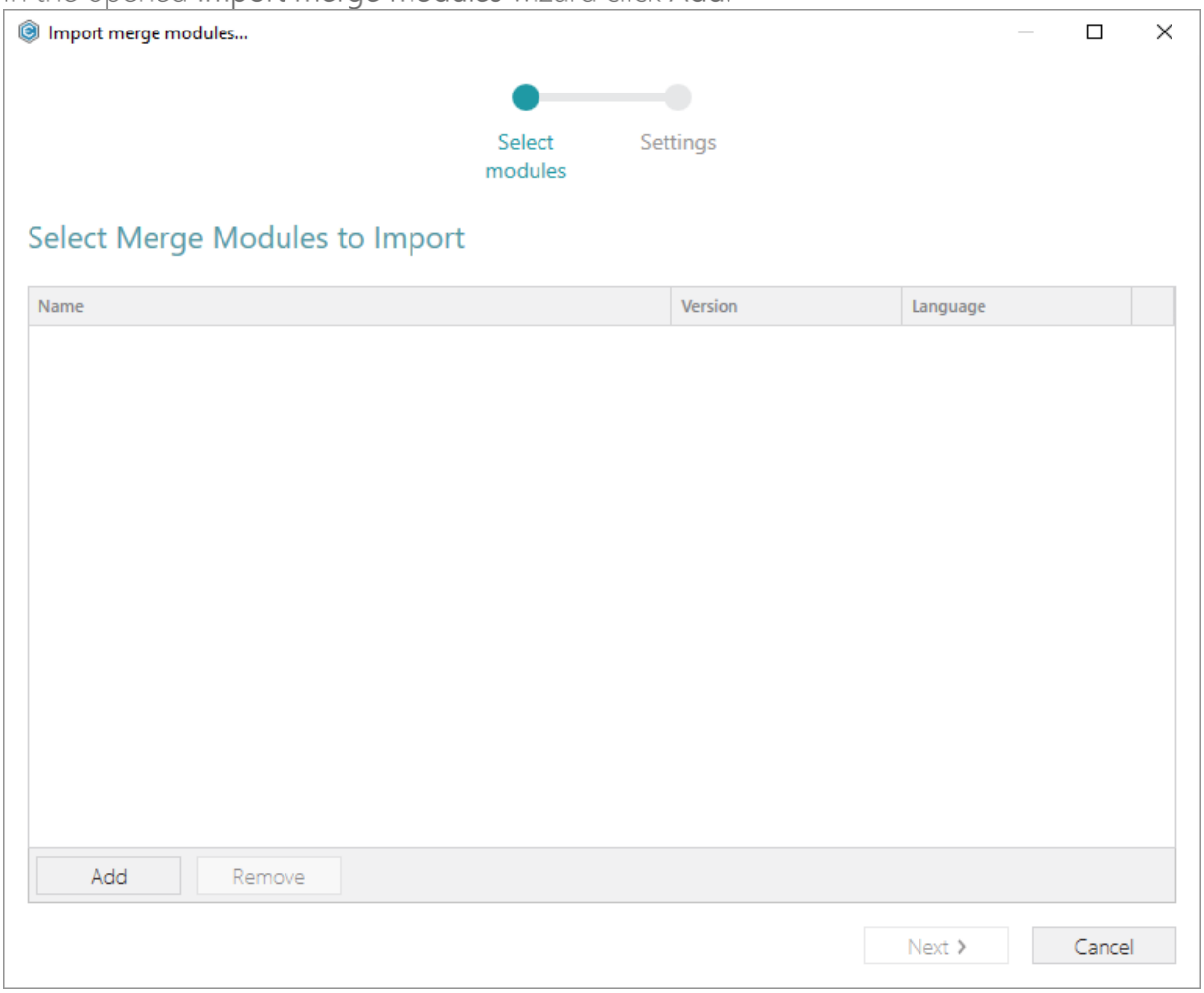
## 3.7.12.1 Import Merge Modules (MSM)

Import Merge Modules (MSM) to the package to deliver the shared resources properly.

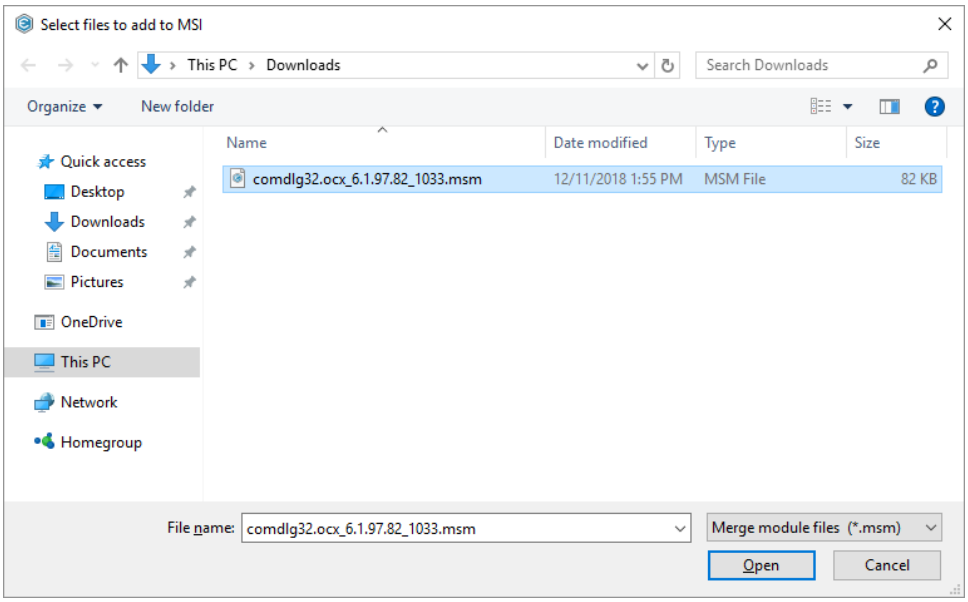
- Select Add and import -> Import merge modules from the MENU.



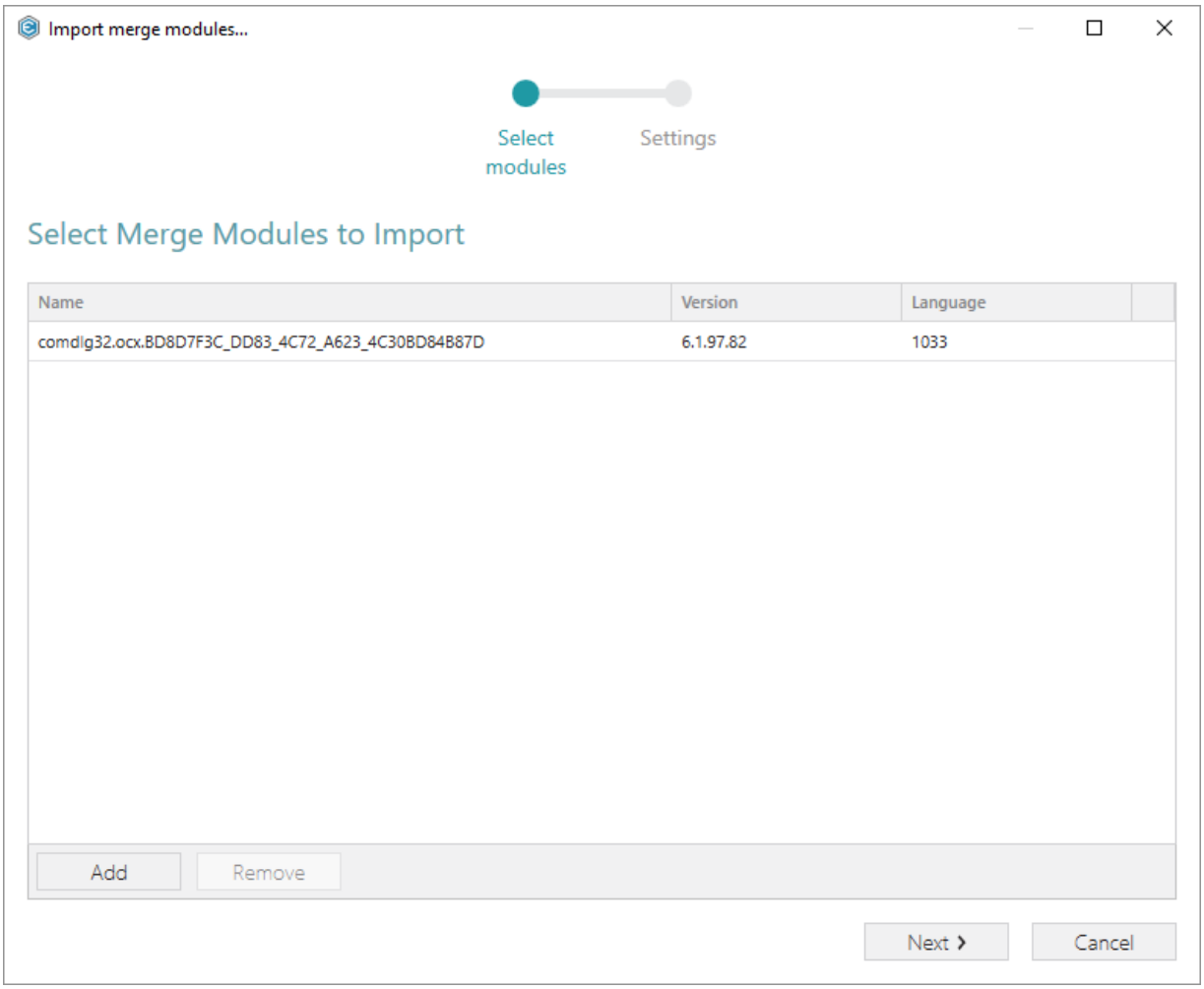
- In the opened Import merge modules wizard click Add.




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

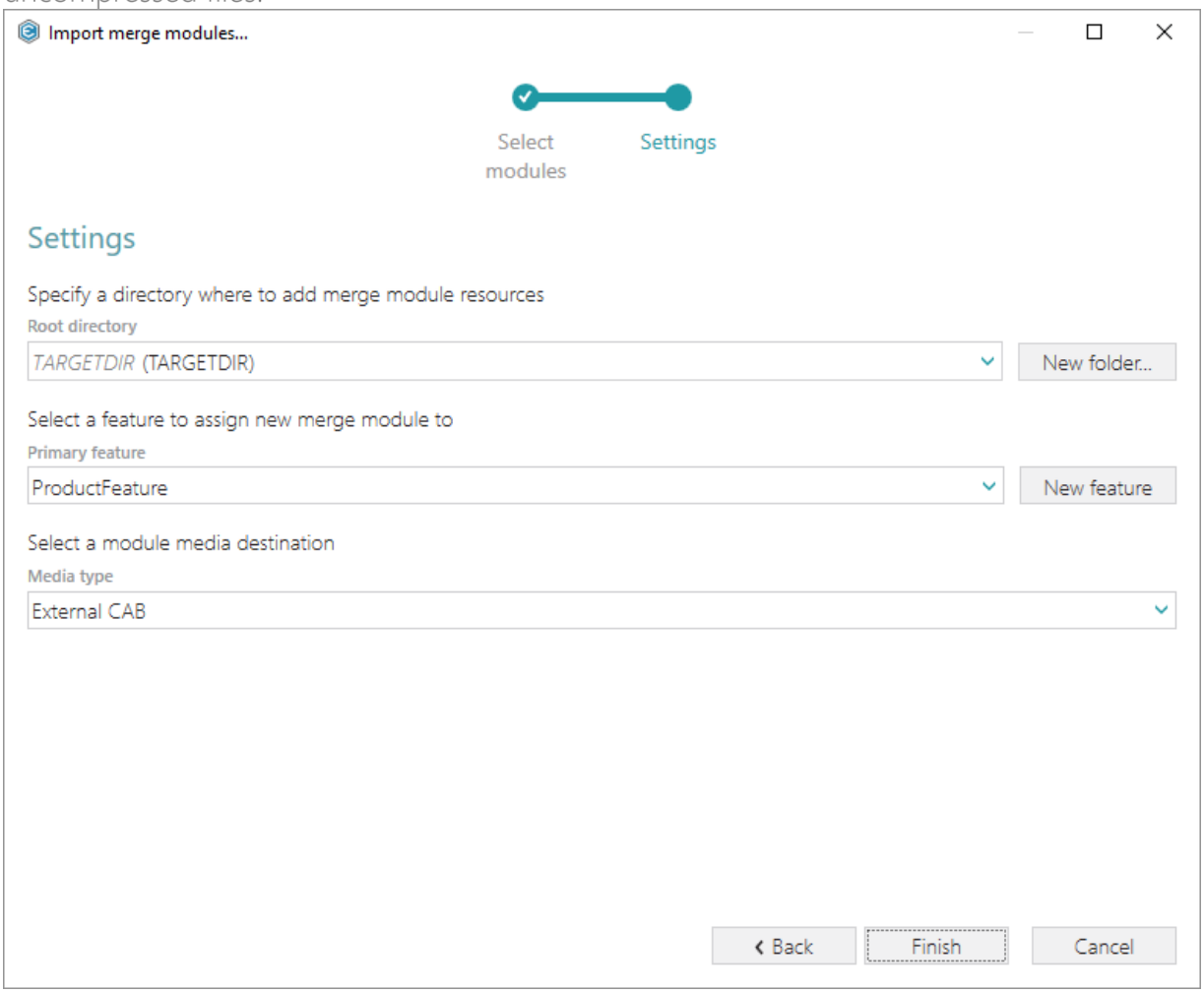


- Click Next.

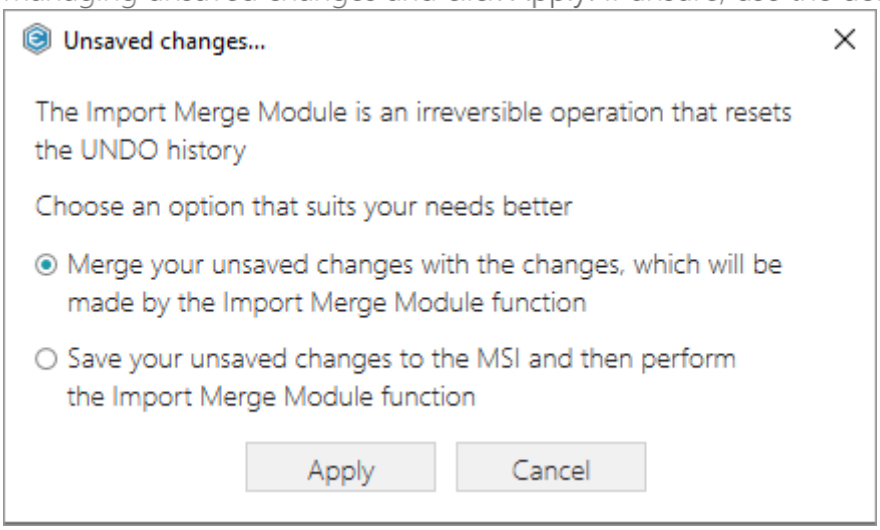



- Select the import settings and click Finish. The Root directory is used as a parent folder for the folder from the merge modules. In most cases, a path to the folder from the merge module is re-defined via the Custom action. The component from the merge module is linked to the selected Primary feature. The Media type option defines how the merge modules files will be saved to the files system: as external cabinet (CAB) or as

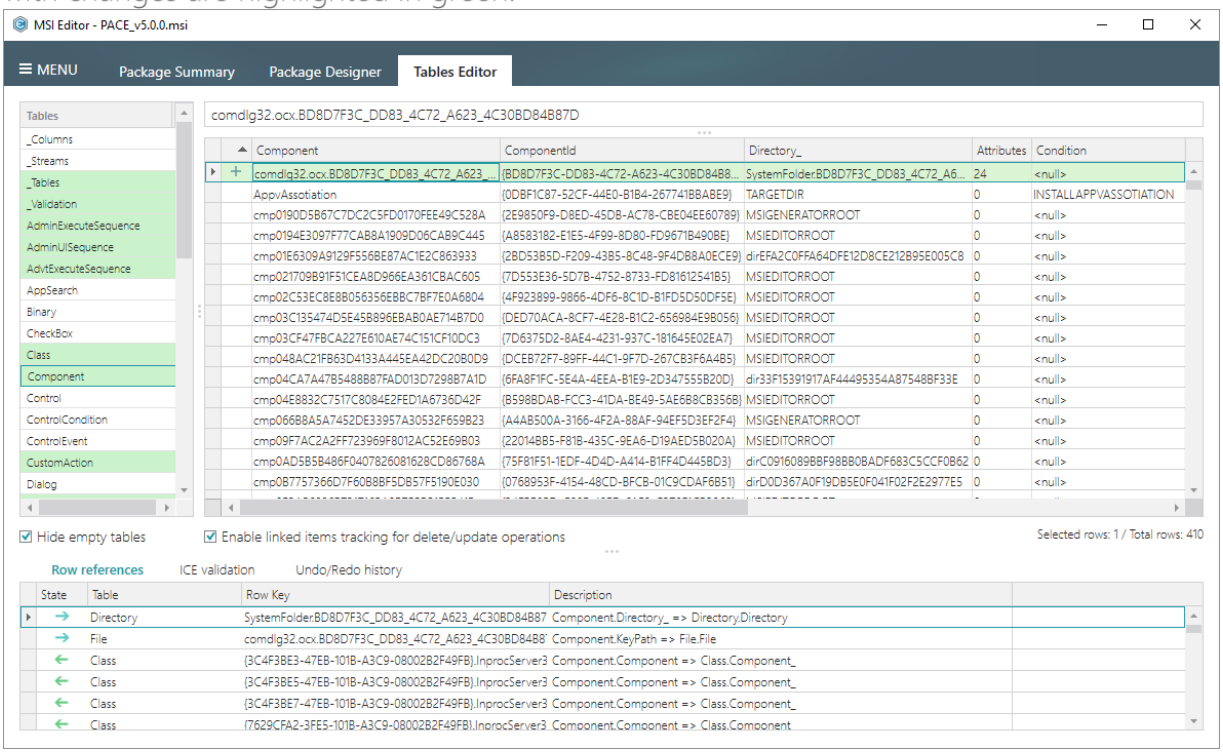
uncompressed files.



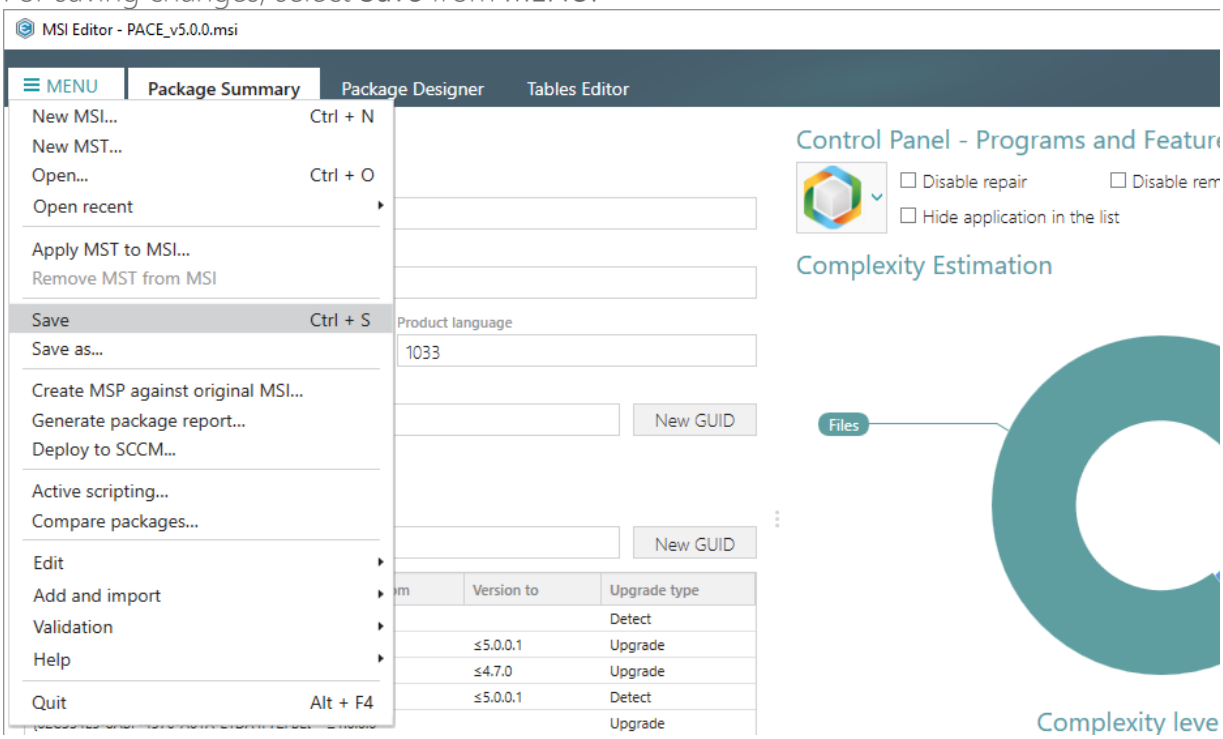

- If you have the unsaved changes and as far as the Import merge modules operation could not be reverted and resets the undo/redo history, you have to choose one of options of managing unsaved changes and click Apply. If unsure, use the default one.



- Find the resouces of the imported merge modues in the Tables Editor tab. Tables and rows with changes are highlighted in green.




- For saving changes, select Save from MENU.

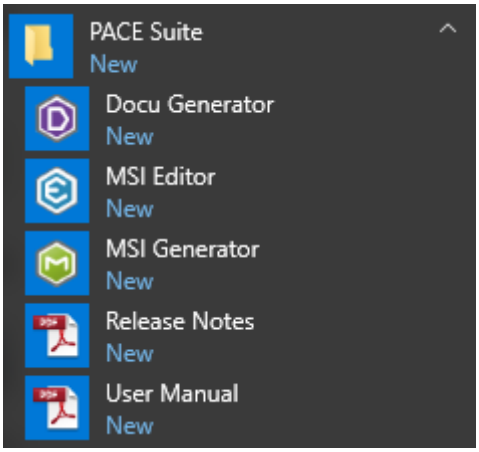


## 3.8 Edit Project

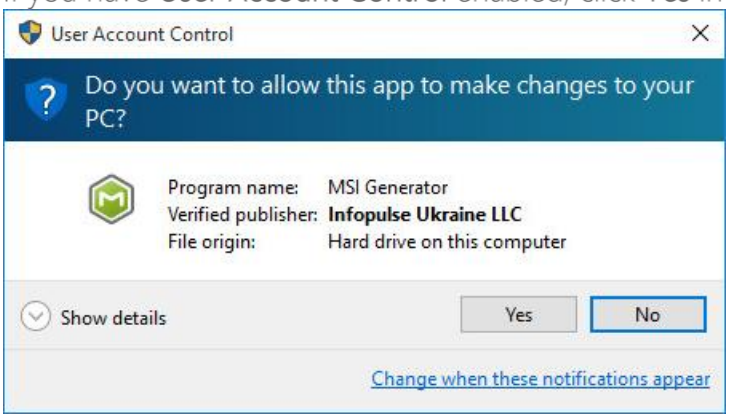
### 3.8.1 Pre-Condition

Open a project file following the instructions below:

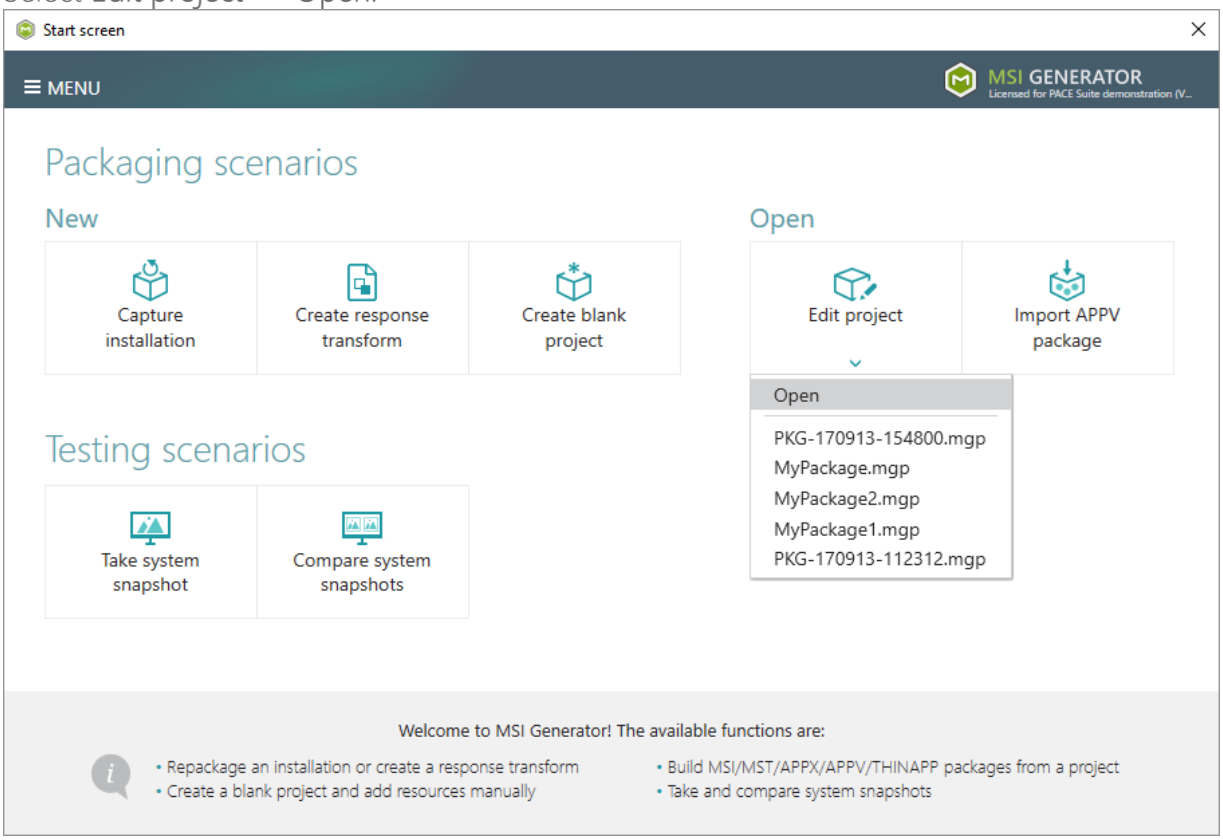
- Launch MSI Generator from the desktop or the start menu shortcut.



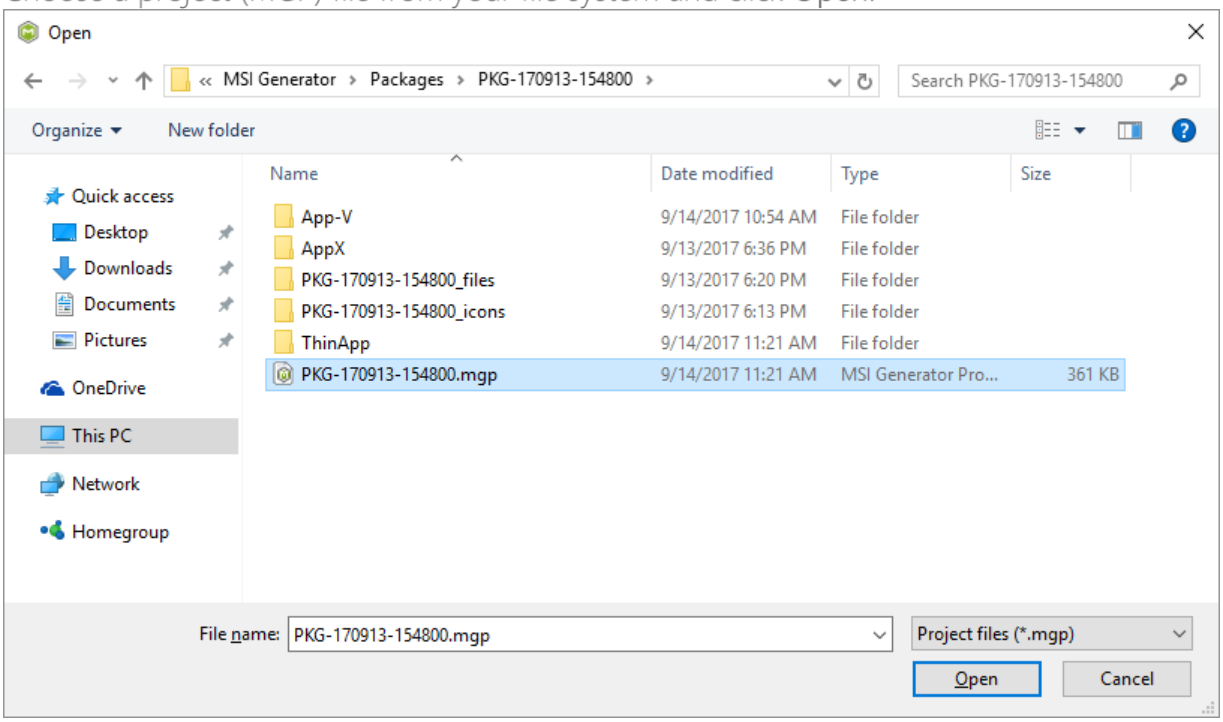

- If you have User Account Control enabled, click Yes in the opened window.



- Select Edit project -> Open.




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



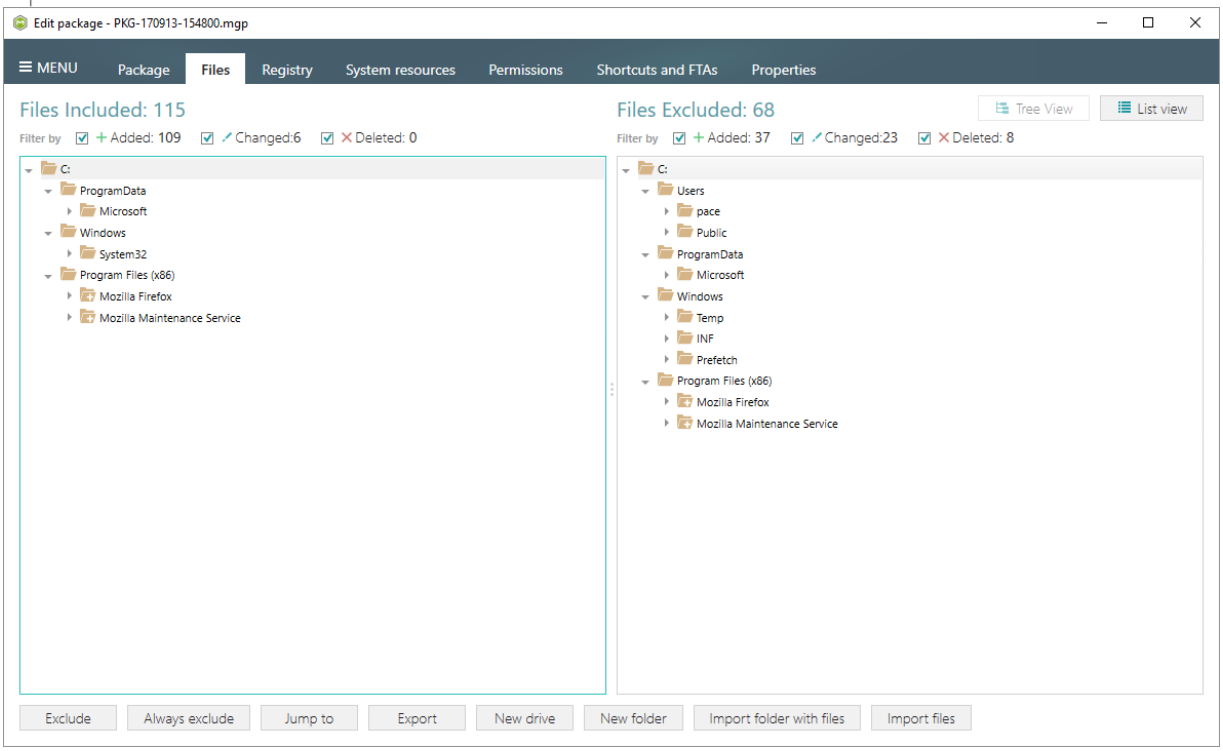
## 3.8.2 File and Folders

MSI Generator filters captured files and folders automatically and puts excluded (filtered) ones to the right pane, named '**Files Excluded**', and the rest of resources – to the left pane, named '**Files Included**'. Resources from the '**Files Included**' pane will be saved to your package, and resources from the '**Files Excluded**' will not be a part of the package, it will be kept as excluded ones only in the project. Note that you can include or exclude necessary resources at any time.

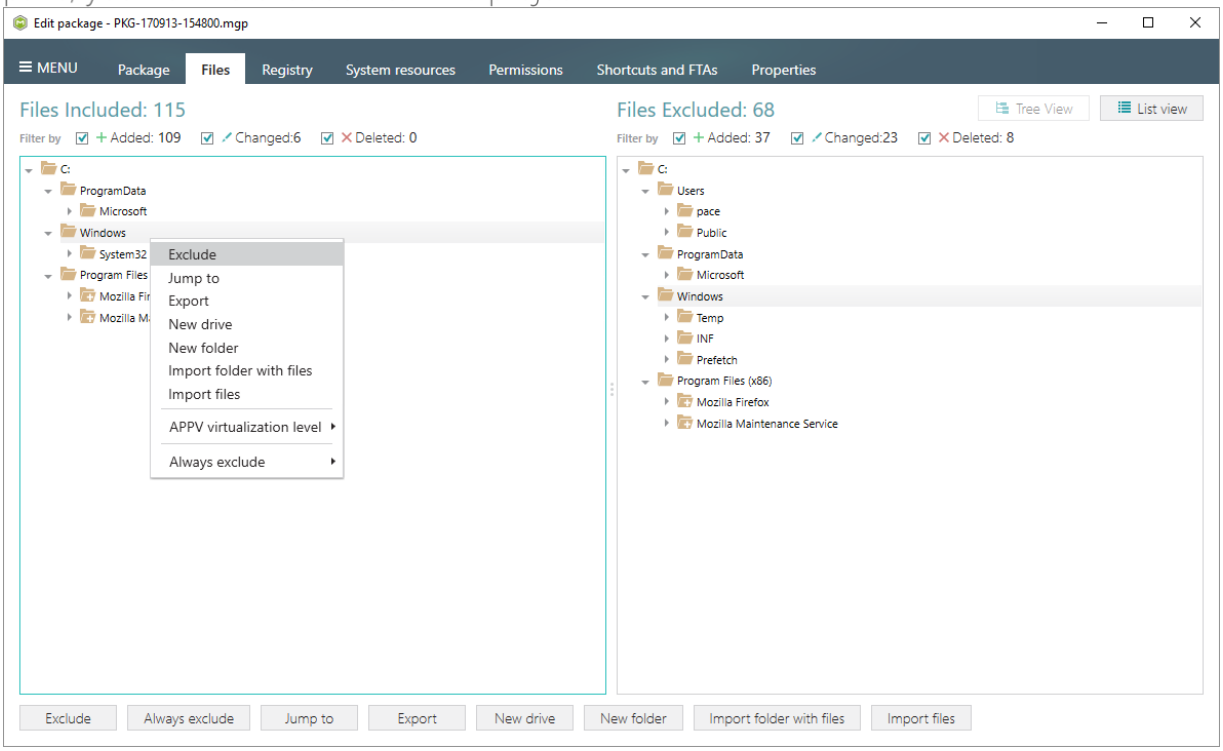
### 3.8.2.1 Exclude Files and Folders

Exclude unnecessary resources from the project.

- Open the **Files** tab.



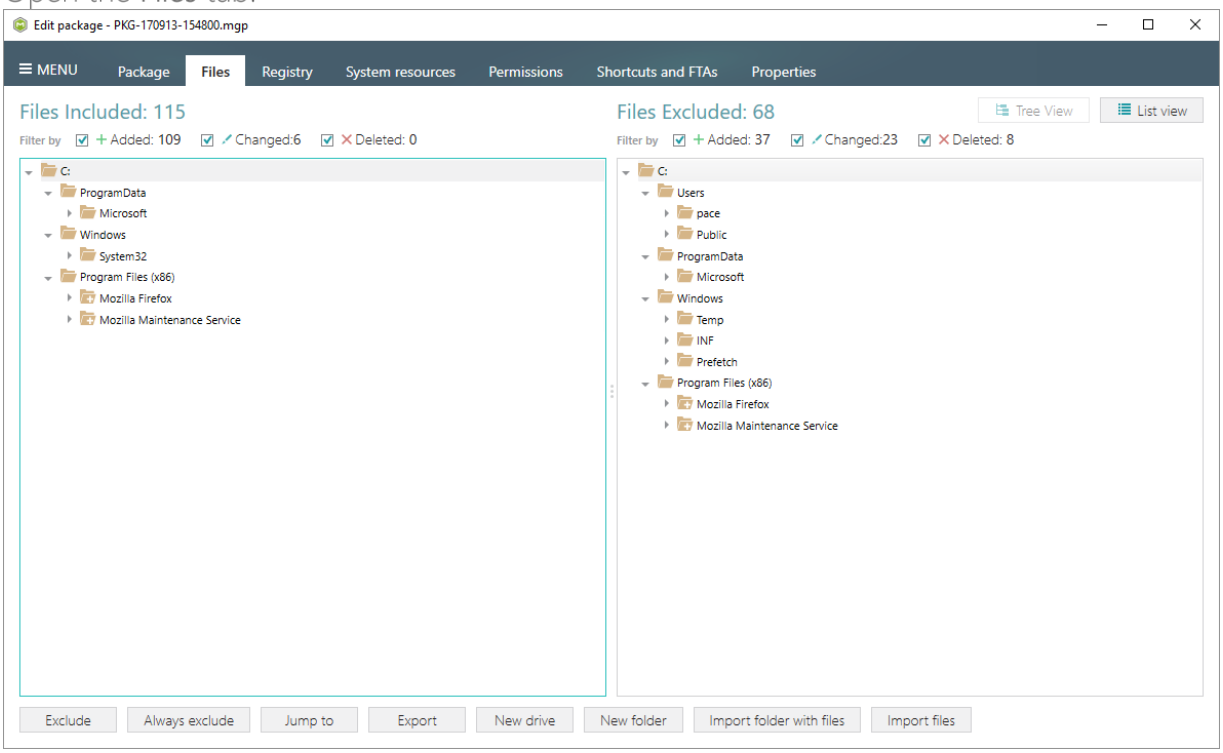
- Select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane, you want to exclude from the project.



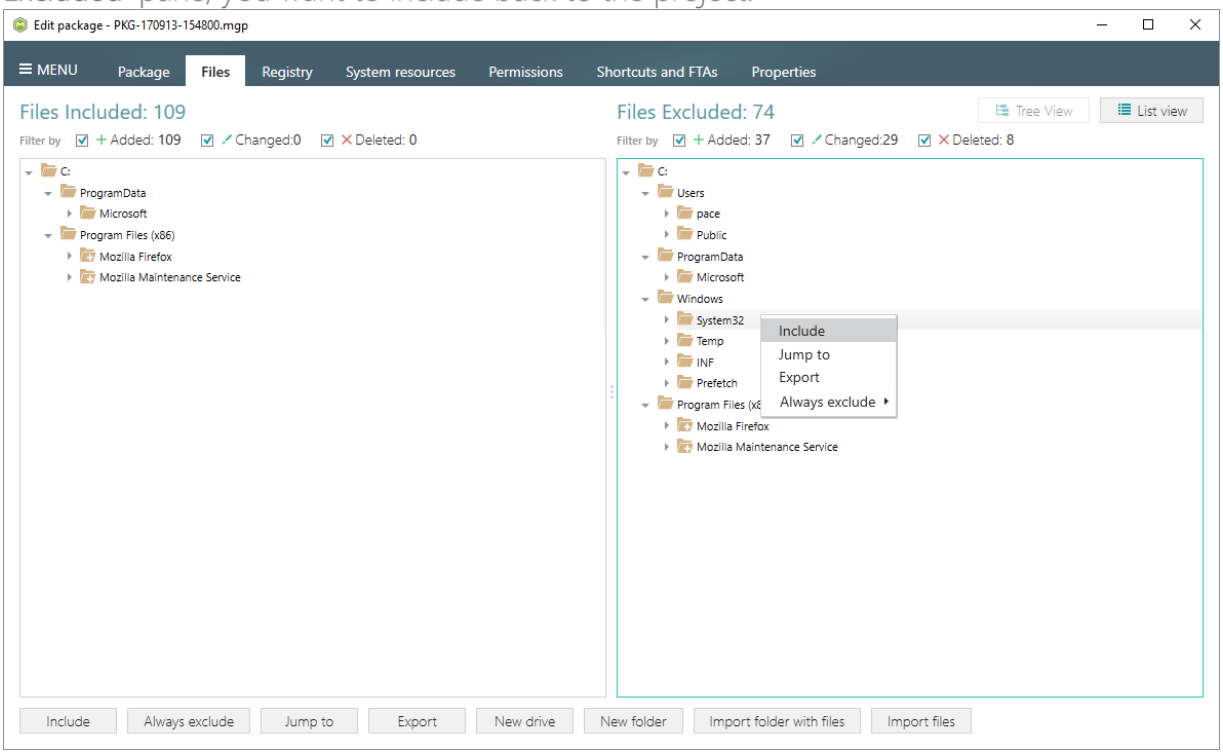

### 3.8.2.2 Include Files and Folders

Include necessary resources, previously excluded from the project.

- Open the **Files** tab.

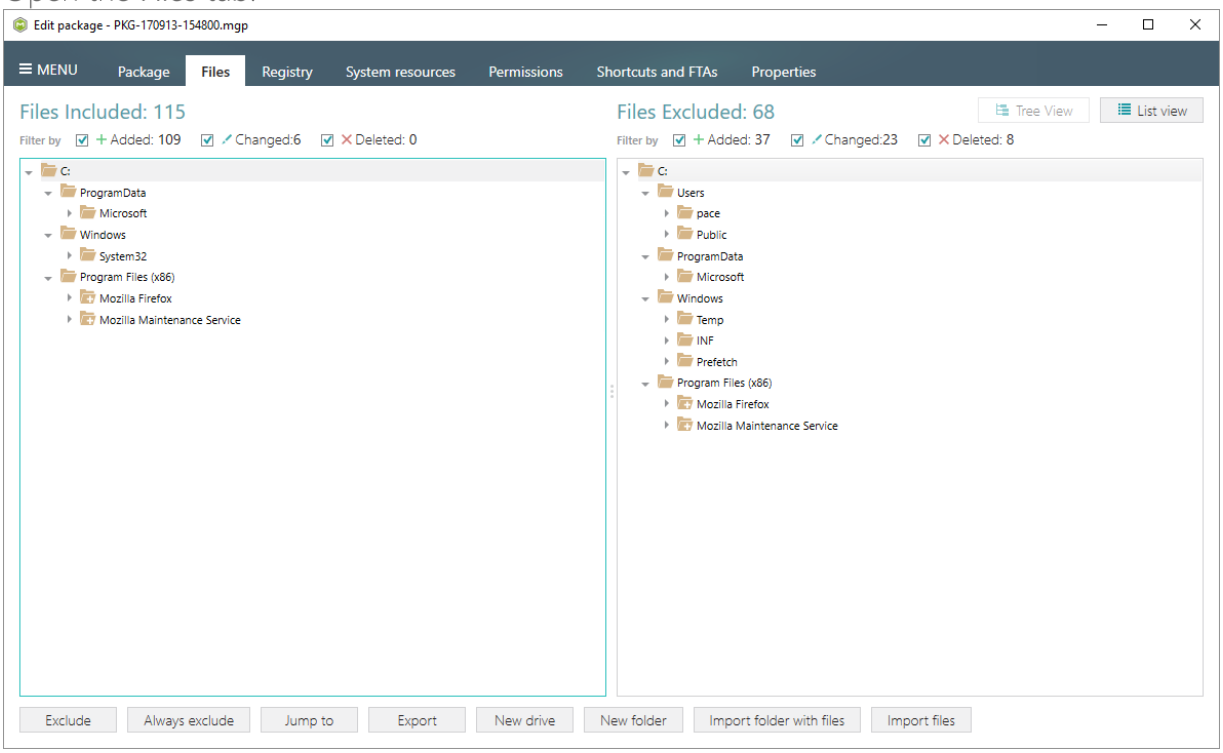


- Select **Include** from the context menu of an item, which is located in the right '**Files Excluded**' pane, you want to include back to the project.

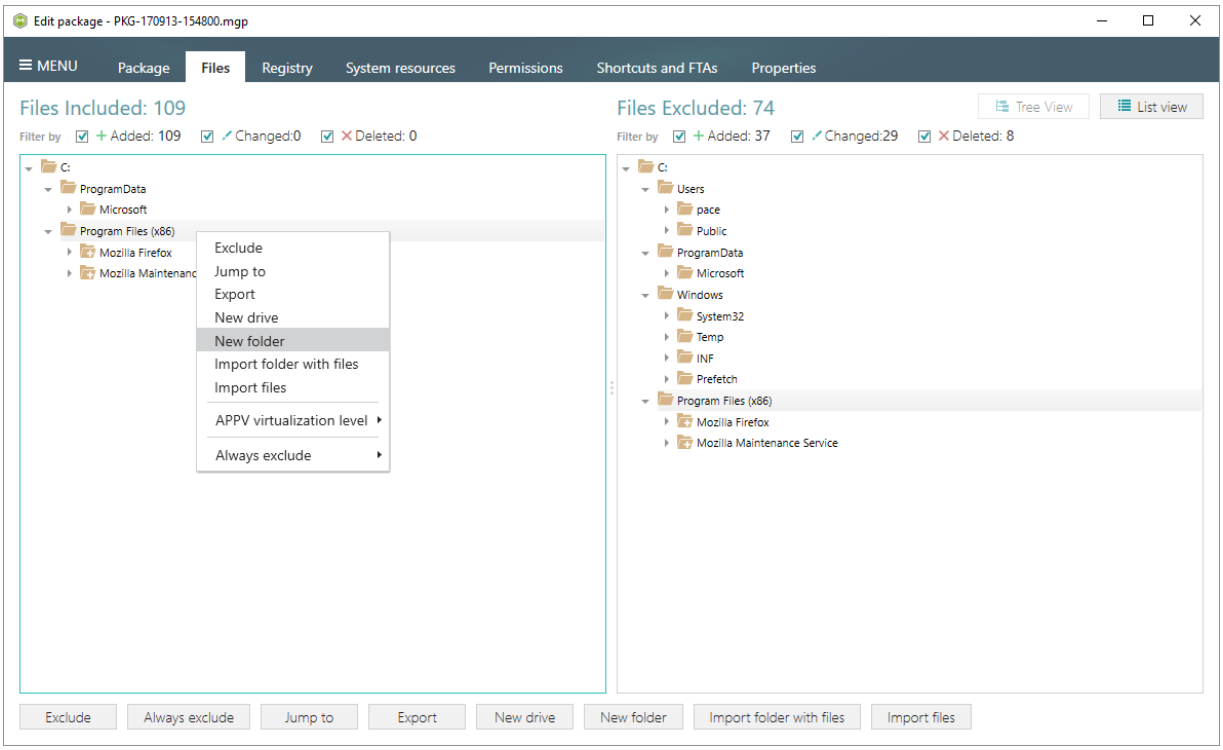



## 3.8.2.3 Add Custom Folder

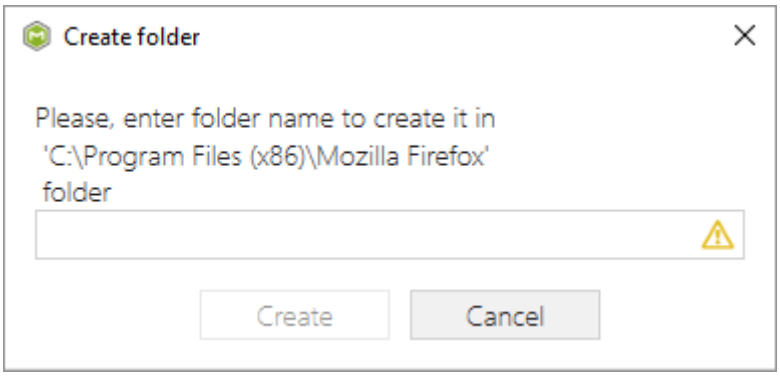
- Open the **Files** tab.



- Select **New folder** from the context menu of a parent folder, to which you want to add a new custom folder.

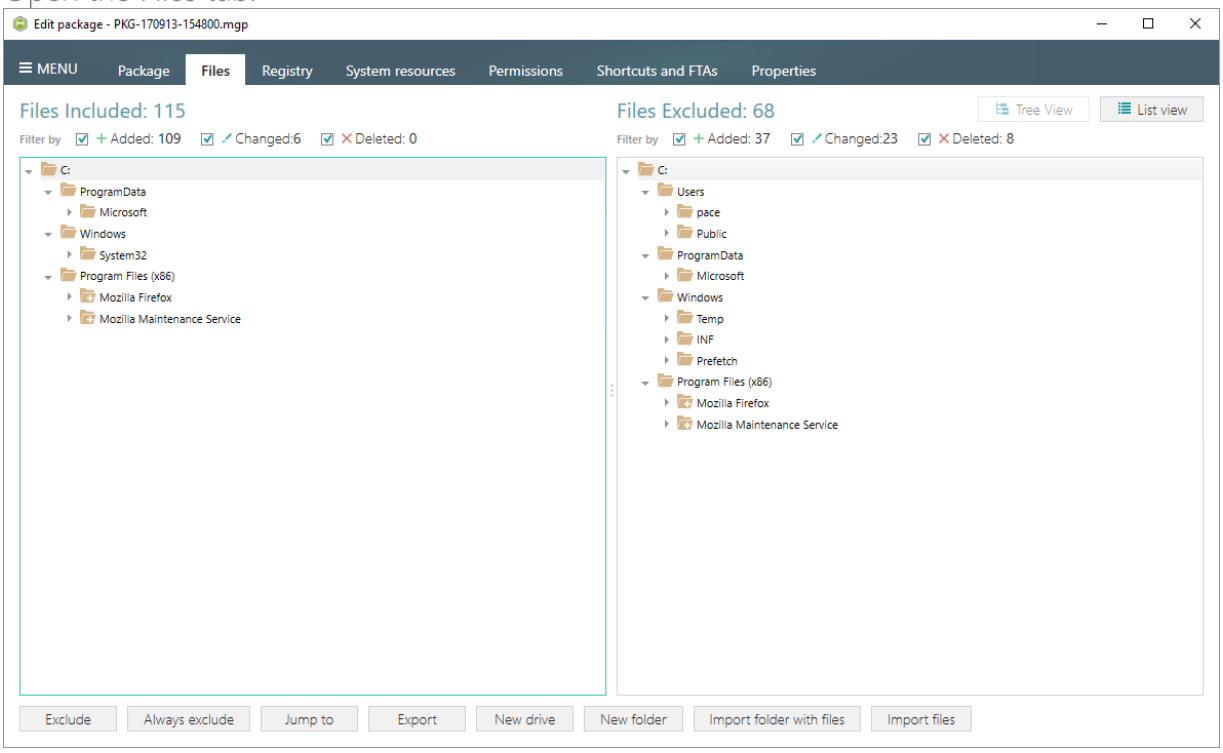



- Enter a folder name and click **Create**.

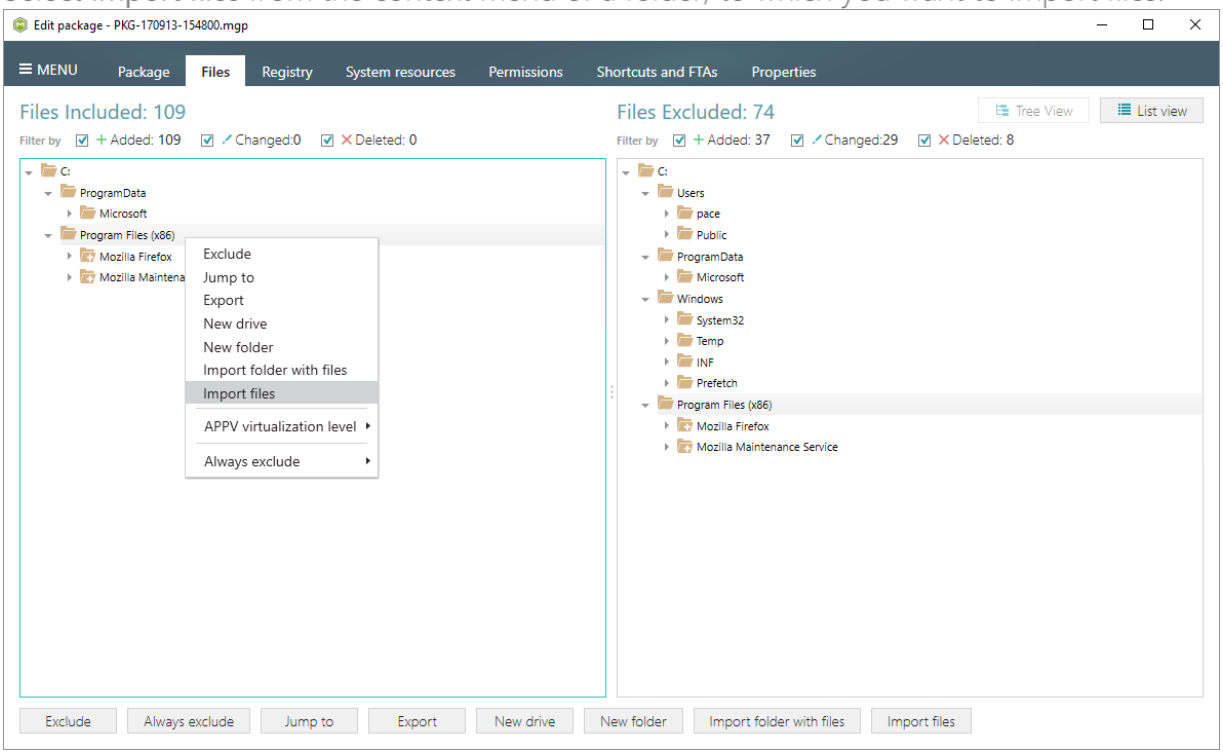


### 3.8.2.4 Import Files

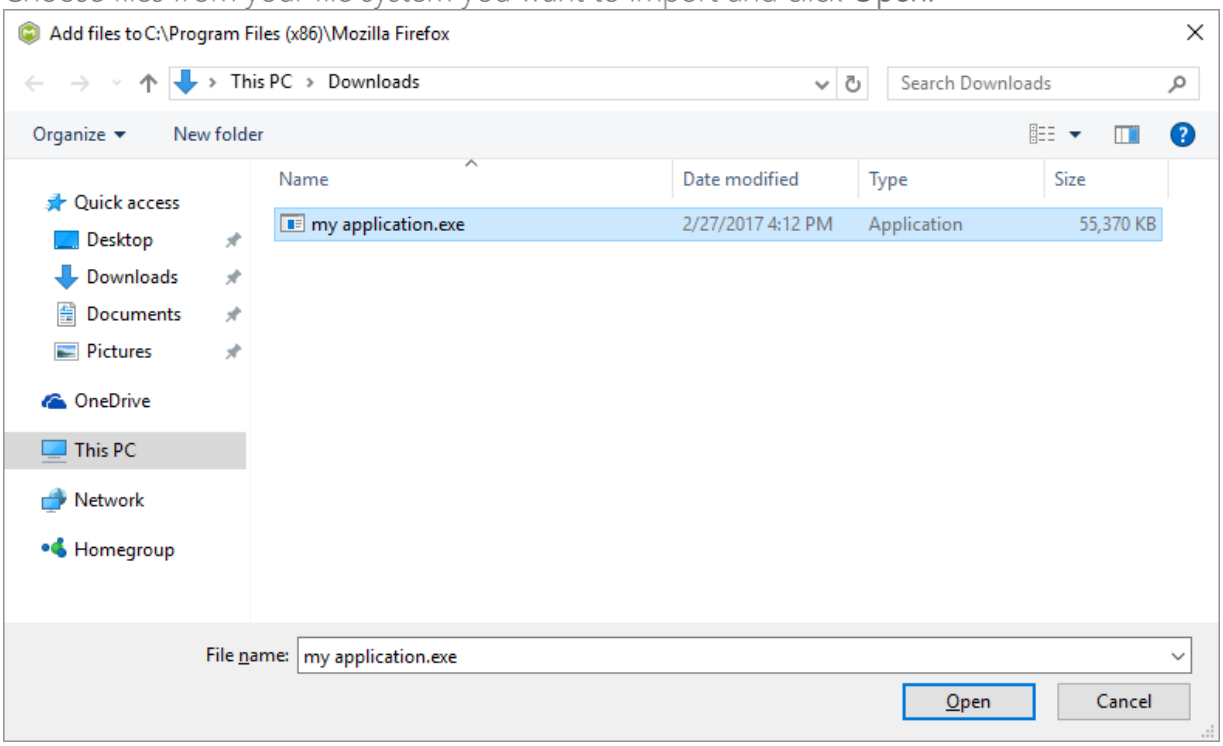
- Open the **Files** tab.




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



- Choose files from your file system you want to import and click Open.



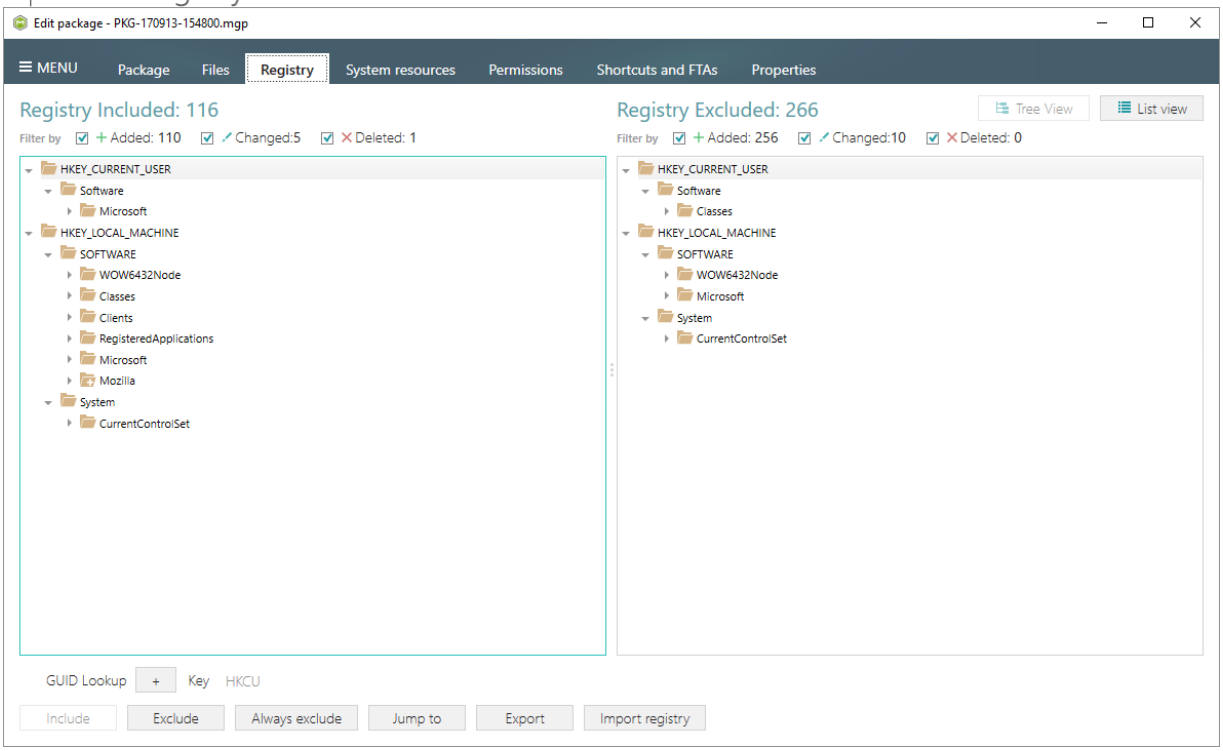
## 3.8.3 Registry

MSI Generator filters captured registry entries automatically and puts excluded (filtered) ones to the right pane, named '**Registry Excluded**', and the rest of resources – to the left pane, named '**Registry Included**'. Resources from the '**Registry Included**' pane will be saved to your package, and resources from the '**Registry Excluded**' will not be a part of the package, it will be kept as excluded ones only in the project. Note that you can include or exclude necessary resources at any time.

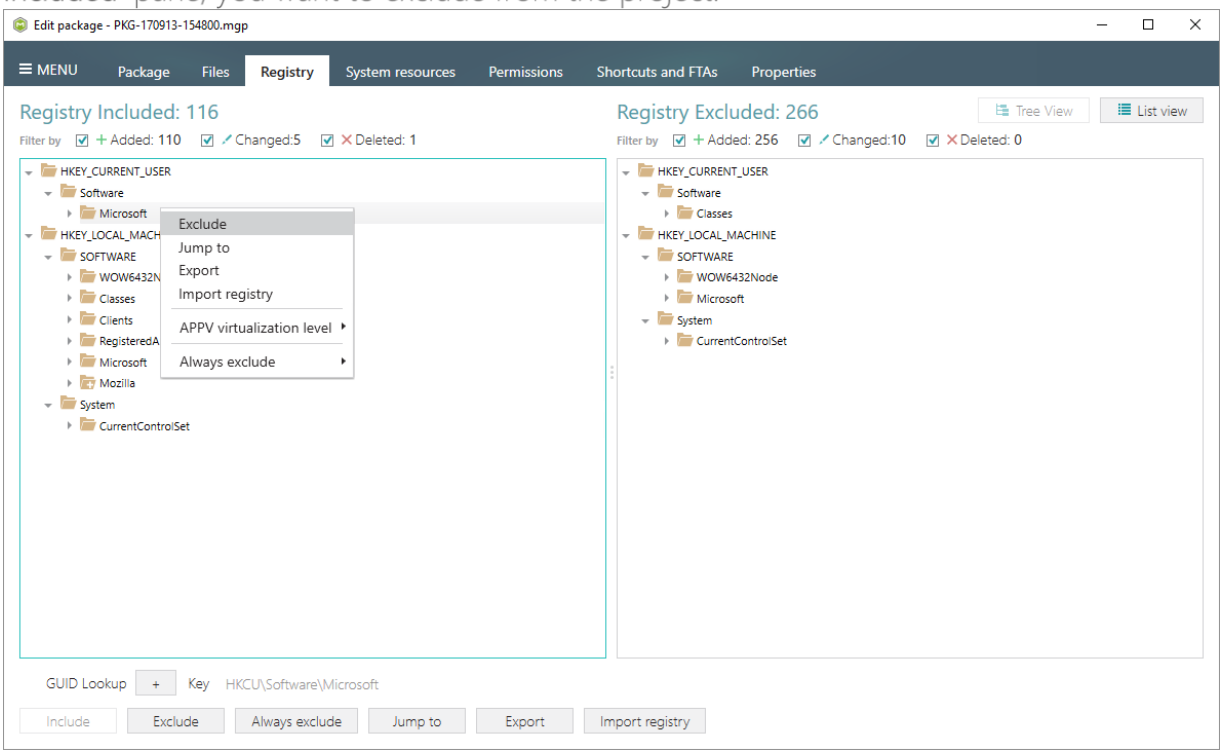
### 3.8.3.1 Exclude Registry Entries

Exclude unnecessary resources from the project.

- Open the **Registry** tab.



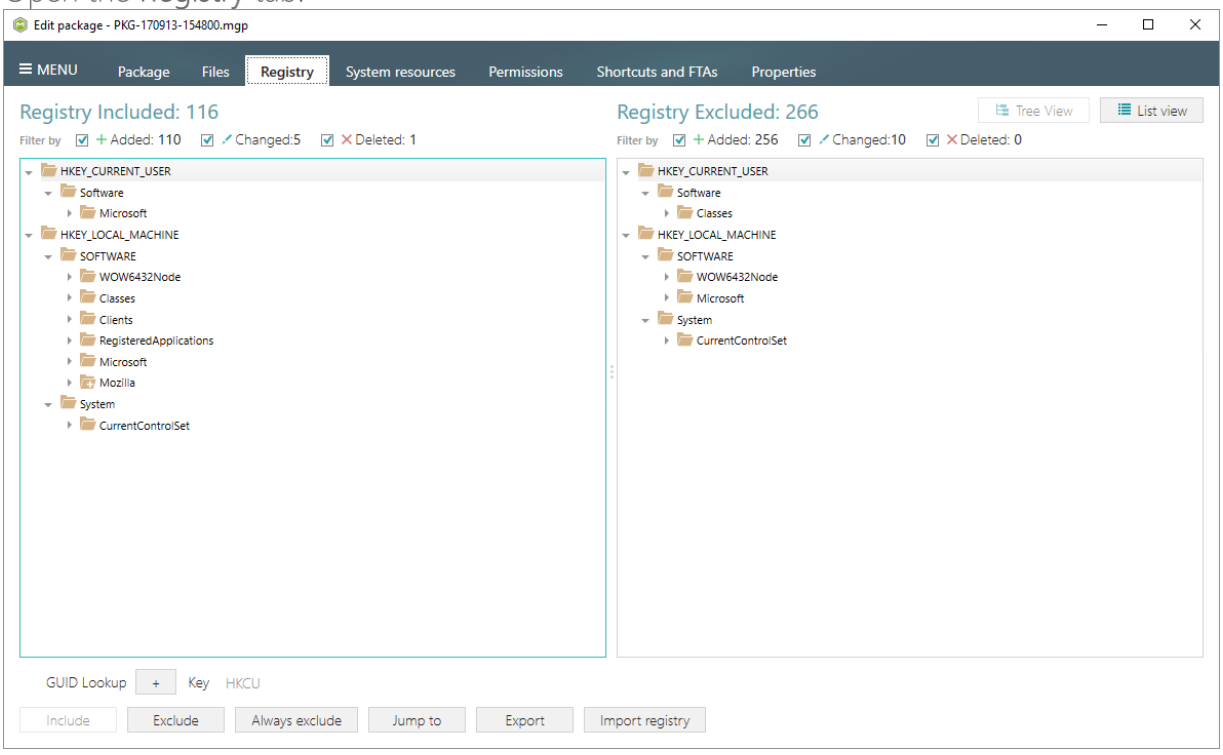
- Select **Exclude** from the context menu of an item, which is located in the left '**Registry Included**' pane, you want to exclude from the project.



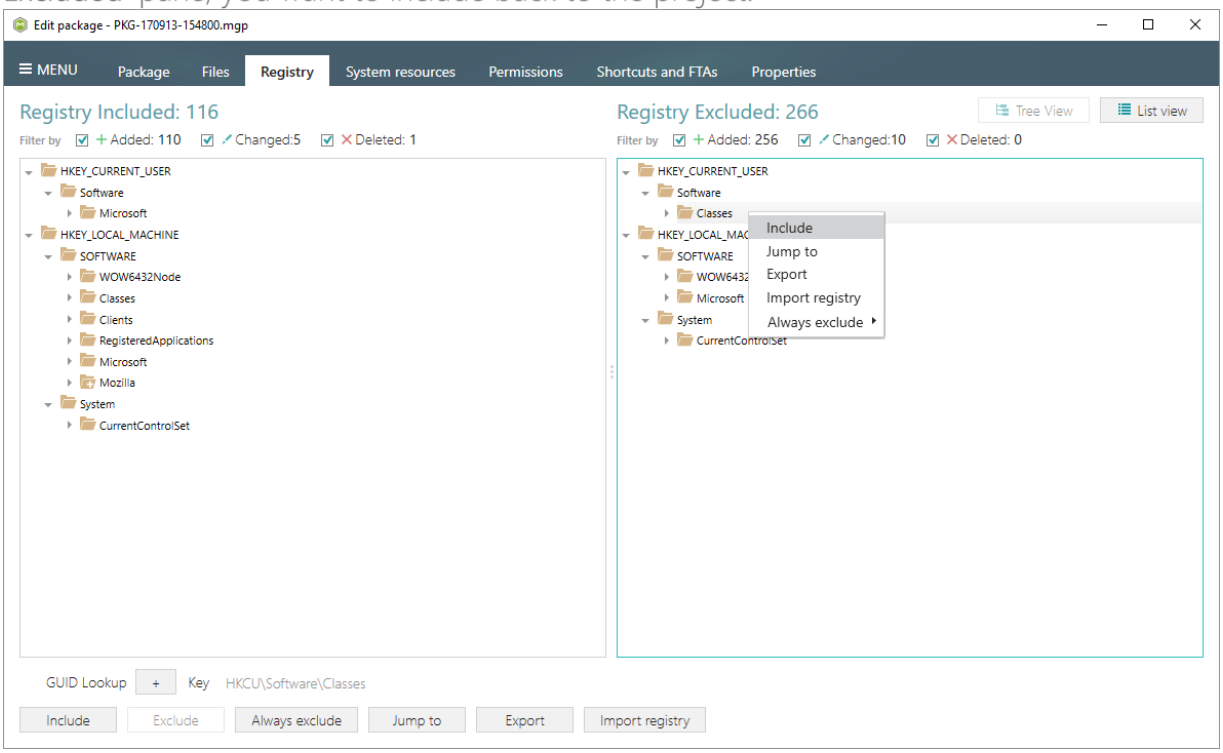

### 3.8.3.2 Include Registry Entries

Include necessary resources, previously excluded from the project.

- Open the **Registry** tab.

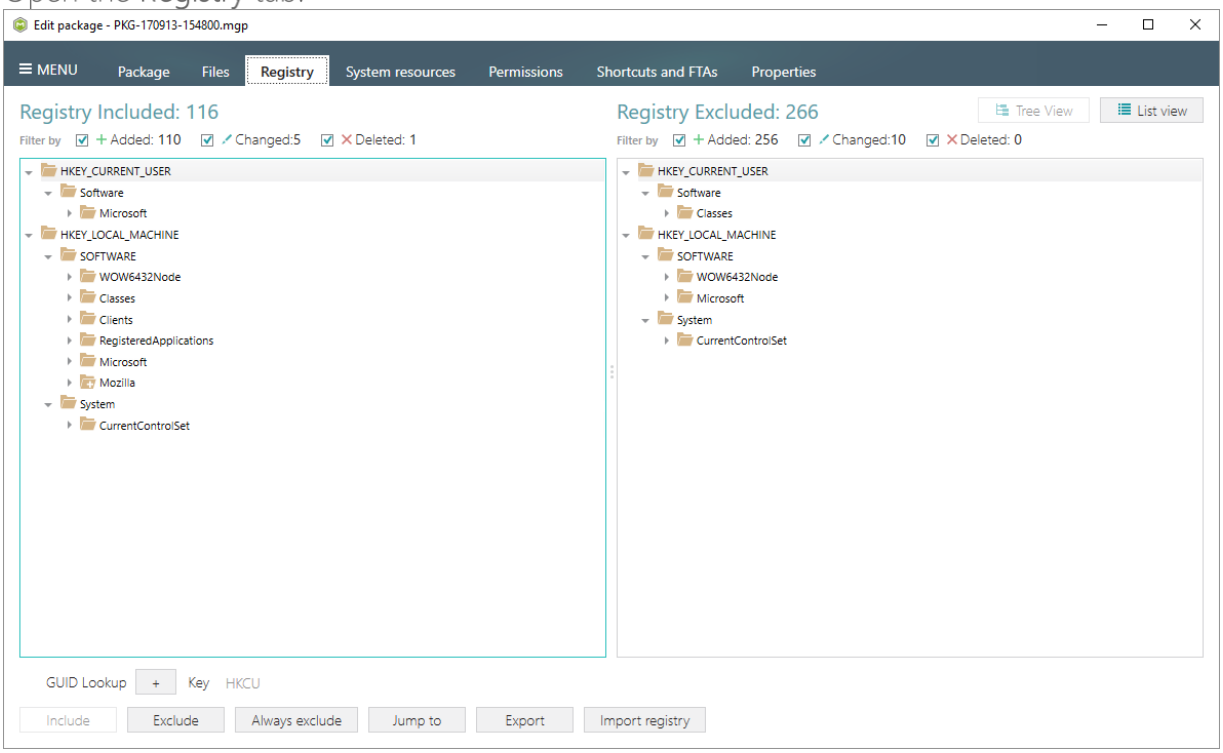


- Select **Include** from the context menu of an item, which is located in the right '**Registry Excluded**' pane, you want to include back to the project.

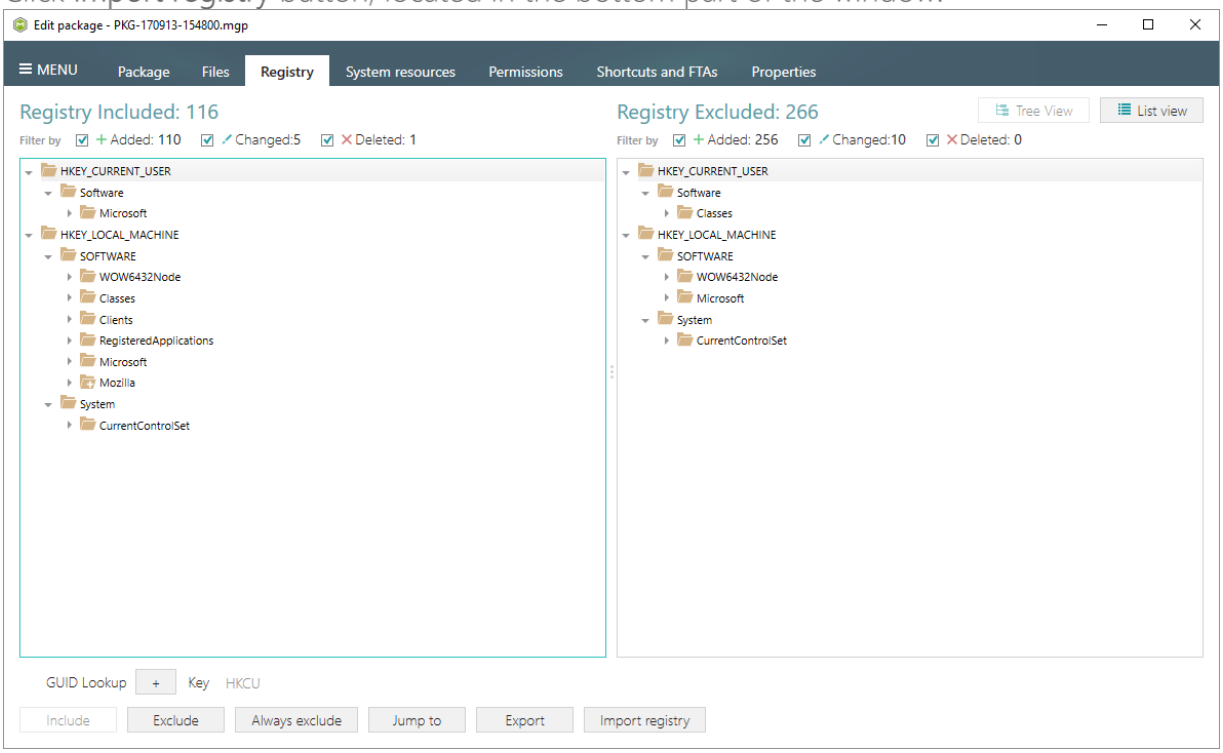



### 3.8.3.3 Import Registry

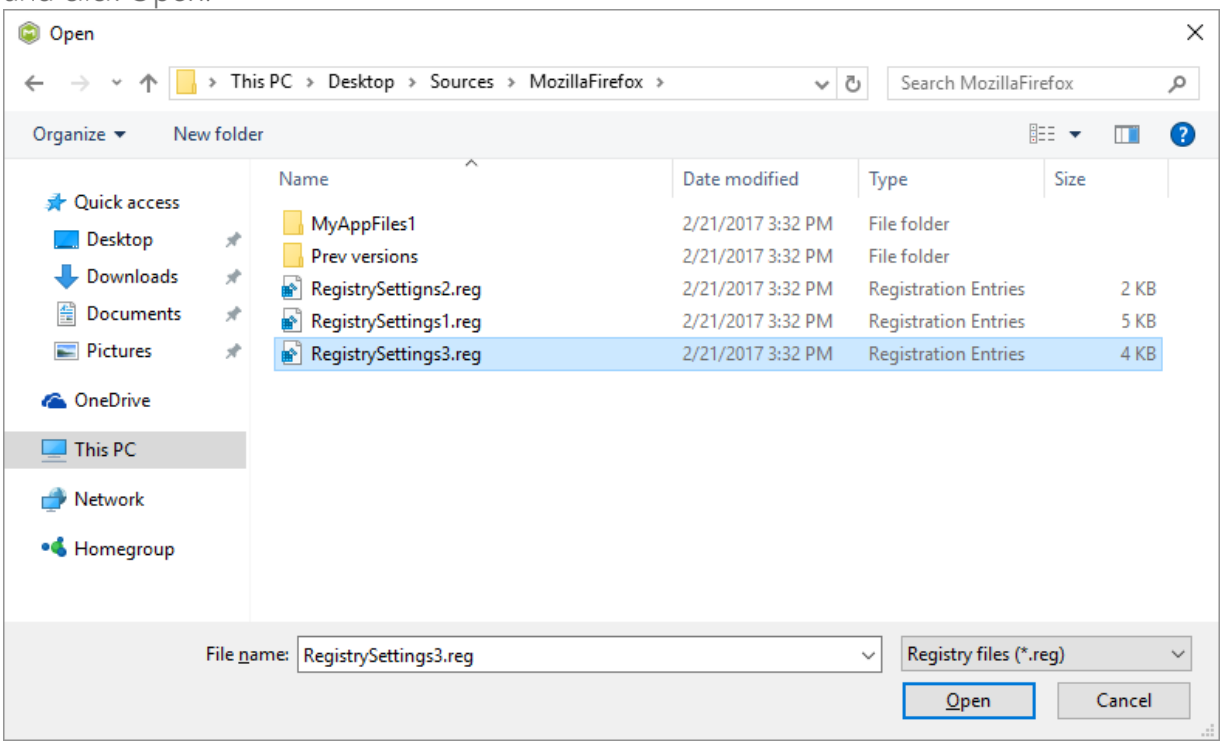
- Open the **Registry** tab.



- Click Import registry button, located in the bottom part of the window.

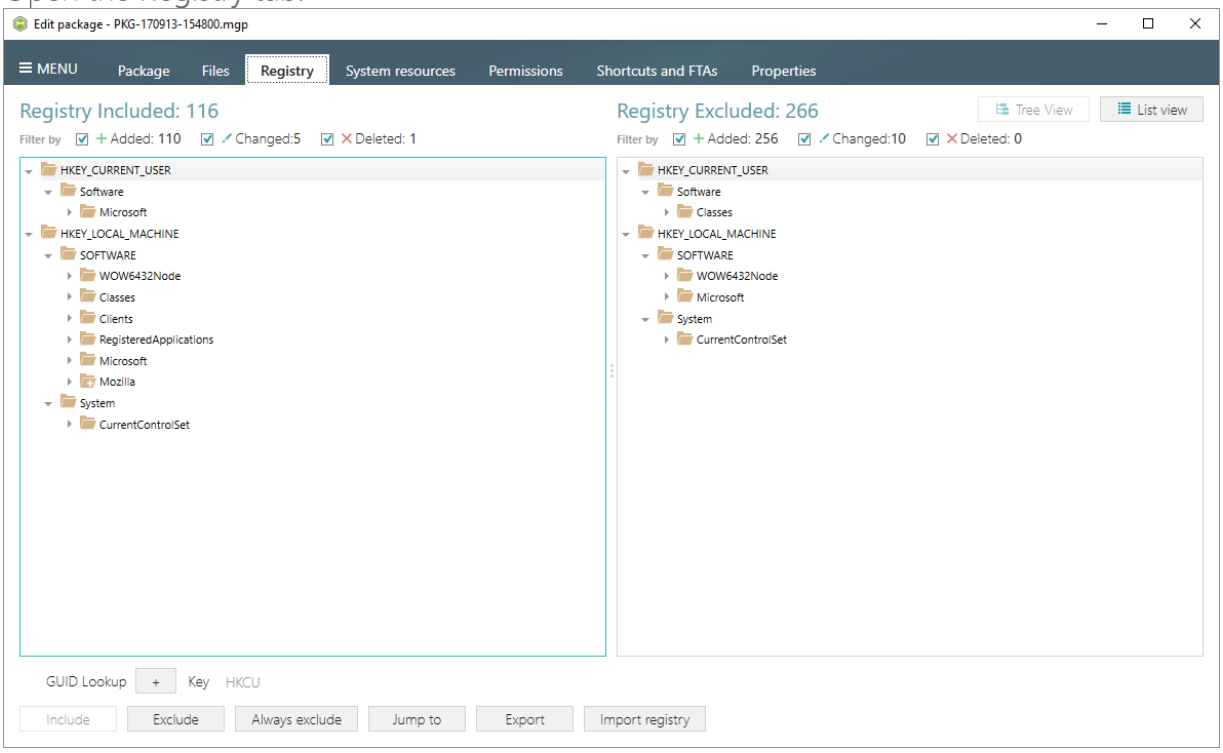



- Choose a REG file from your file system, content of which you want to import to the project and click Open.

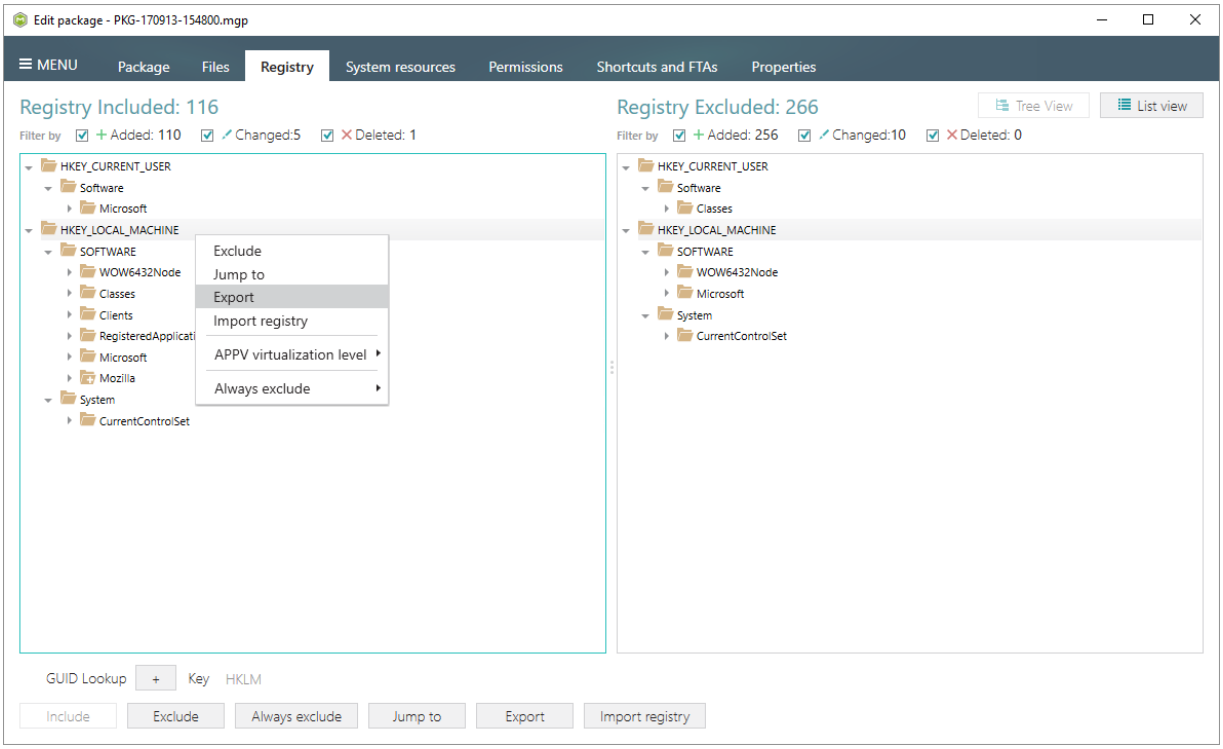


### 3.8.3.4 Export Registry

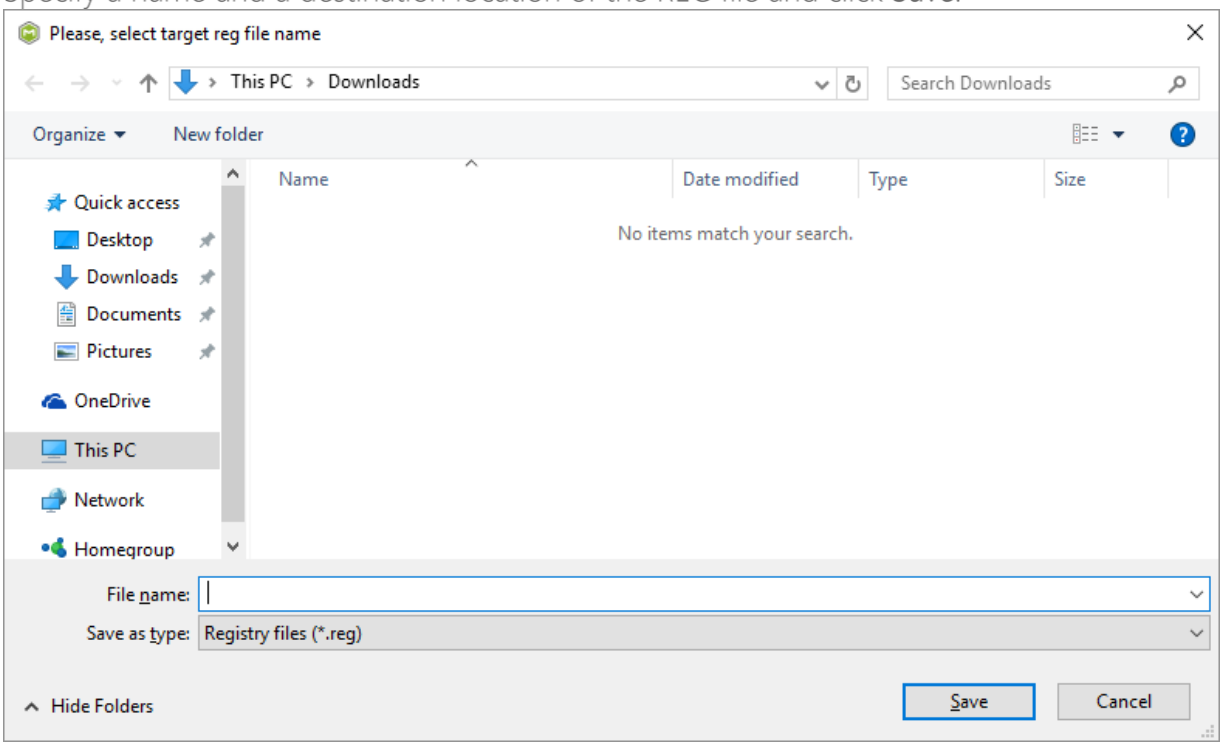
- Open the **Registry** tab.




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



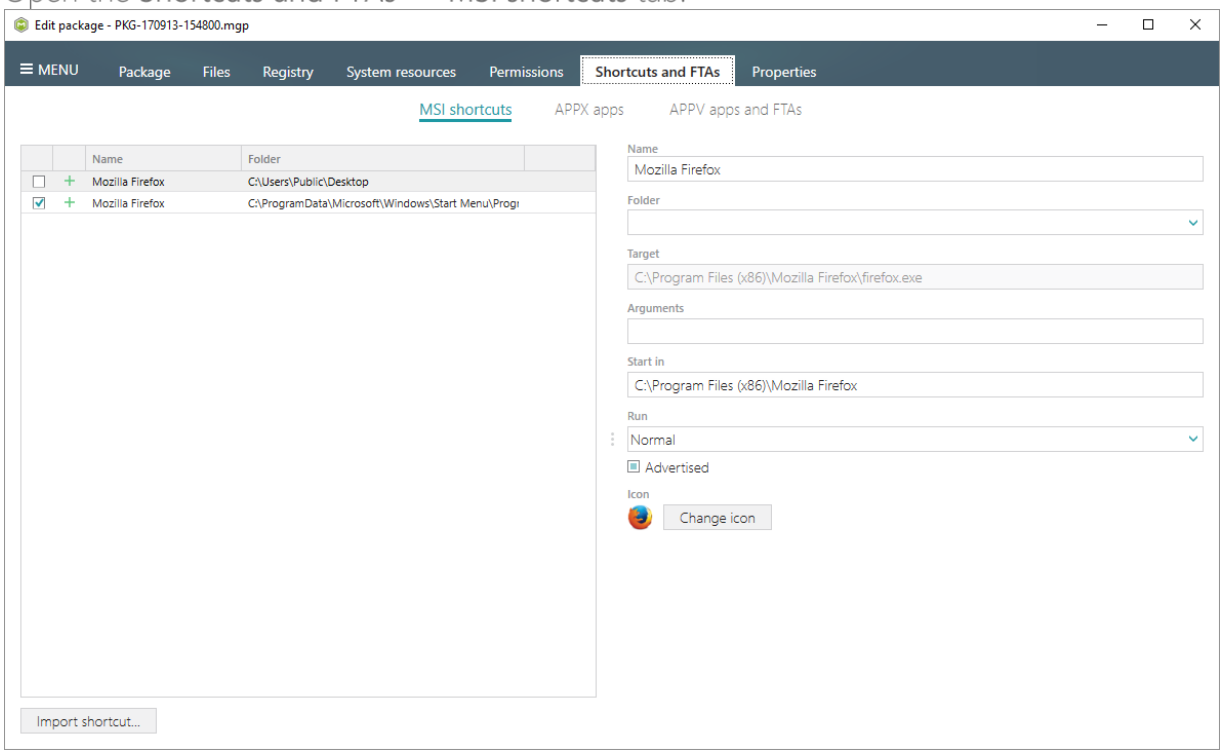
- Specify a name and a destination location of the REG file and click Save.



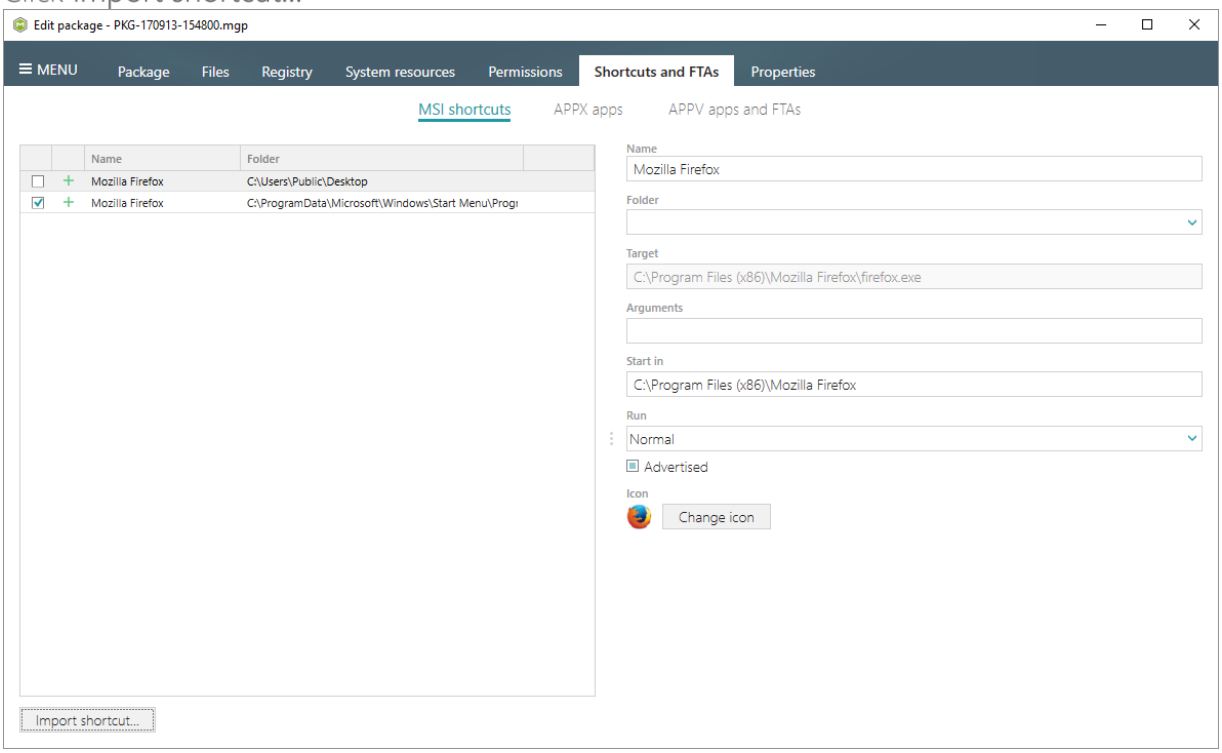

## 3.8.4 Shortcuts

### 3.8.4.1 Import Shortcut

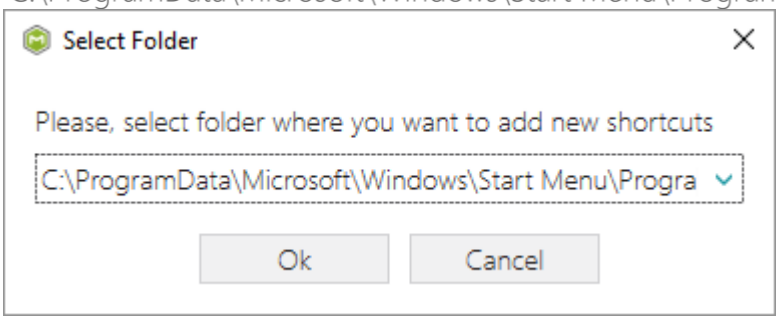
[1]. Open the Shortcuts and FTAs -> MSI shortcuts tab.



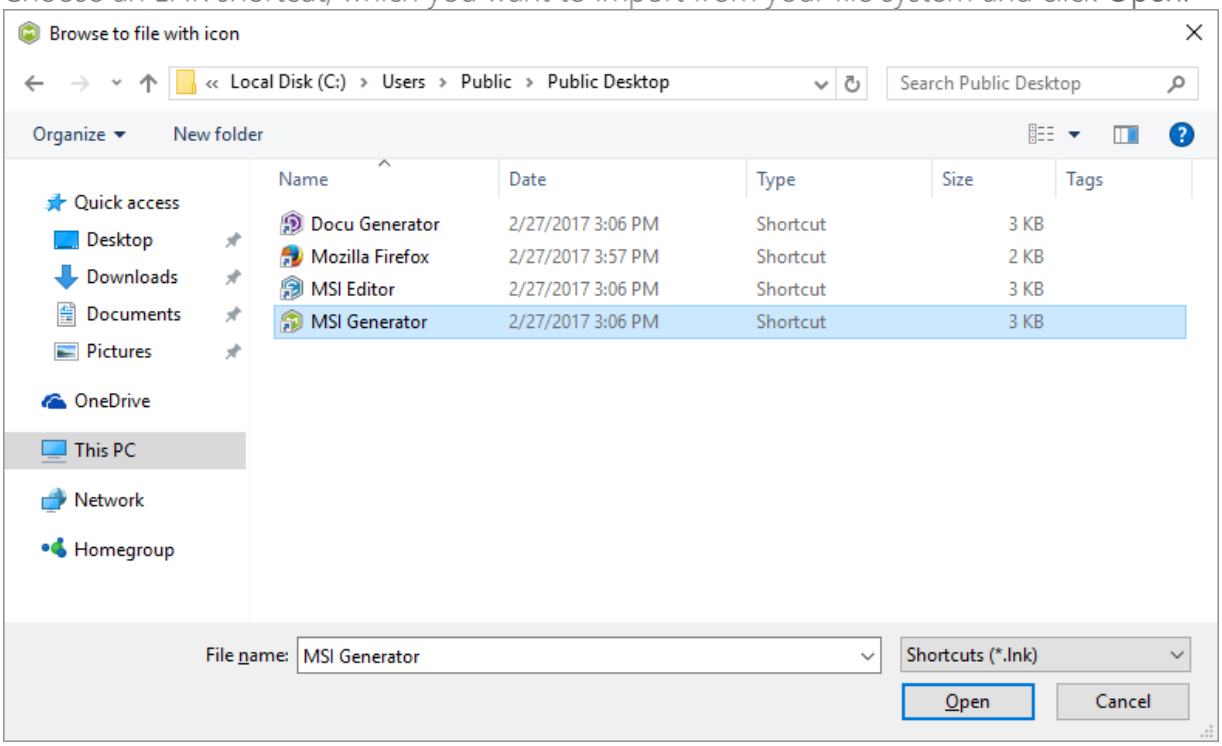
[2]. Click Import shortcut...




[3]. Select a folder, to which you want to import shortcuts (e.g. 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs') and click Ok.

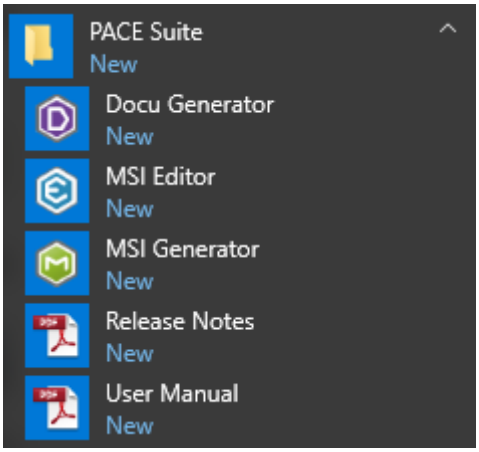


[4]. Choose an LNK shortcut, which you want to import from your file system and click Open.

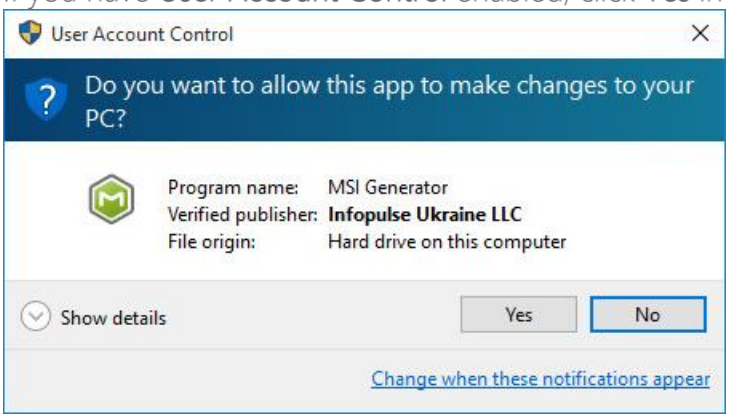


## 3.9 Edit APPV

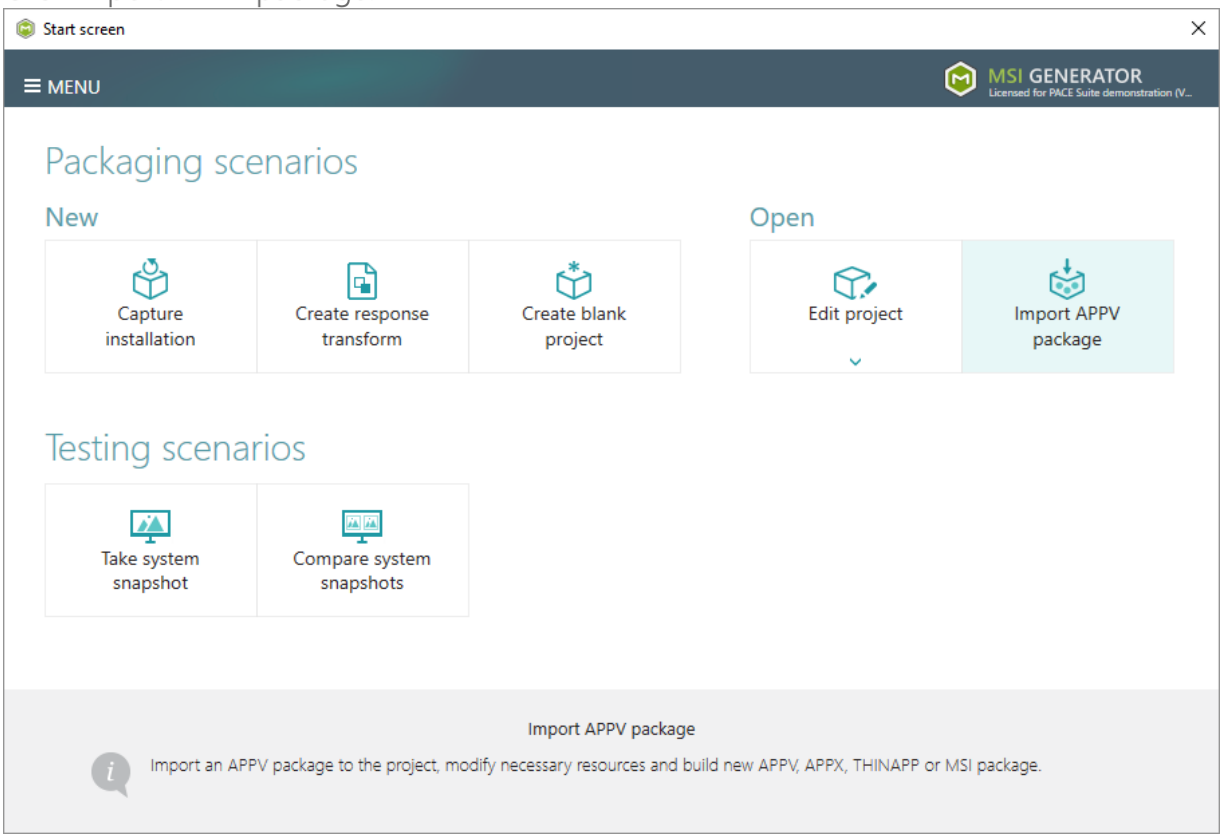
- Launch MSI Generator from the desktop or the start menu shortcut.



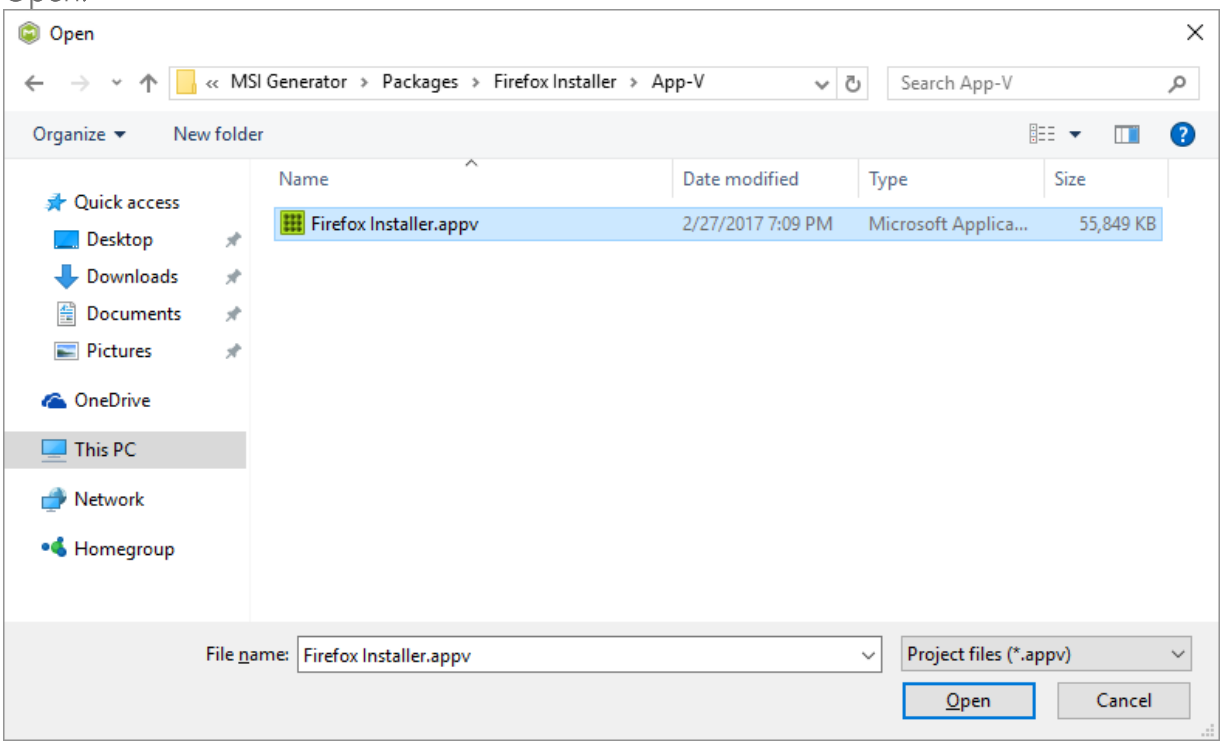

- If you have User Account Control enabled, click Yes in the opened window.



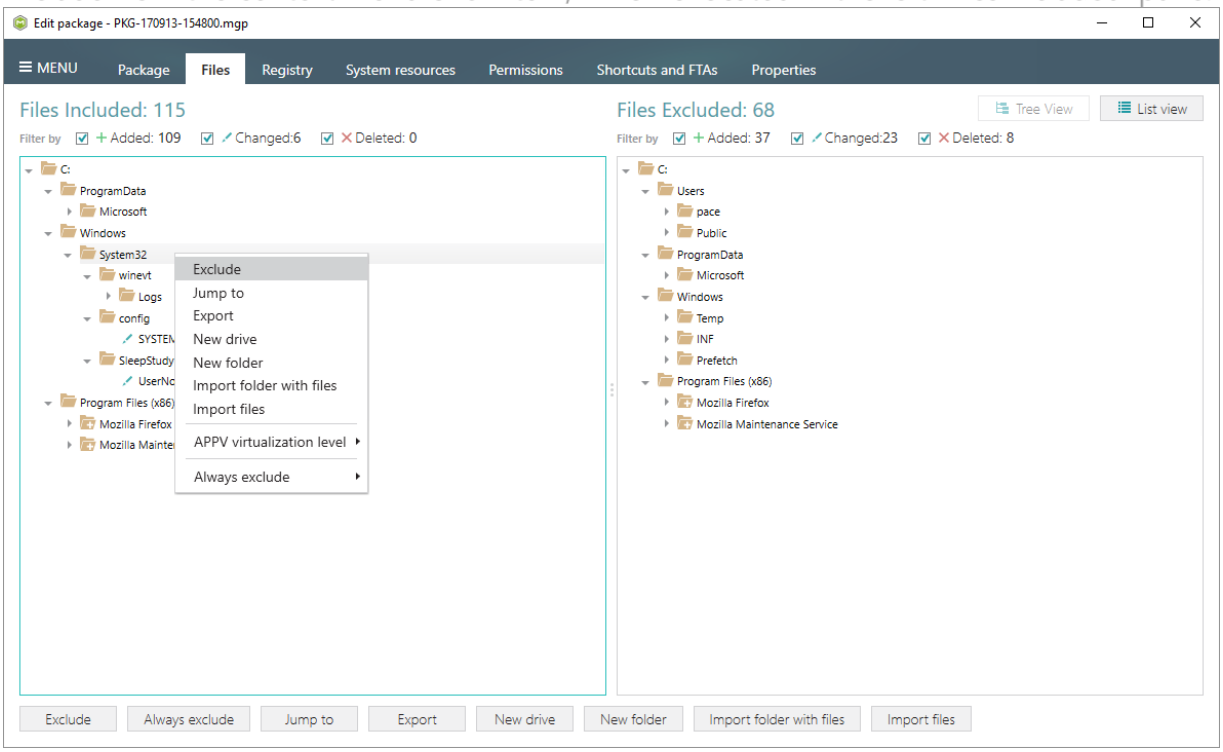

- Click Import APPV package.



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

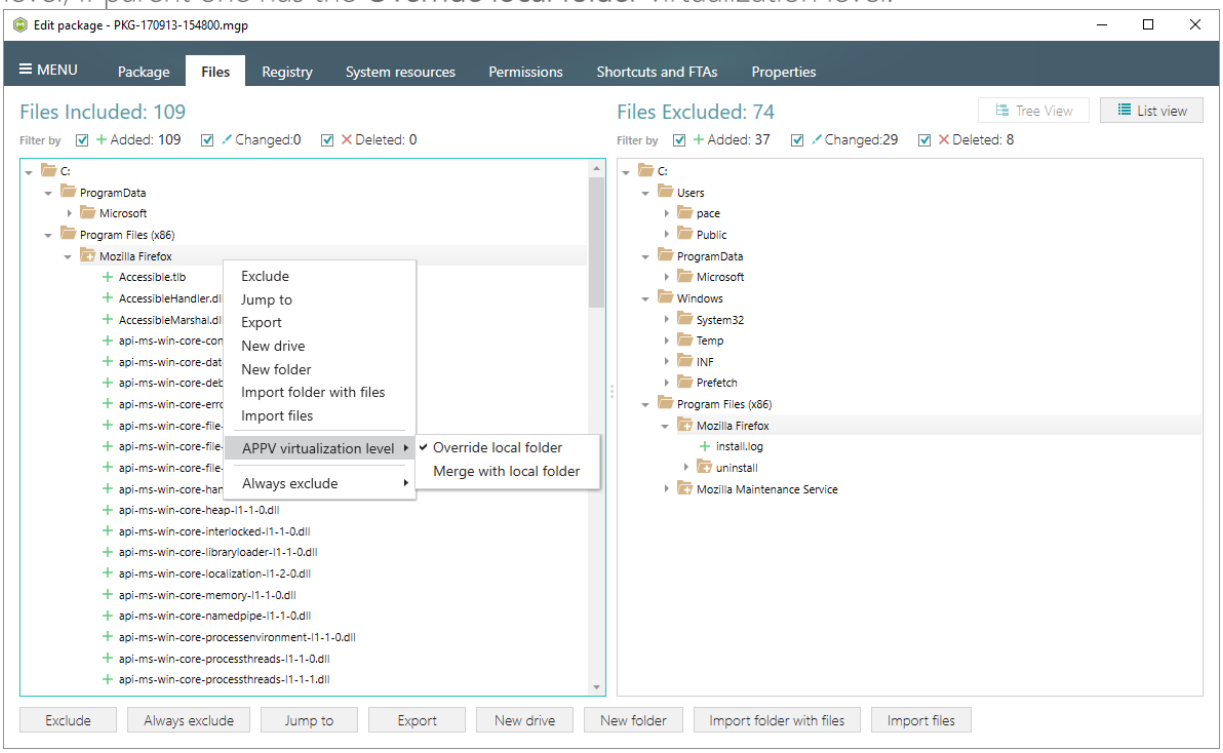



- In order to review and exclude unnecessary files or folders, go to the **Files** tab, and select **Exclude** from the context menu of an item, which is located in the left '**Files Included**' pane.

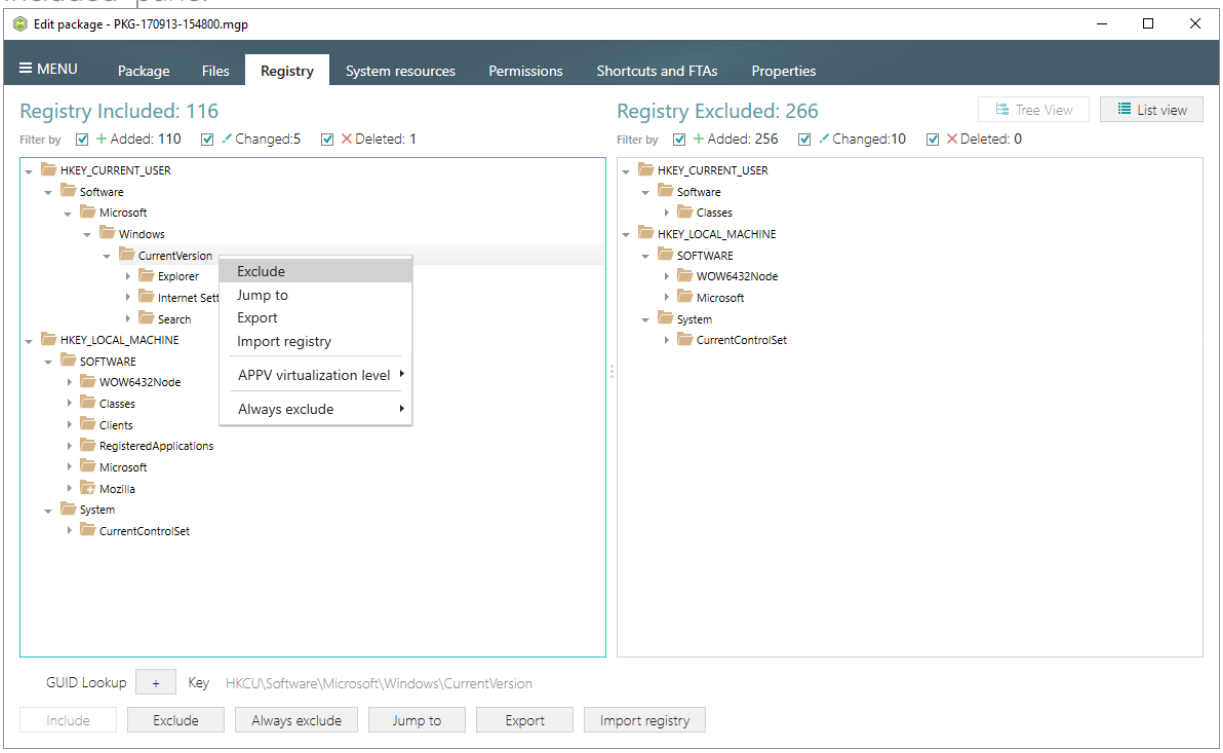



- In order to change the default virtualization (isolation) level of a folder, open the folder context menu, and select either **Override local folder** or **Merge with local folder** option from the **APPV virtualization level** submenu. Note that if the **APPV virtualization level** submenu is disabled with the **Override local folder** option selected, it means that this option is inherited from the parent folder. Child folder could not have the **Merge with local folder** virtualization

level, if parent one has the Override local folder virtualization level.

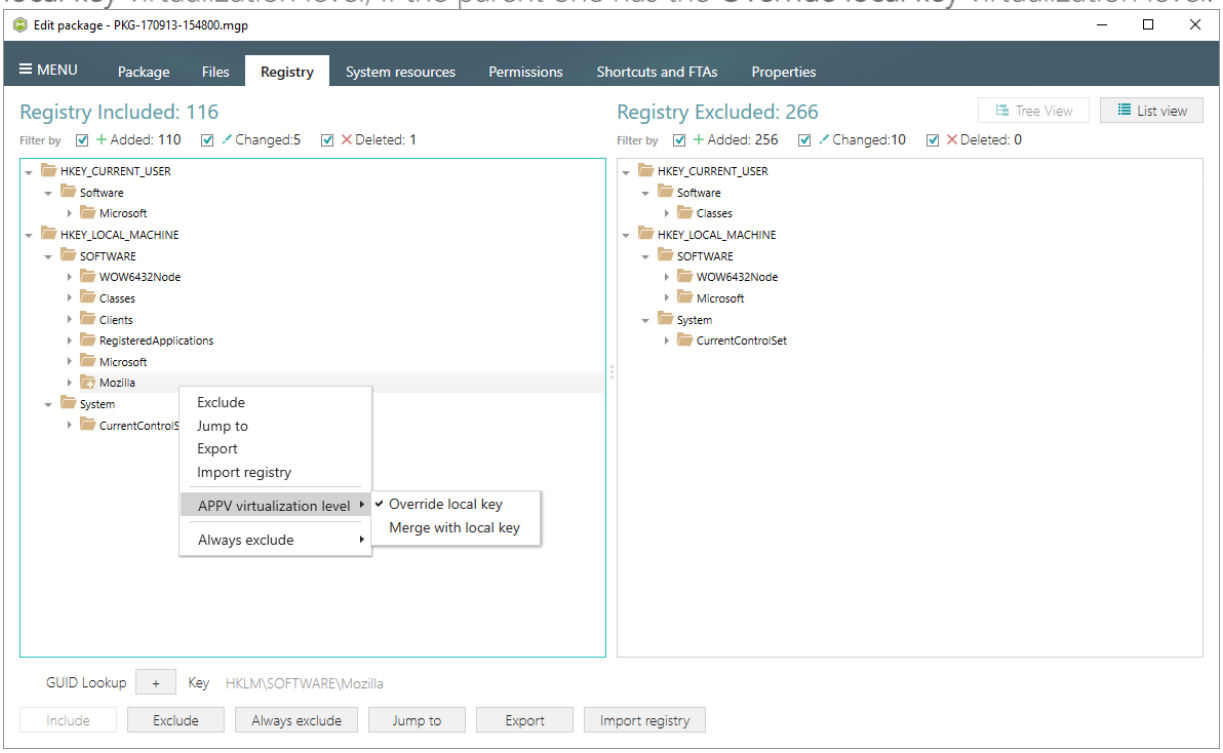



- In order to review and exclude unnecessary registry keys or values, go to the Registry tab, and select Exclude from the context menu of an item, which is located in the left 'Registry Included' pane.

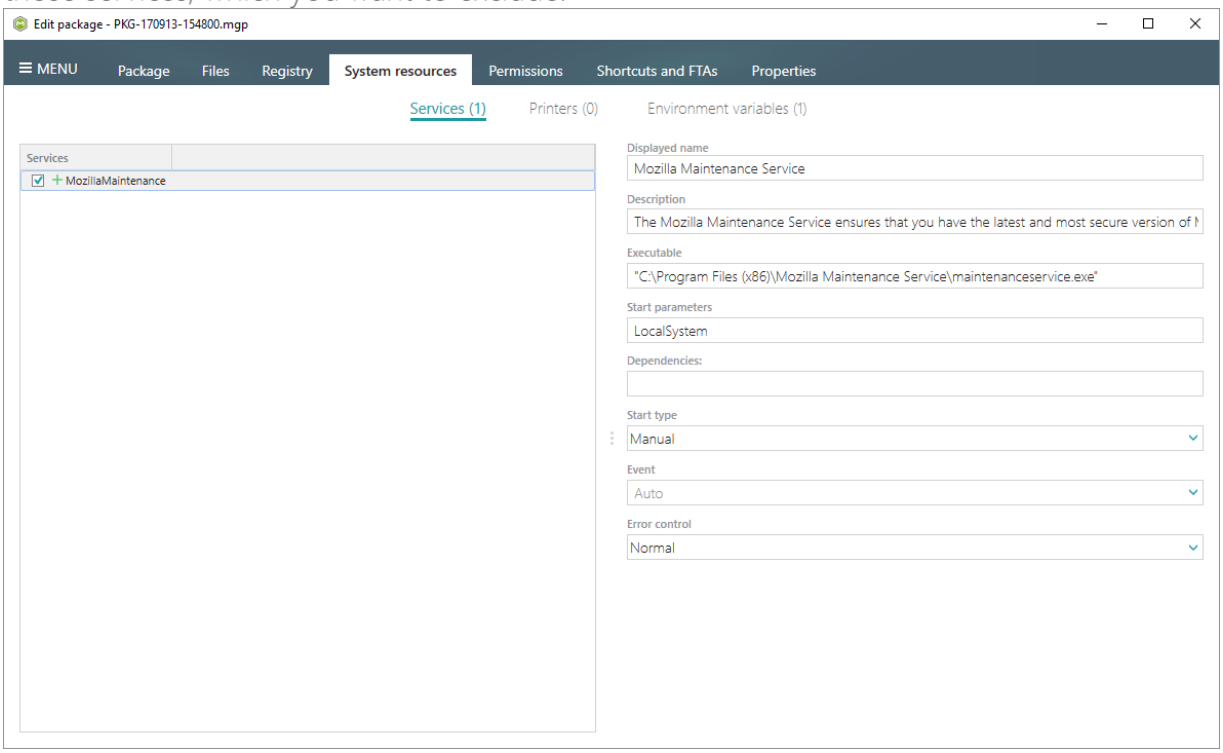



- In order to change the default virtualization (isolation) level of a registry key, open the registry key context menu, and select either Override local key or Merge with local key option from the APPV virtualization level submenu. Note that if the APPV virtualization level submenu is disabled with the Override local key option selected, it means that this option is inherited from the parent registry key. A child registry key could not have the Merge with

local key virtualization level, if the parent one has the Override local key virtualization level.

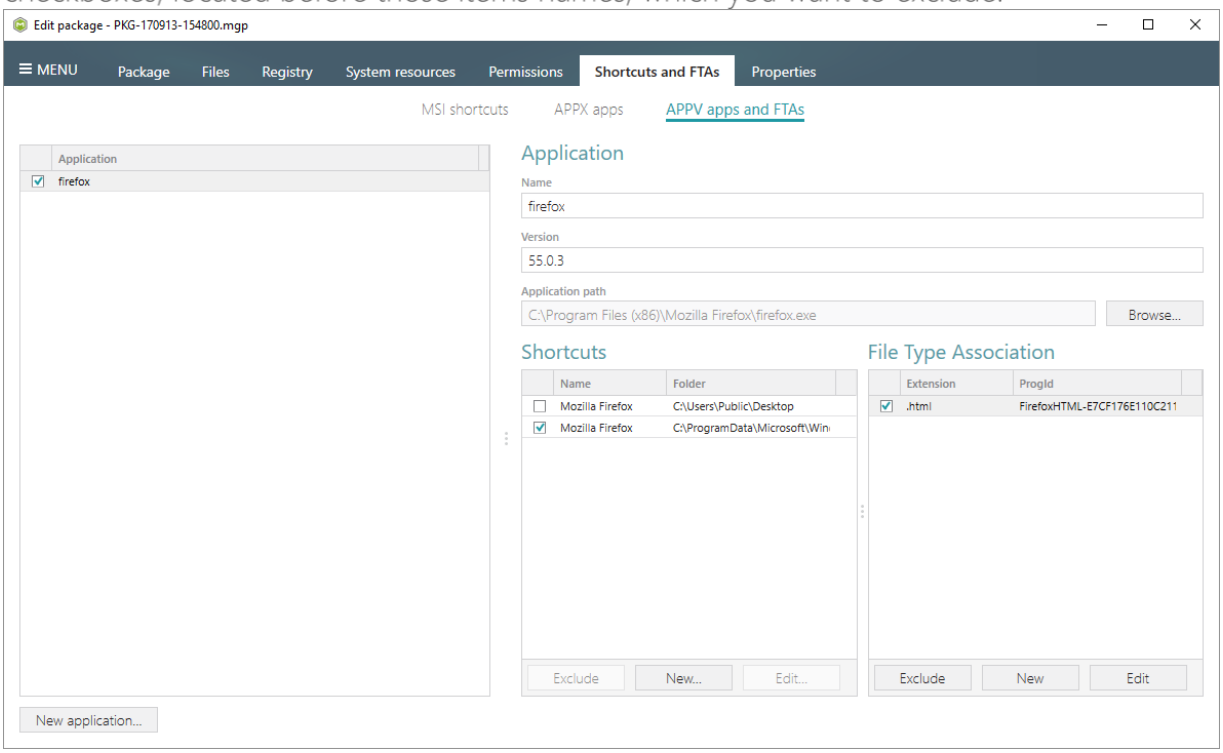



- In order to review and exclude unnecessary services, go to the System resources -> Services tab, and uncheck the checkbox, located before the service name in the list, for those services, which you want to exclude.

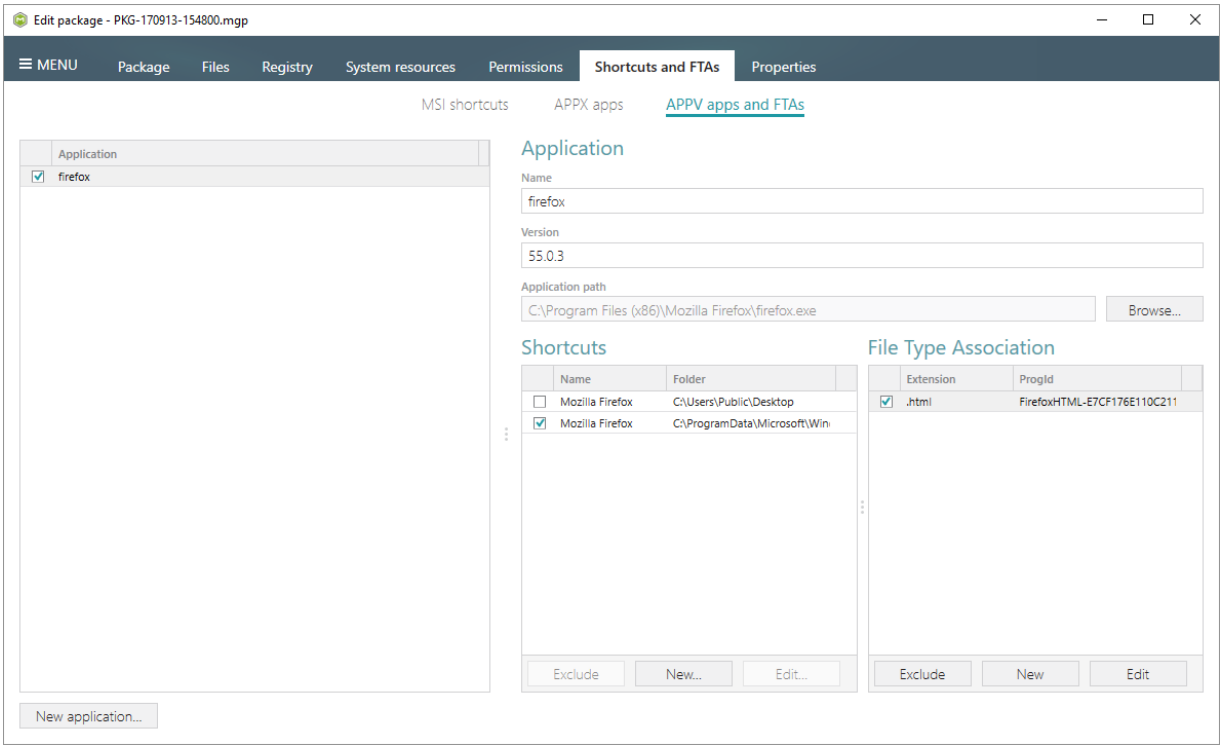



- In order to review and exclude unnecessary APPV applications, shortcuts and their file type associations, go to the Shortcuts and FTAs -> APPV apps and FTAs tab, and uncheck

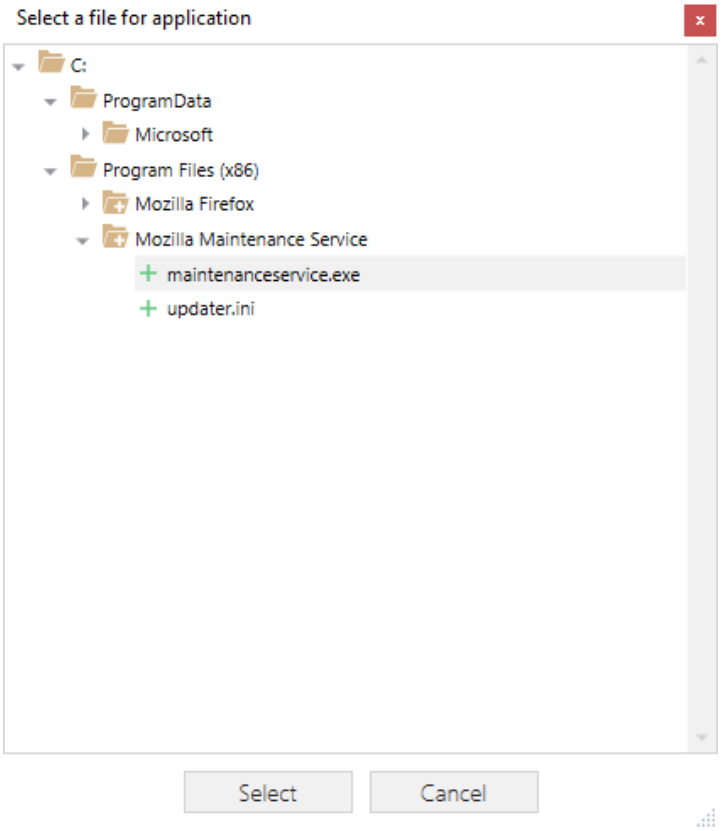
checkboxes, located before those items names, which you want to exclude.



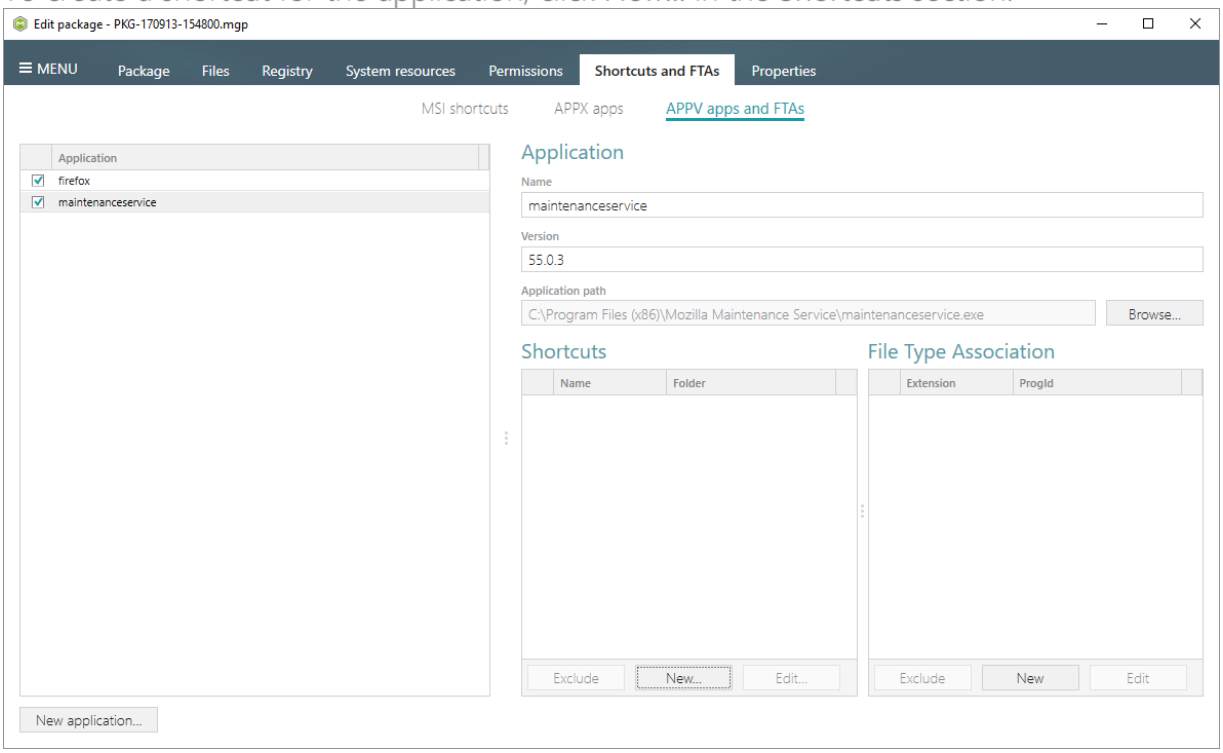

- If you need to add a new APPV application, its shortcuts and associate file extensions with this application, click **New application...** on the **Shortcuts and FTAs -> APPV apps and FTAs** tab.



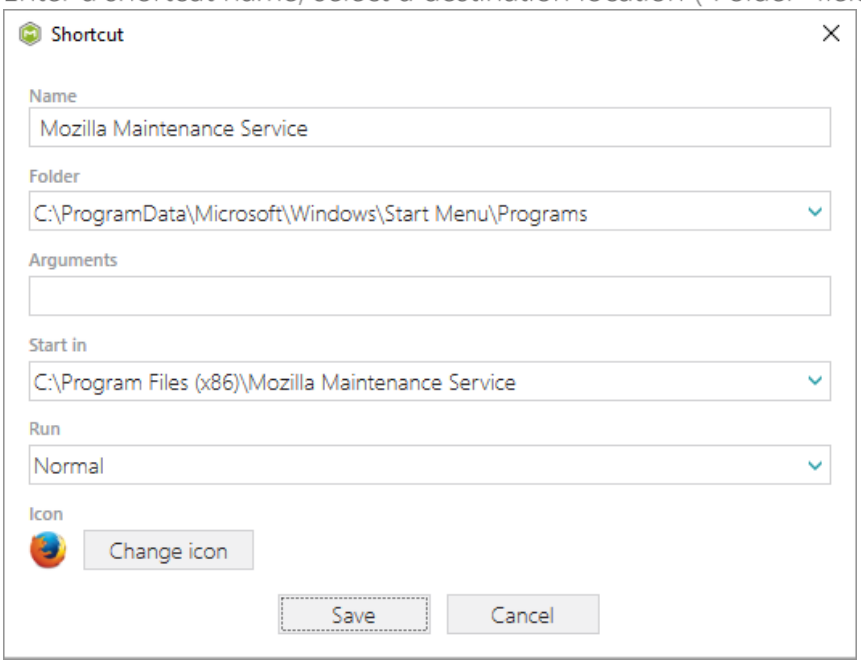
- Select the executable file (EXE) from the tree and click **Select**.



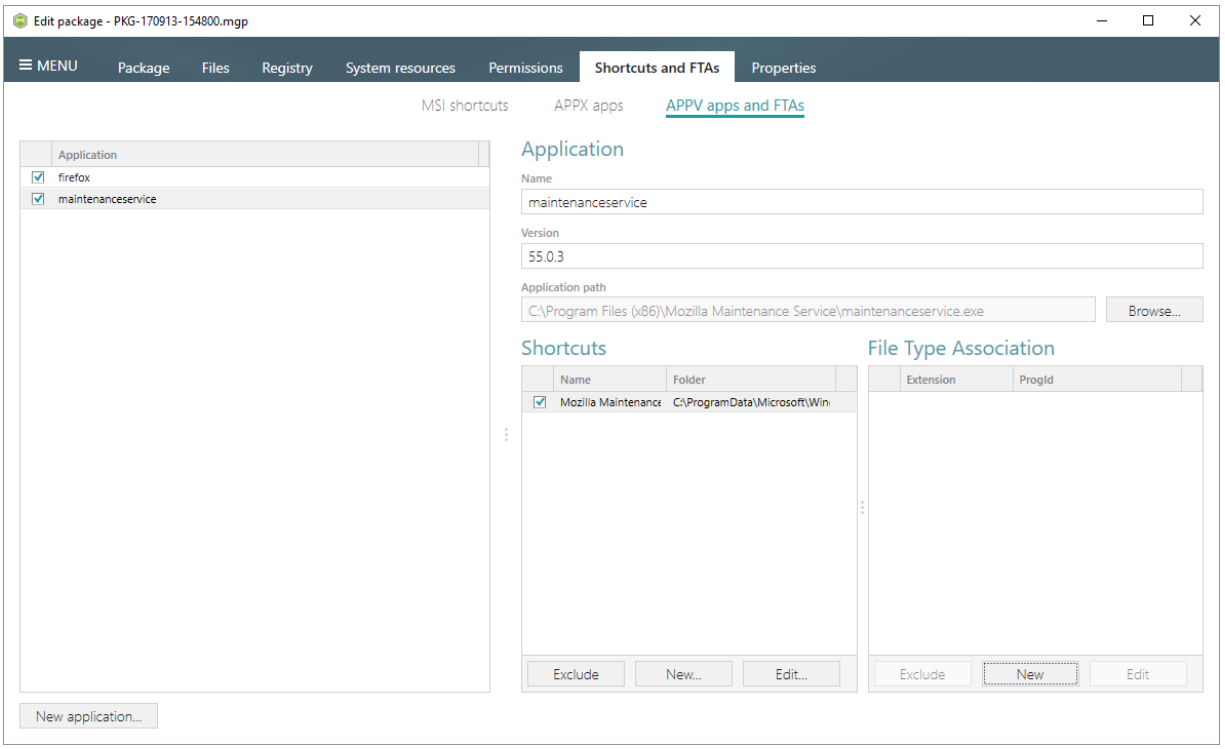

- To create a shortcut for the application, click **New...** in the **Shortcuts** section.



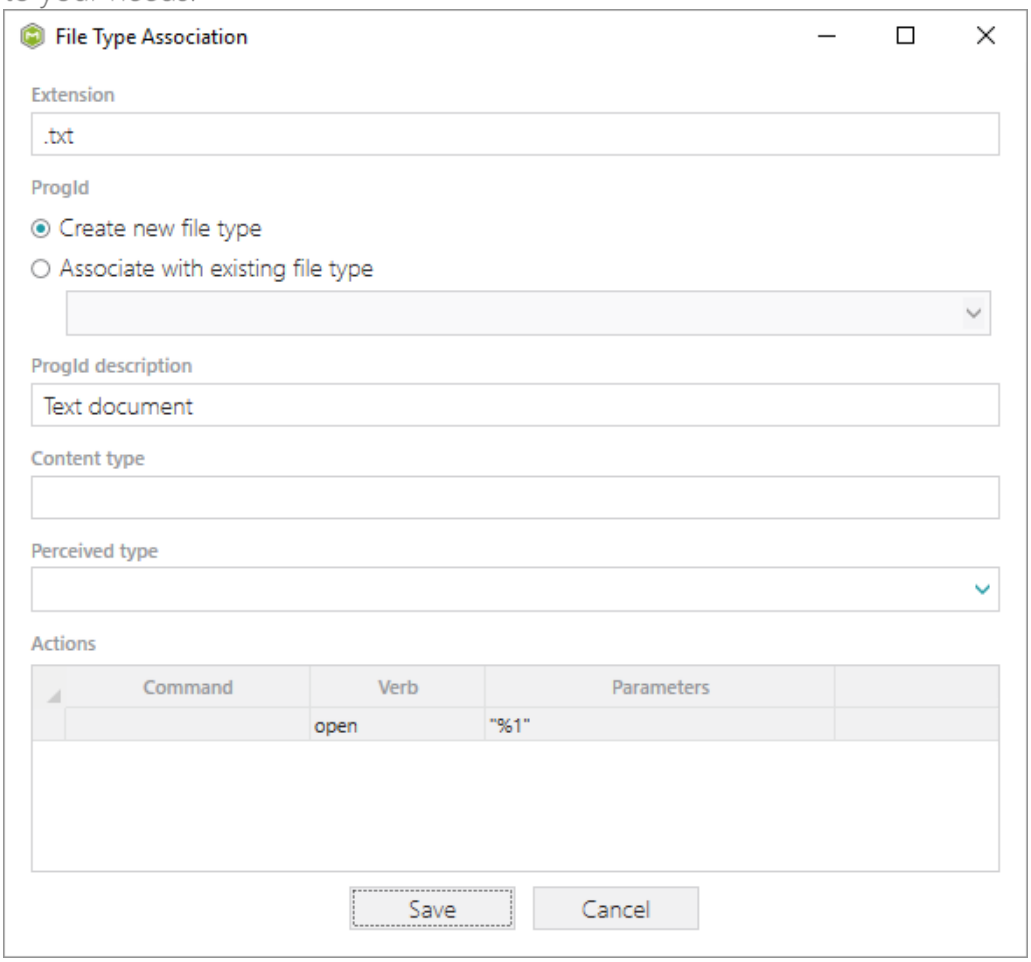
- Enter a shortcut name, select a destination location ("Folder" field), icon and click **Save**.



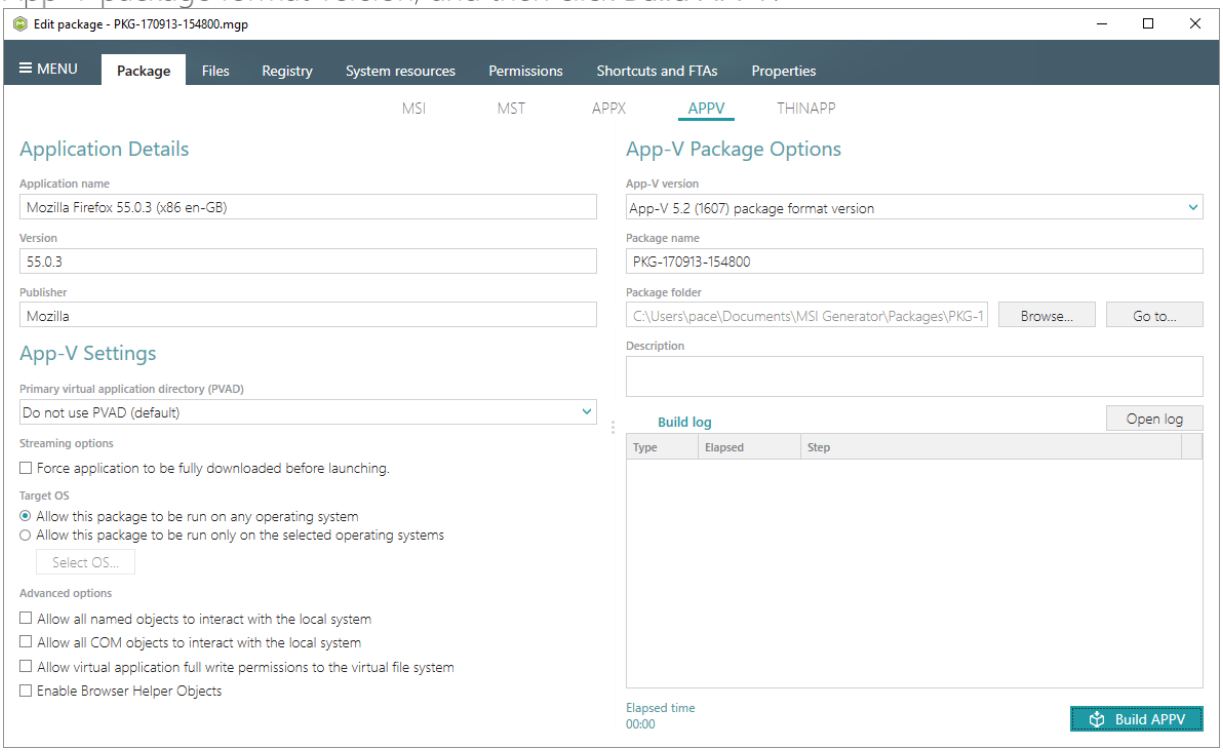

- To associate some file extension with your application, click **New...** in the **File Type Association** section.



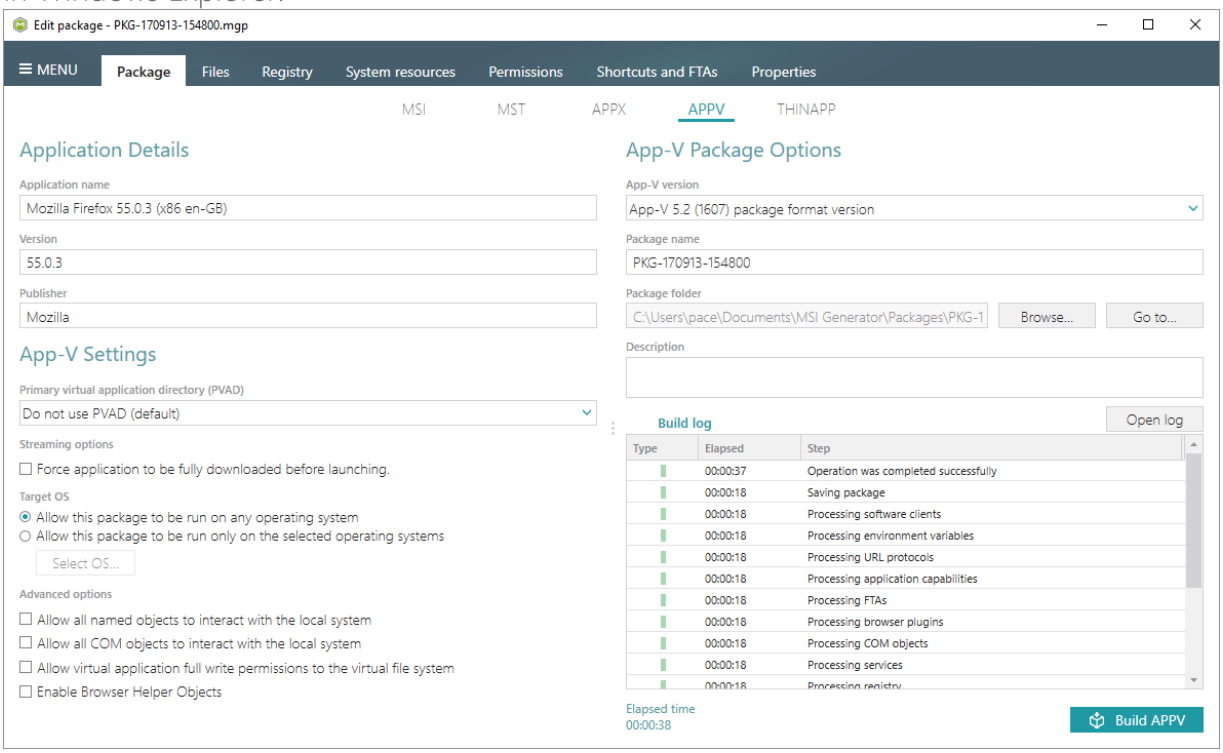
- Enter a file extension and click **Save**. Optinally you can modify other FTA settings according to your needs.




- Finaly, to build APPV package from the project, navigate to the **Package** -> **APPV** tab, review and update (if necessary) **Application Details**, select the necessary **App-V Settings**, App-V package format version, and then click **Build APPV**.



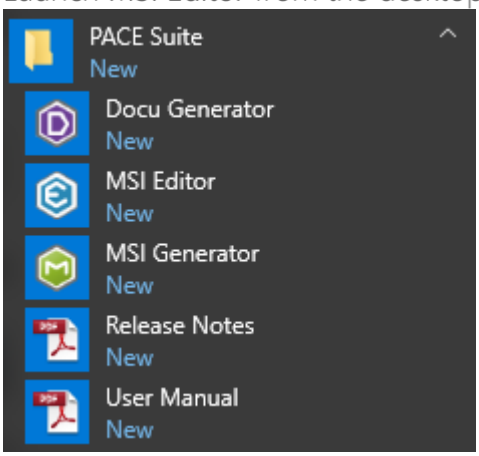
- Click **Go to...**, located next to the **Project folder** field, to open the package containing folder in Windows Explorer.



## 3.10 Deploy (Publish) to SCCM

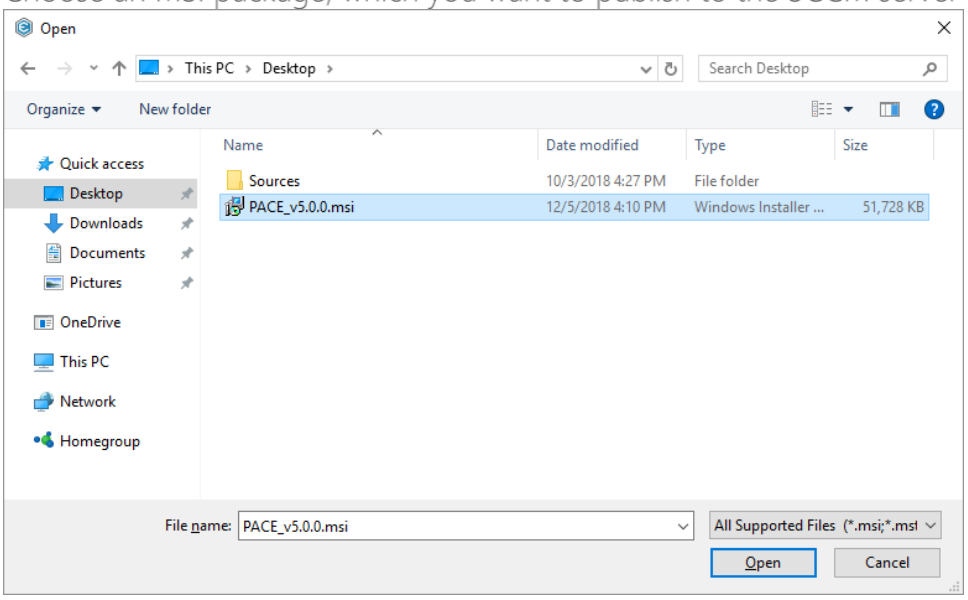
NOTE For publishing MSI packages to Microsoft SCCM 2007 (SP1/SP2/R1/R2) or SCCM 2012 (SP1/SP2/R2/R2 SP1) server, ensure that your system contains Microsoft Management Framework 3.0 and Windows Remote Management (WinRM) service enabled.

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

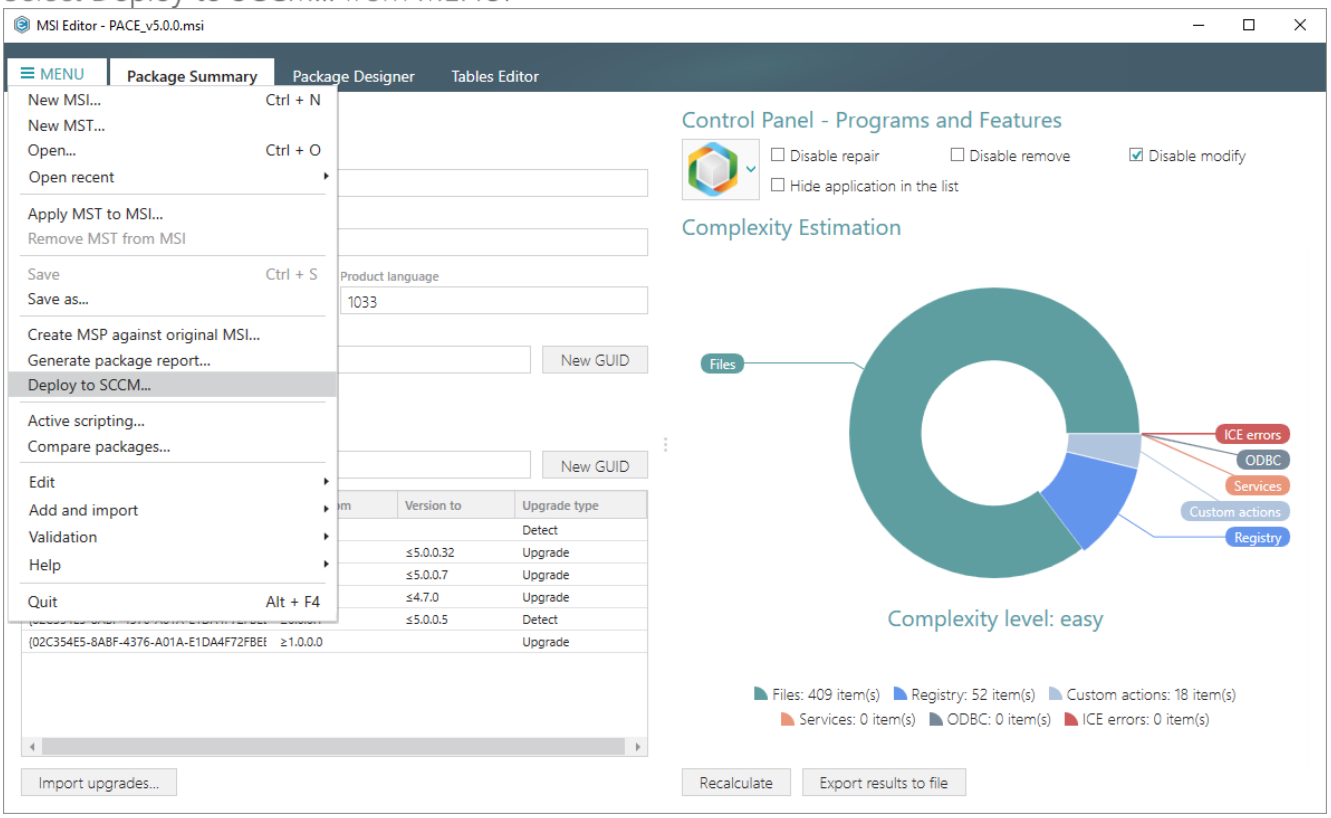



- Select Open... from MENU.

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

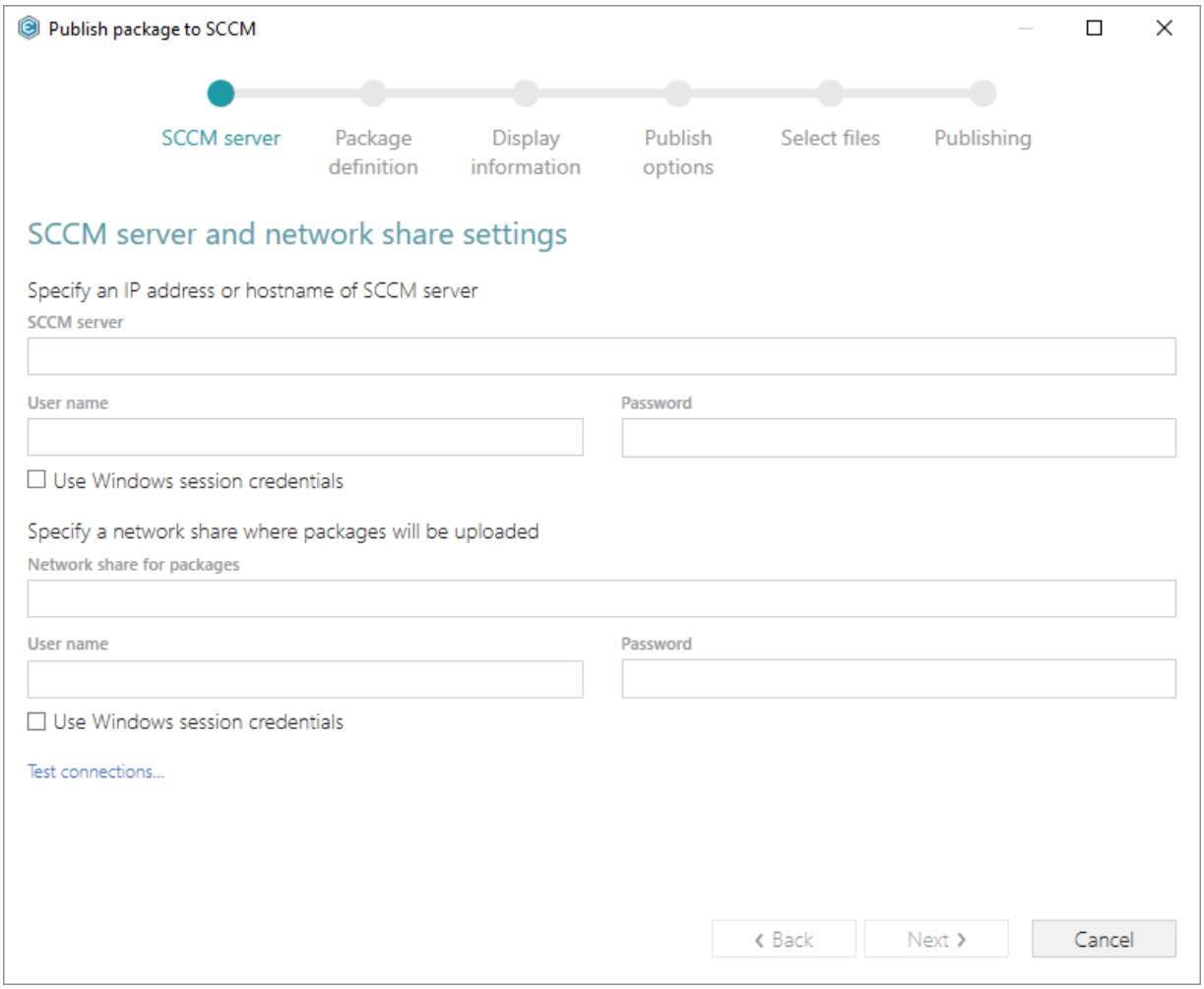



- Select **Deploy to SCCM...** from **MENU**.

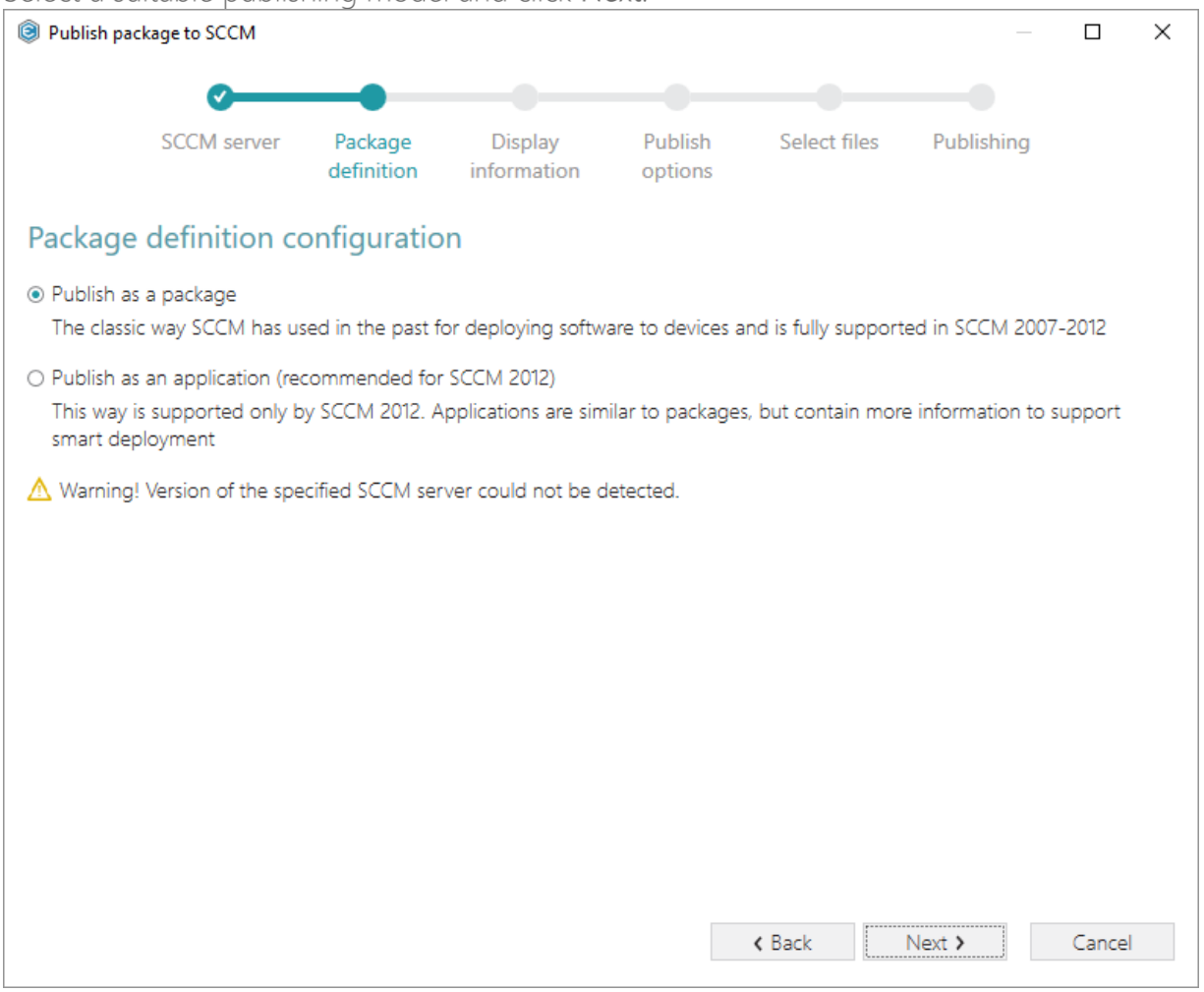



- Specify an IP address or a hostname of your SCCM server, user name and password for connecting to the SCCM. Enter a network share, where the package should be uploaded, and credentials for connecting to the network share. Click **Next** to choose a publishing model. Click **Test connections...** to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

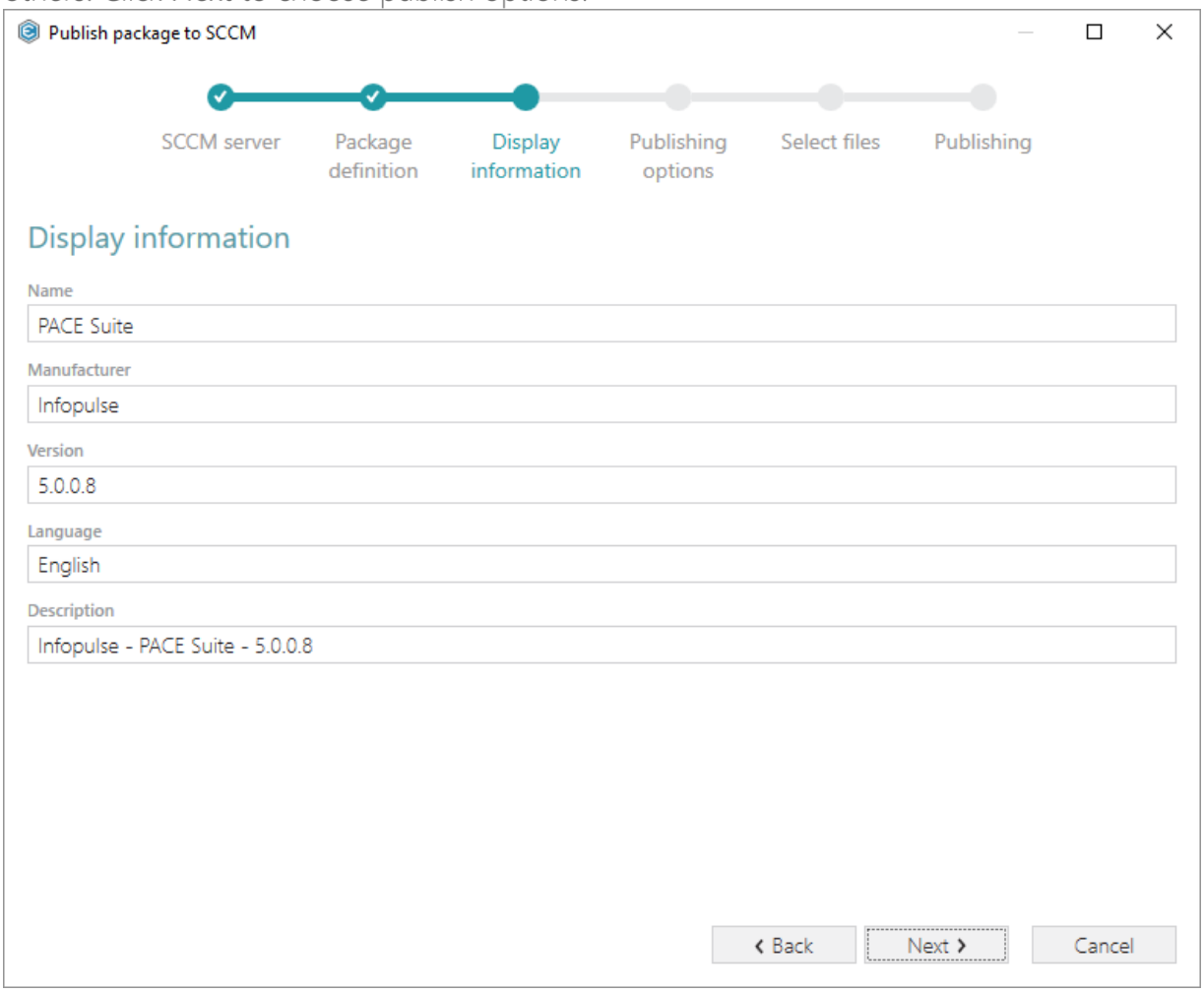
share.


Publish package to SCCM
SCCM server
Package definition
Display information
Publish options
Select files
Publishing
SCCM server and network share settings
Specify an IP address or hostname of SCCM server
SCCM server
User name
Password
Use Windows session credentials
Specify a network share where packages will be uploaded
Network share for packages
User name
Password
Use Windows session credentials
Test connections...
Back
Next
Cancel

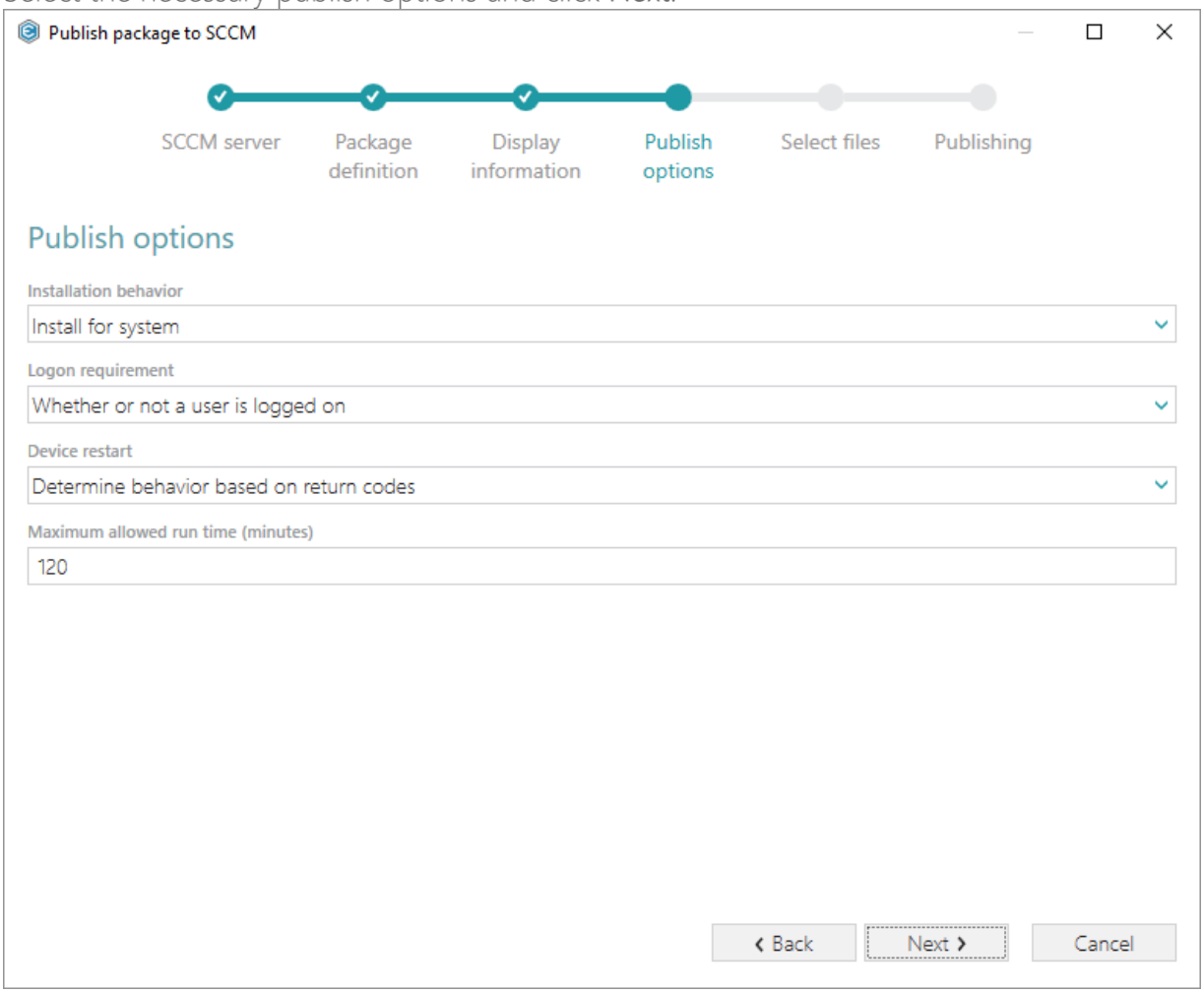
- Select a suitable publishing model and click **Next**.



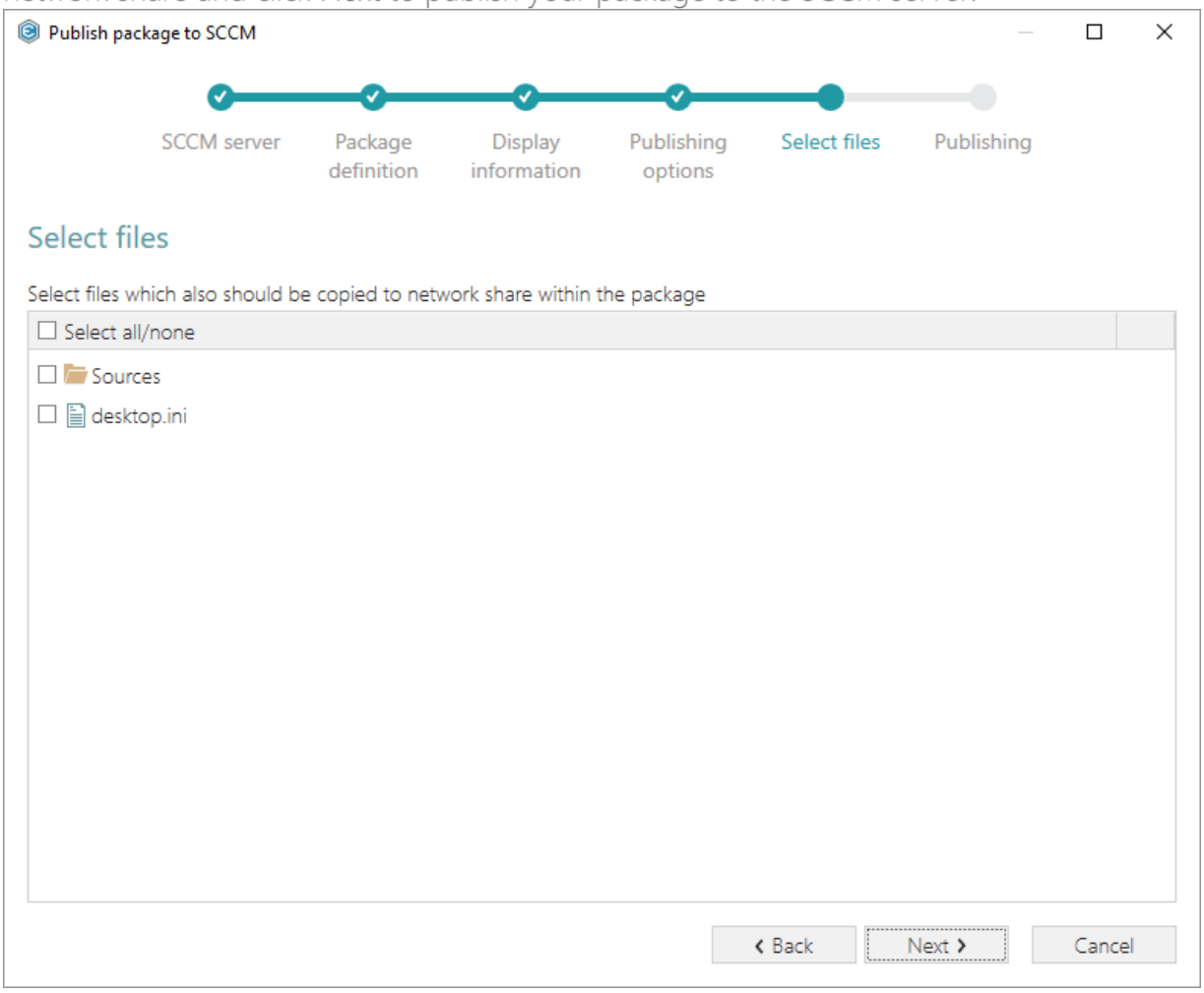
- Review and update package display information such as name, manufacturer, version and others. Click Next to choose publish options.



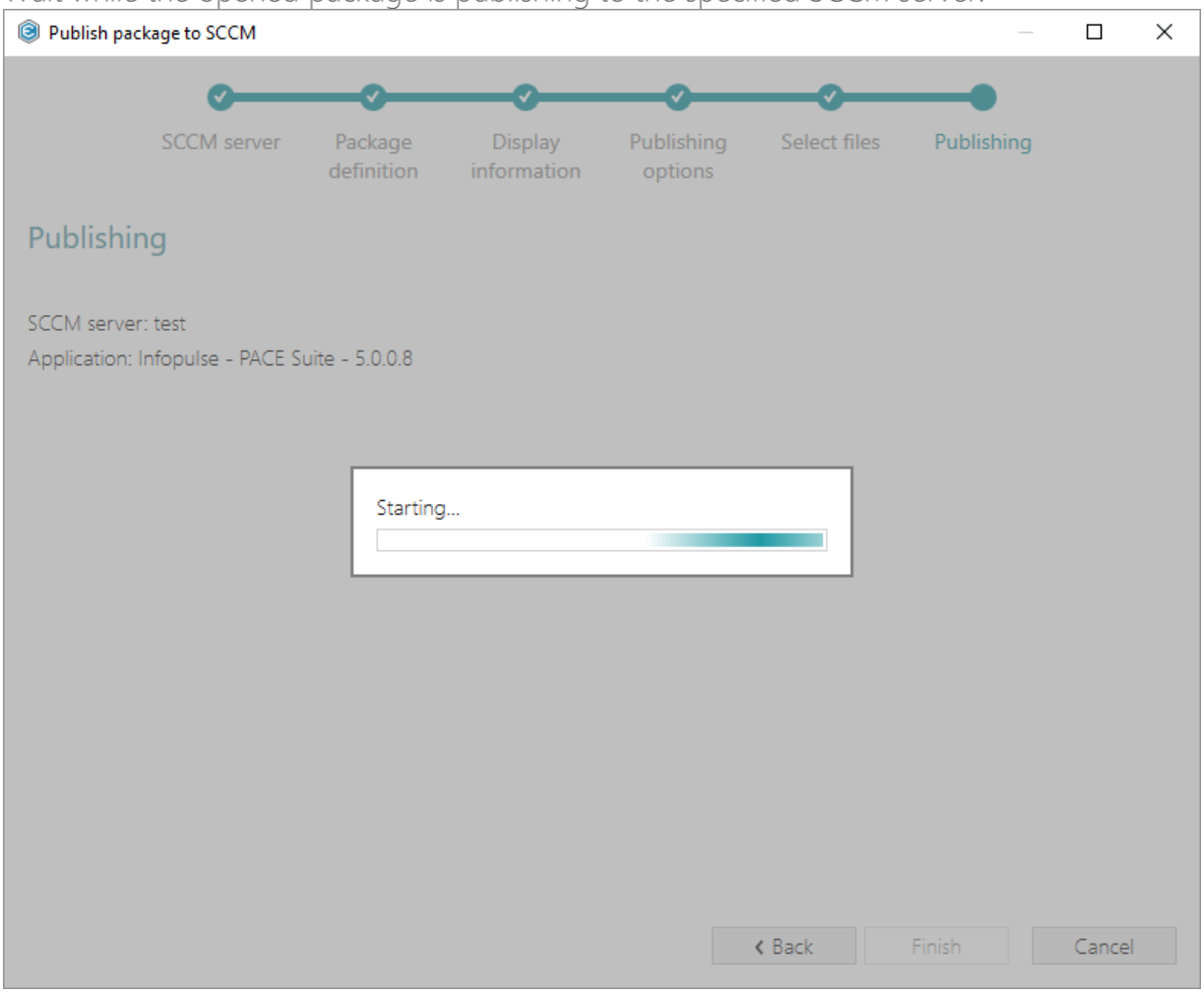
- Select the necessary publish options and click **Next**.



- 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.



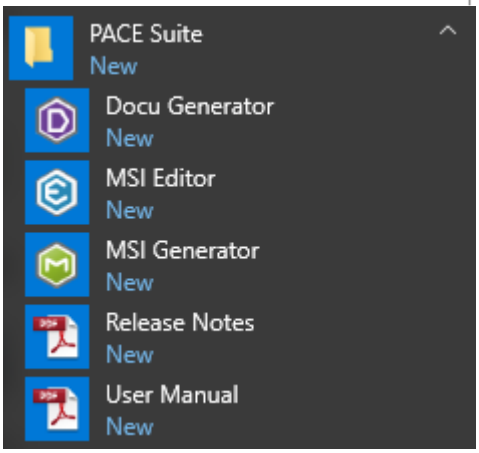
- Wait while the opened package is publishing to the specified SCCM server.




## 3.11 Generate Package Report

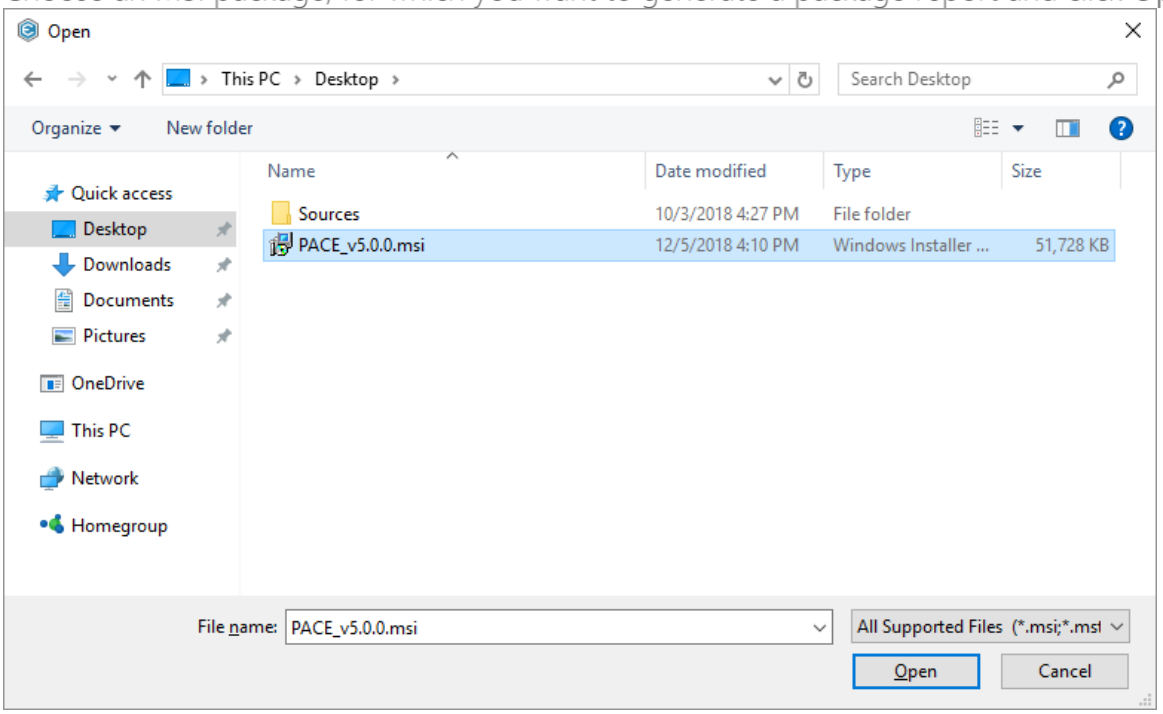
NOTE PACE Suite supports Word Document (.DOCX) and Excel Workbook (.XLSX) format types and allows generating a report with no Microsoft Office installed on the system.

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

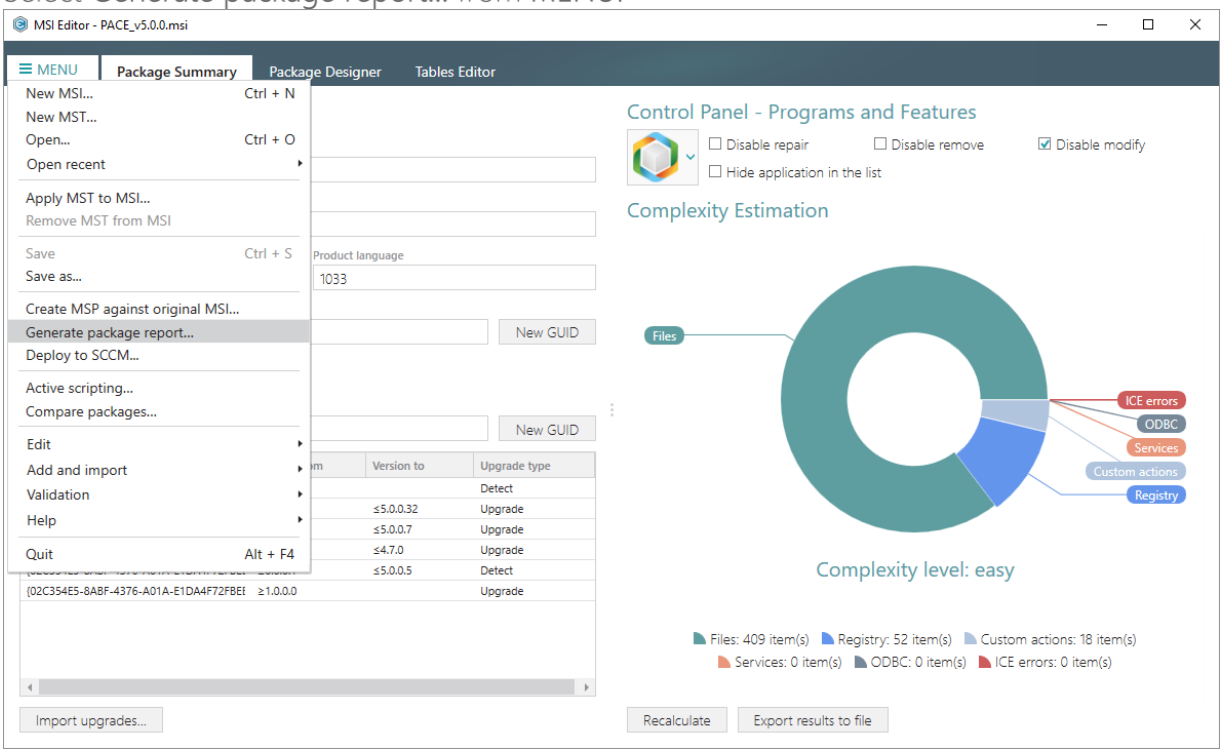


- Select Open... from MENU.

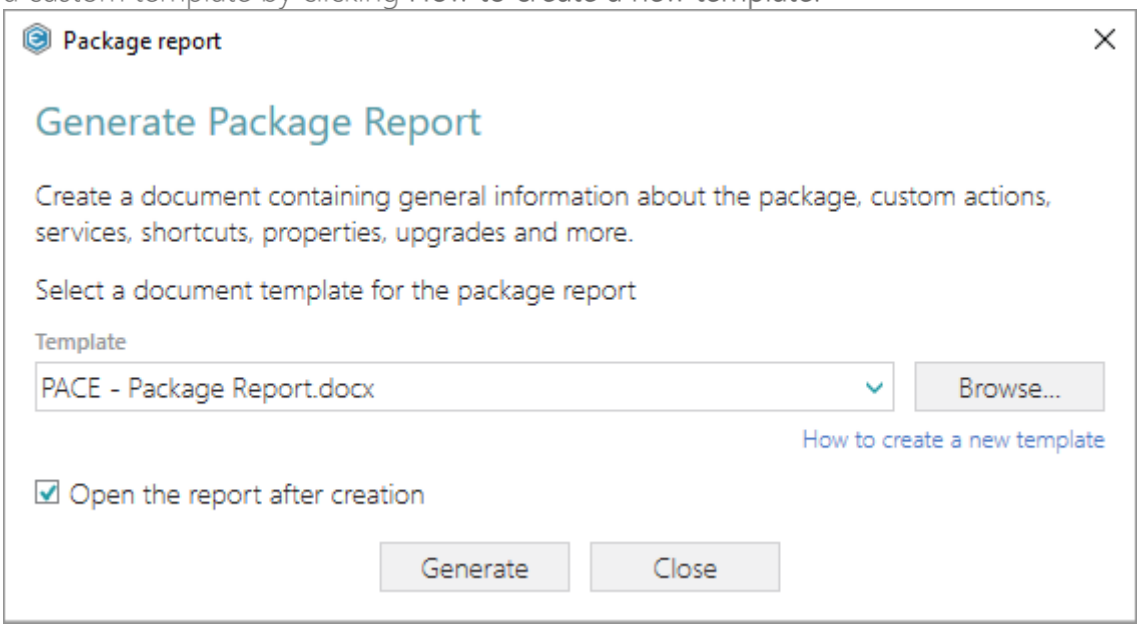
- Choose an MSI package, for which you want to generate a package report and click Open.



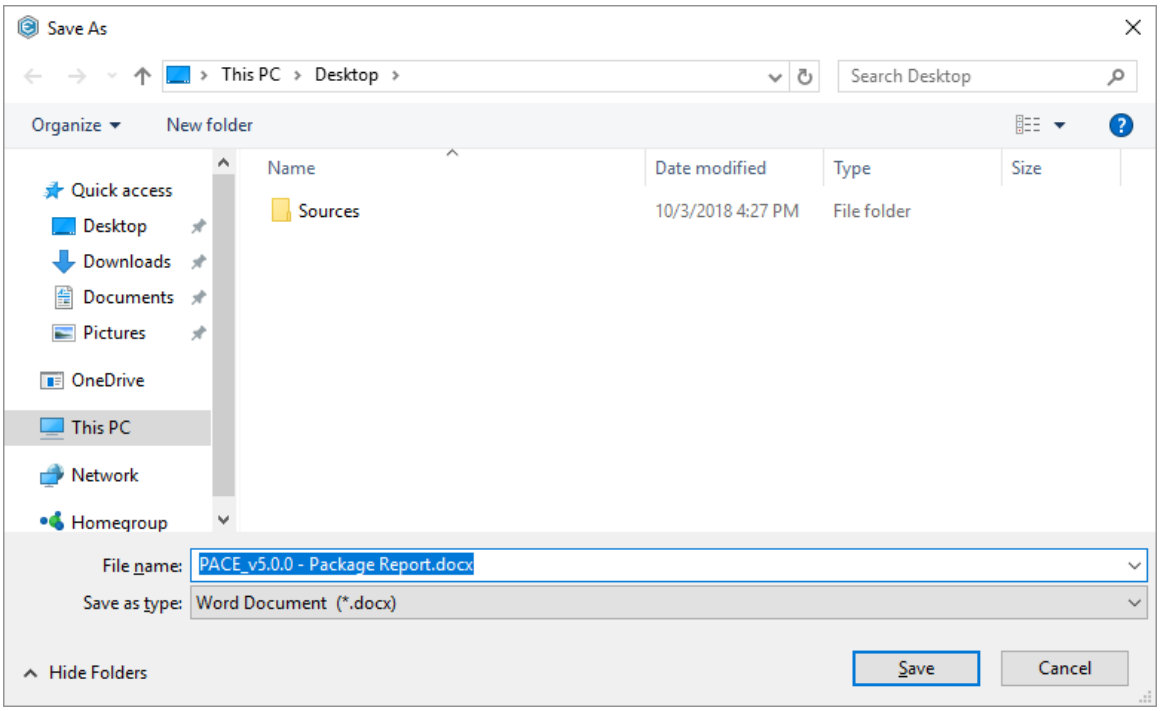
- Select Generate package report... from MENU.




- 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.

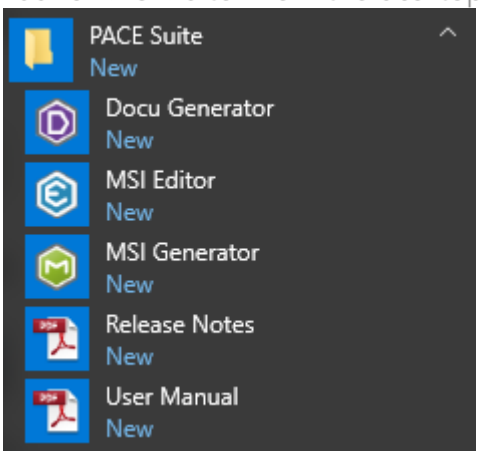


- Specify a file name and a destination location of the generated package report and click Save.



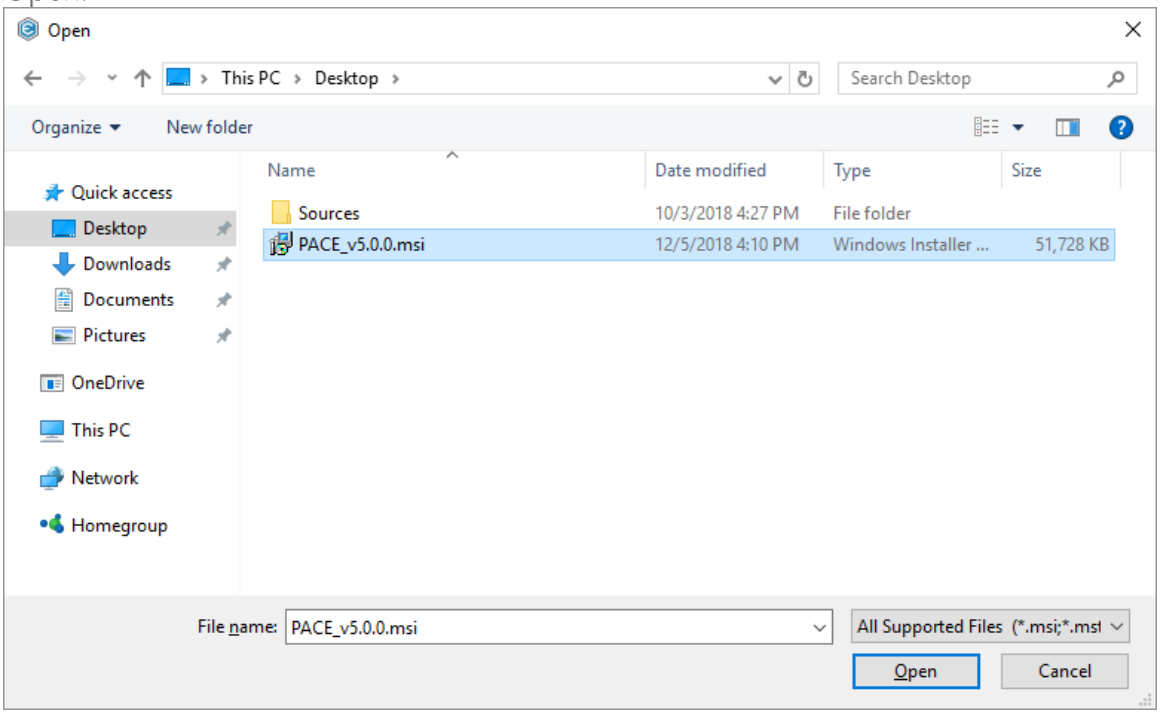

## 3.12 Validate Package

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

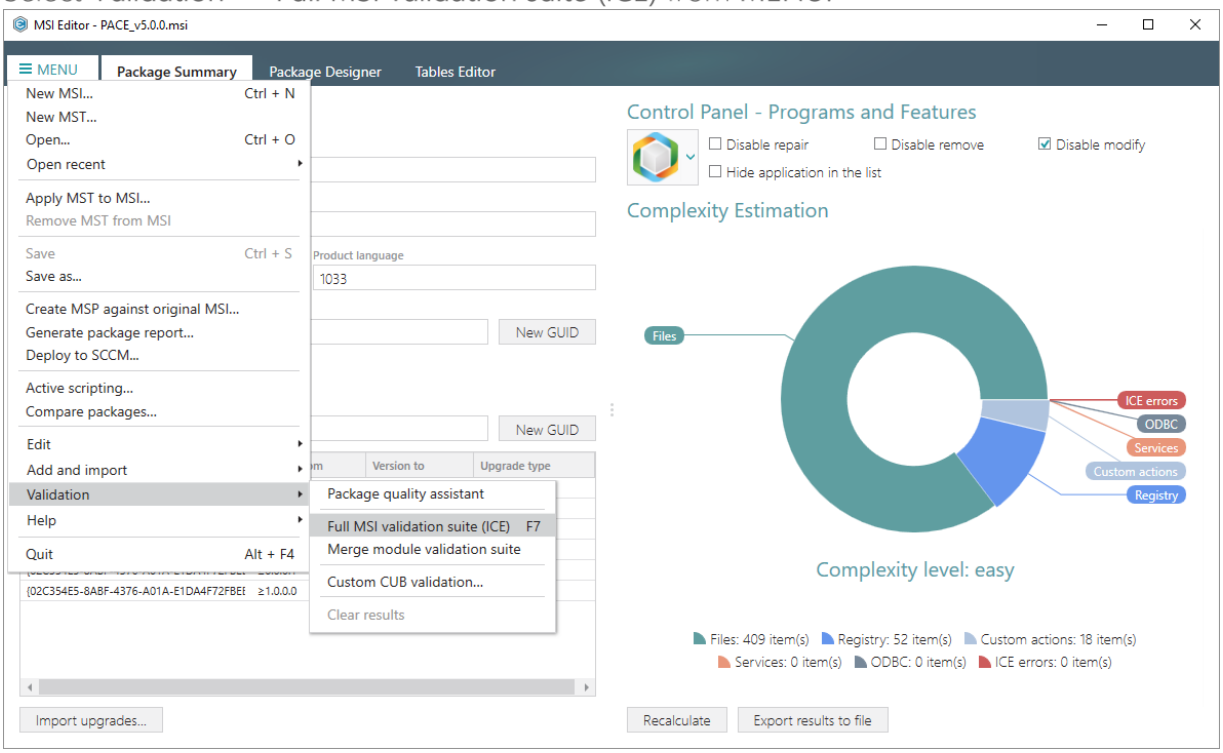


- Select Open... from MENU.

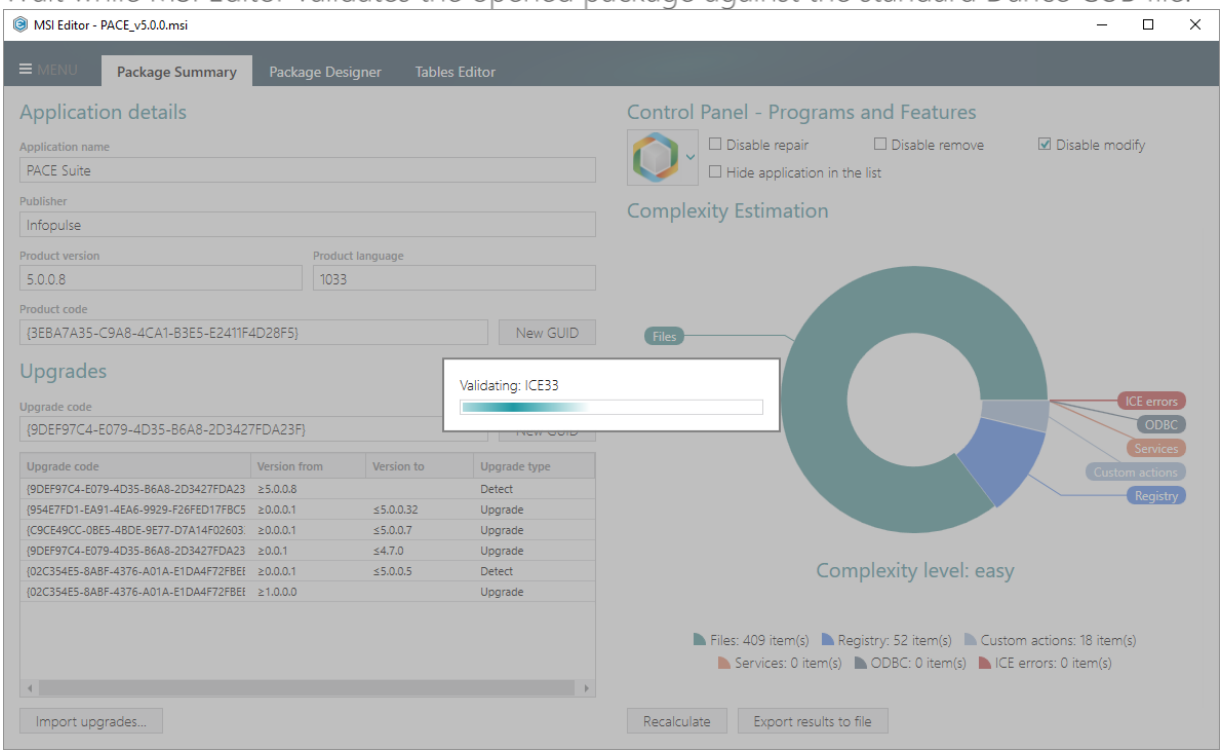
- Choose an MSI package, which you want to validate against the standard CUB files and click Open.



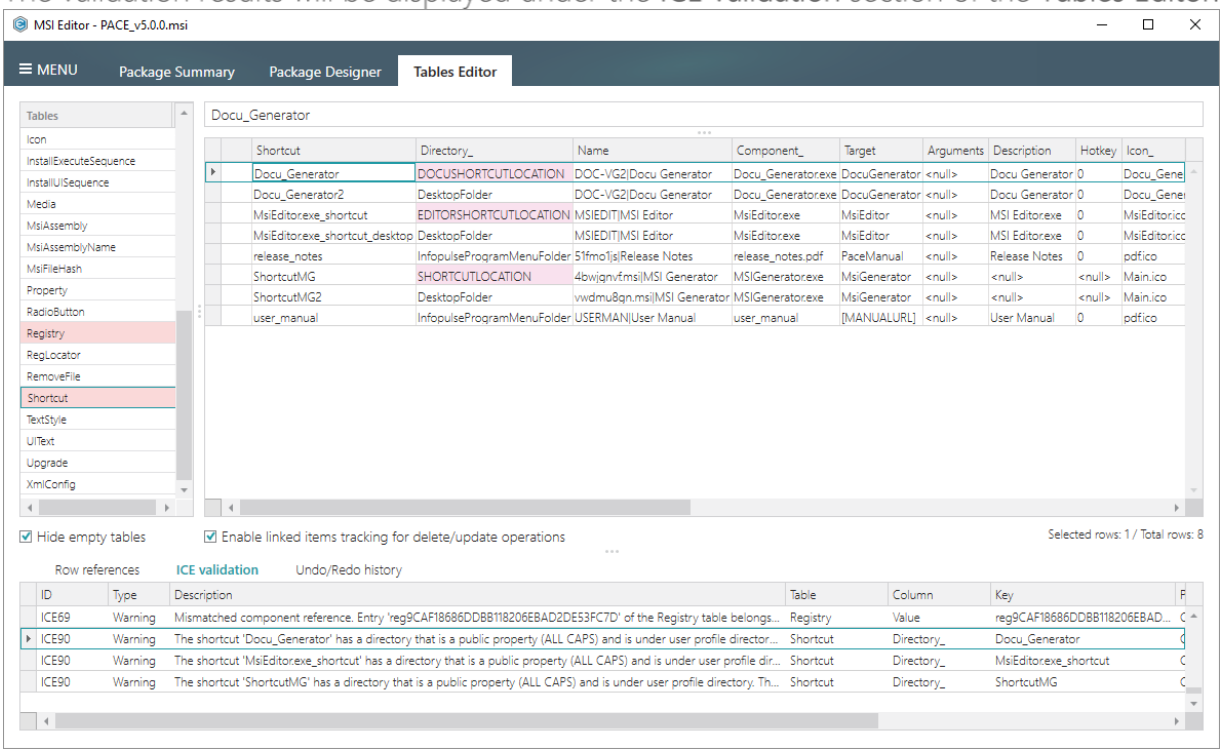
- Select Validation -> Full MSI validation suite (ICE) from MENU.




- Wait while MSI Editor validates the opened package against the standard Darice CUB file.

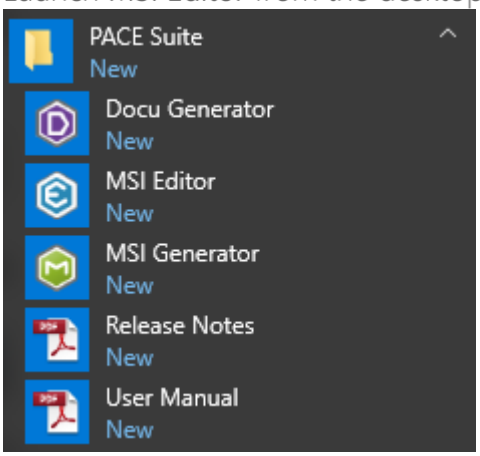


- The validation results will be displayed under the **ICE validation** section of the **Tables Editor**.



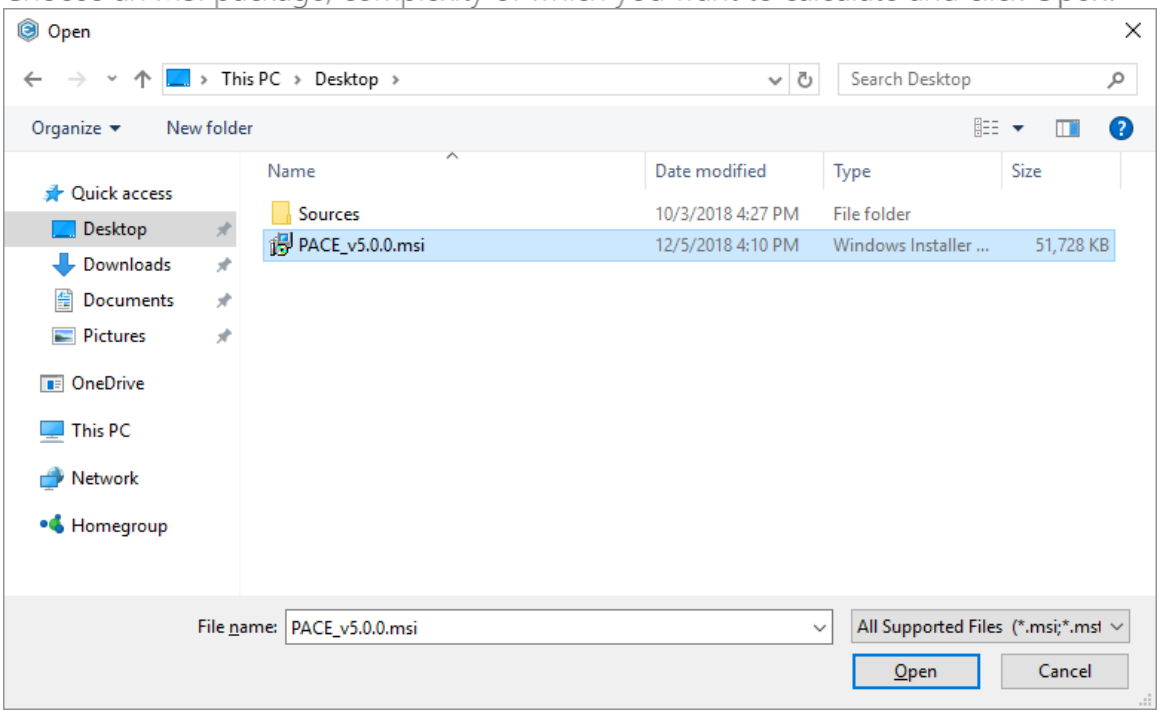

## 3.13 Get Package Complexity

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

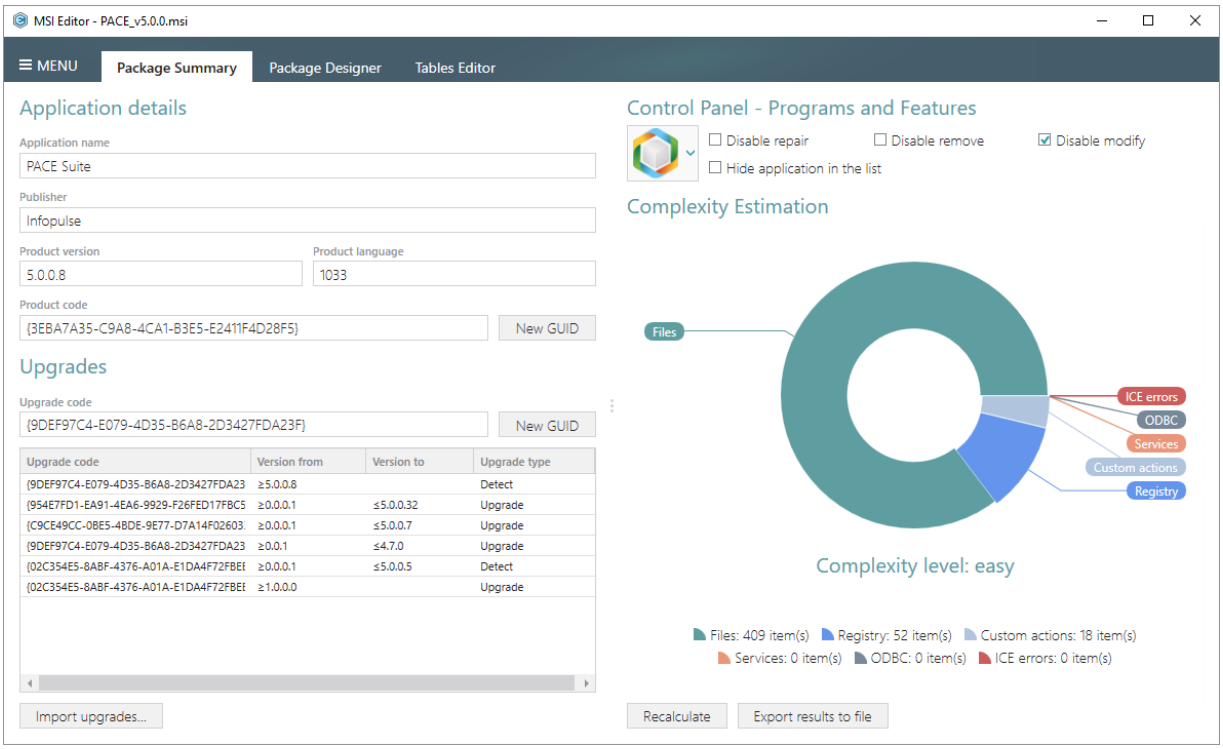


- Select Open... from MENU.

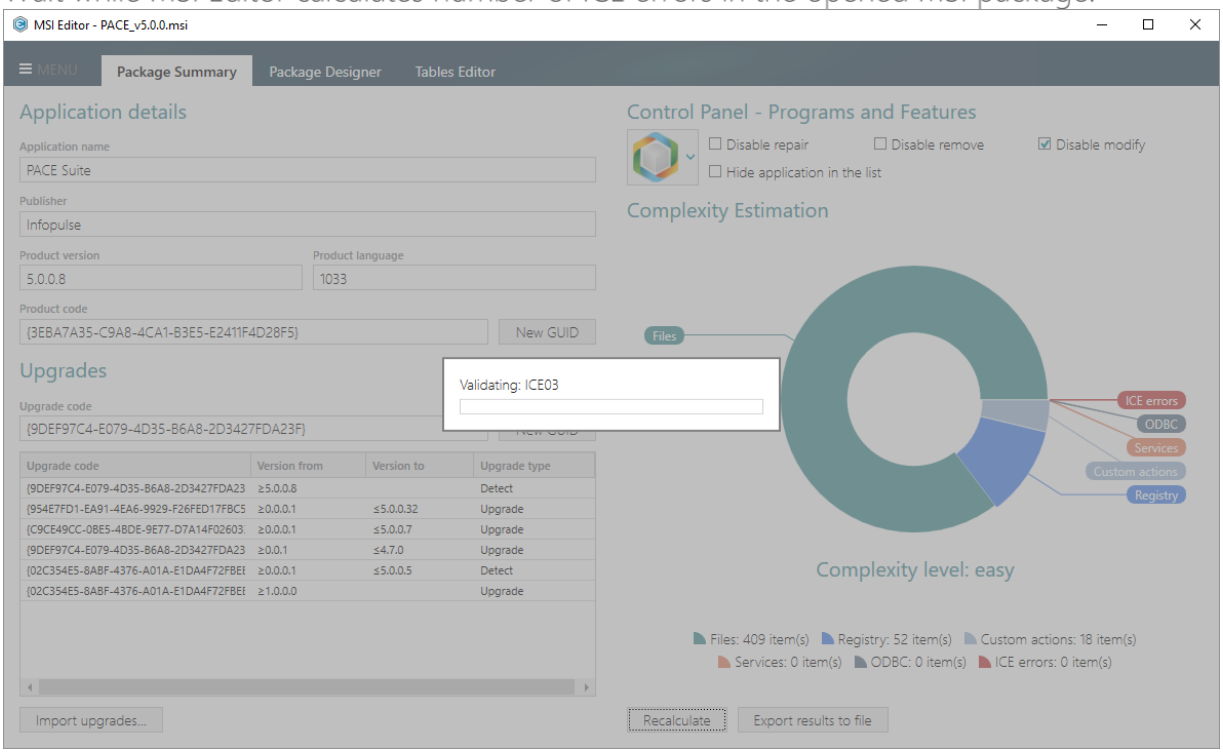
- Choose an MSI package, complexity of which you want to calculate and click Open.



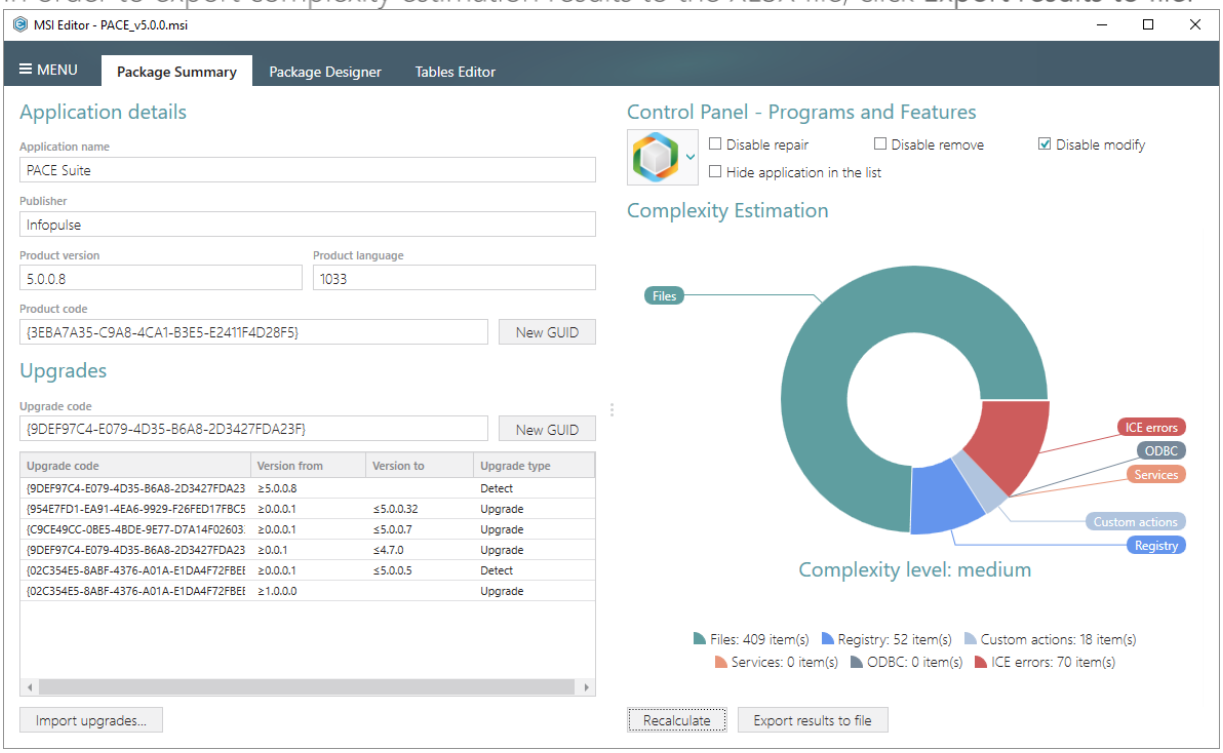
- Click **Recalculate**, located in the bottom, to re-calculate the complexity level including ICE validation errors.



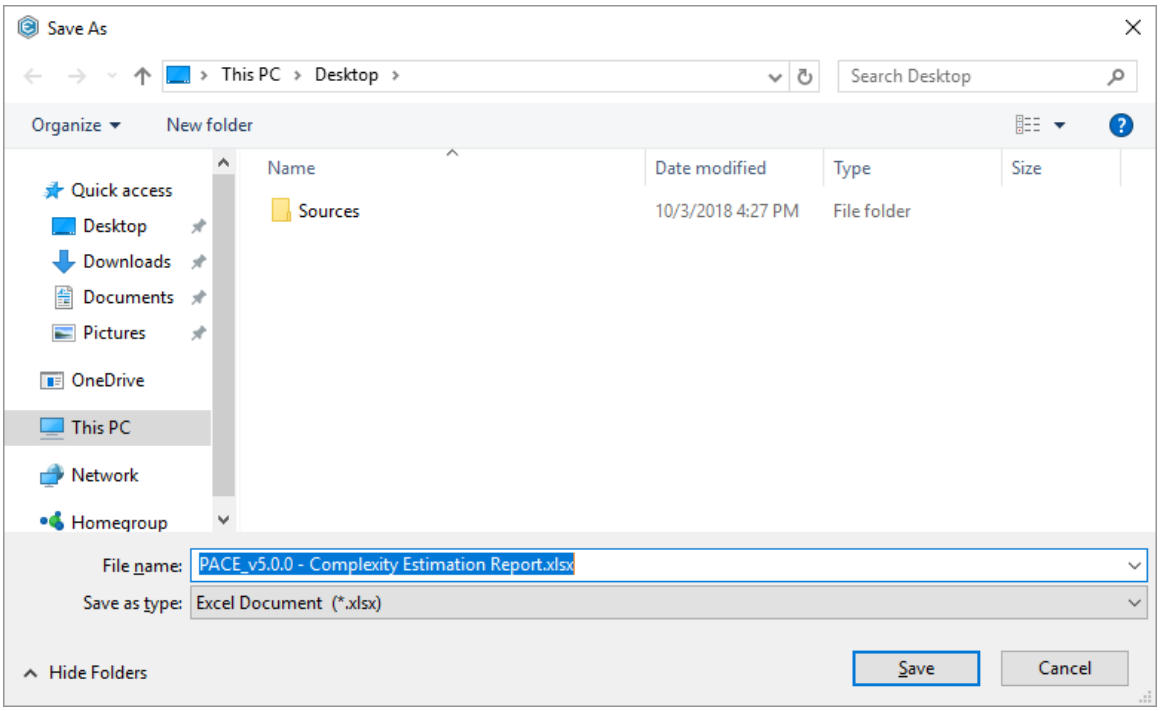

- Wait while MSI Editor calculates number of ICE errors in the opened MSI package.



- In order to export complexity estimation results to the XLSX file, click Export results to file.




- Specify a name and a destination location of the exported complexity estimation results and click Save.

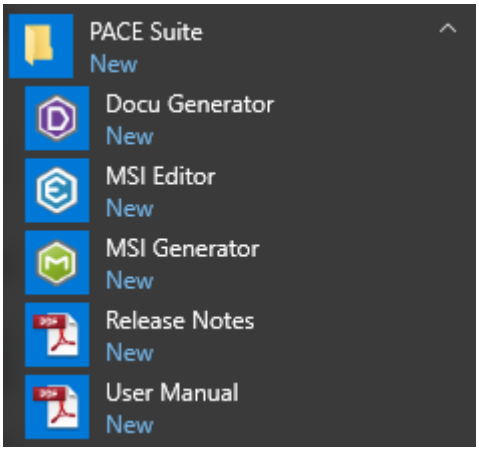


## 3.14 Create Discovery Documentation

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.

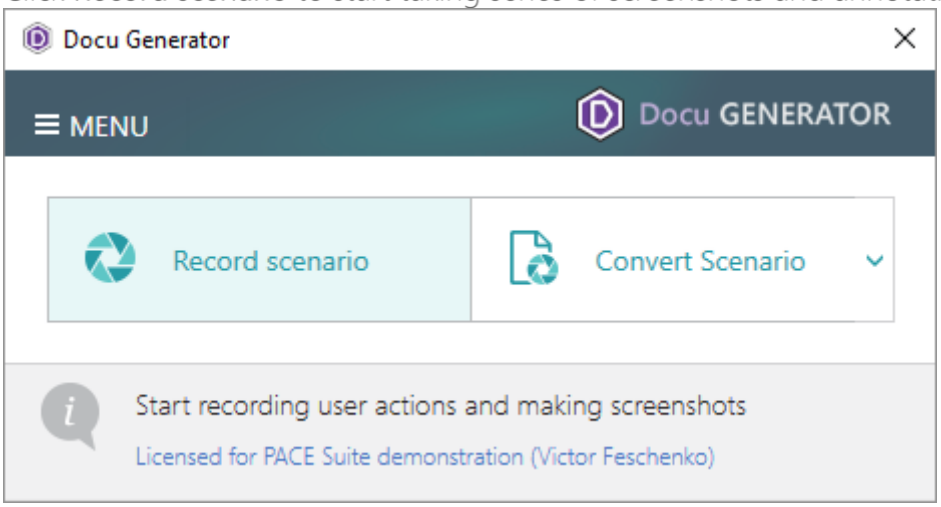
NOTE In order to take screenshots of application, which is launched with elevated permissions, launch Docu Generator with elevated permissions as well.

- Launch Docu Generator.



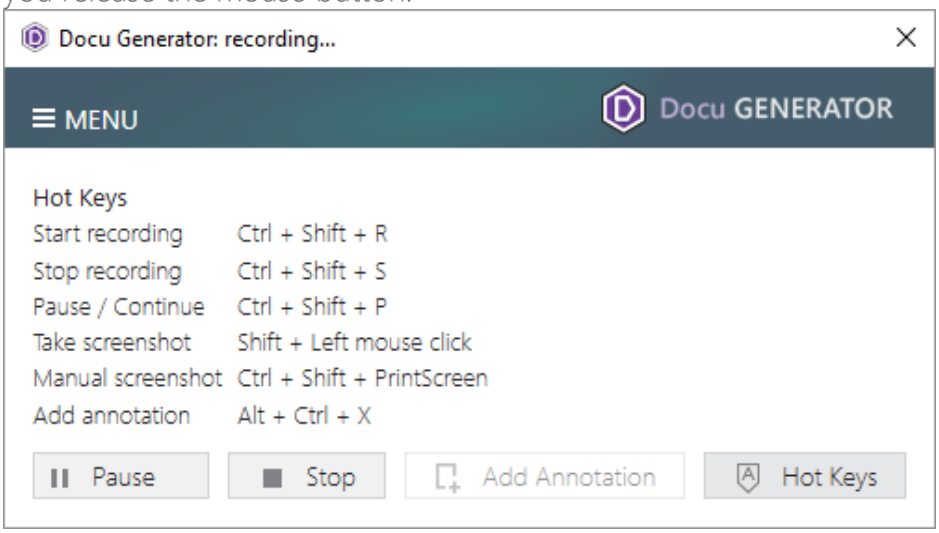

In order to launch Docu Generator with elevated permissions, press Shift and select Run as administrator from the context menu of the Docu Generator's shortcut.

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

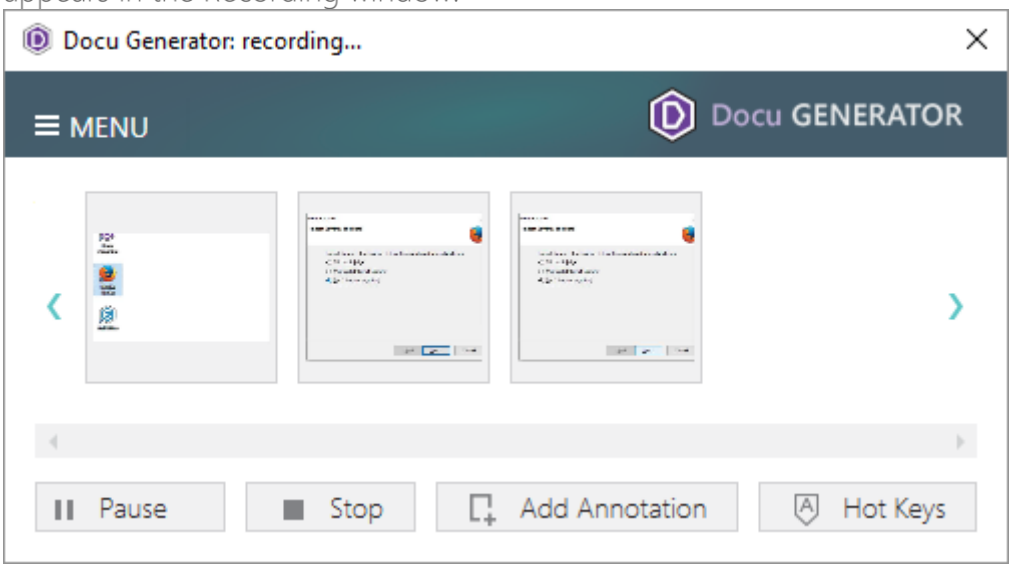



- Hover your mouse pointer on the Hot Keys button to check which of hot keys are used to take screenshots. Press the Shift key + Left mouse click to take a screenshot of an active window, dialog or screen area. Press Ctrl + Shift + PrintScreen, then click and drag the mouse pointer to select a rectangular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

you release the mouse button.

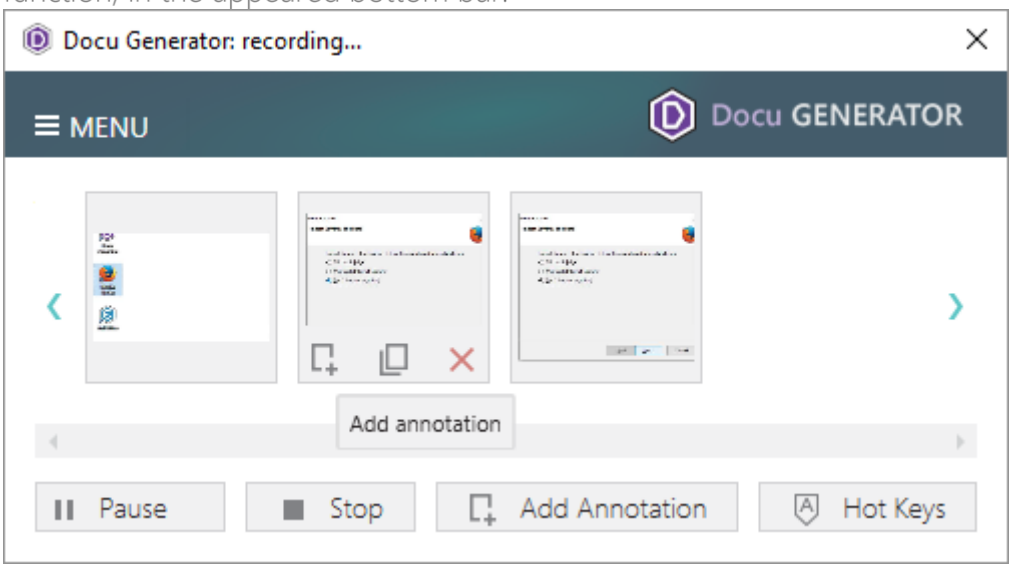



- Now, do the actions you want to record into the document. Launch executables, go through dialogs, select options and enter values just telling Docu Generator when to take another screenshot, as described above. A preview of each new screenshot immediately appears in the Recording window.

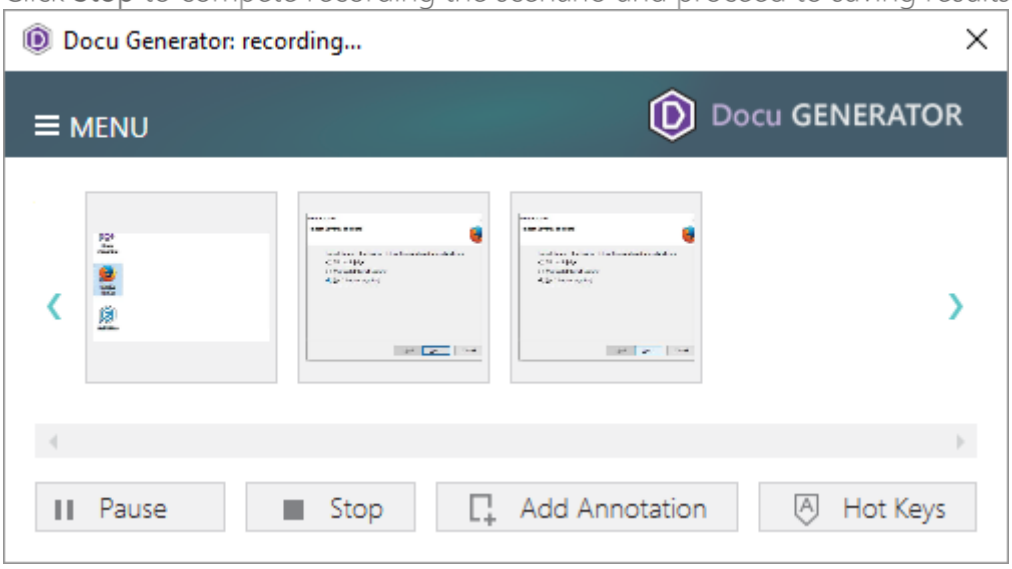



- In order to Add annotation, Copy to clipboard or Remove a screenshot, hover mouse pointer on a preview of the screenshot in the Recording window and select the respective

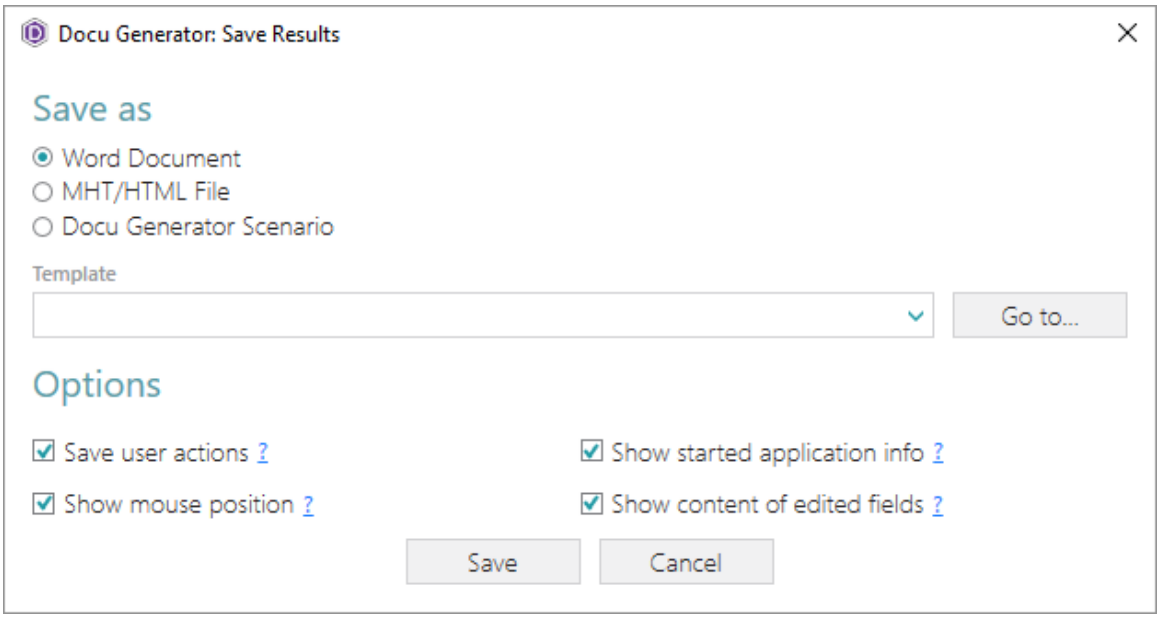
function, in the appeared bottom bar.




- 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.<br>In order to mark a position where screenshots and comments are to be placed to, add the following textual placeholder into a template:<br>[SCR]<br>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.<br>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.<br>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.

NOTE If Microsoft Word is unavailable on the machine, the option of saving DOC file is inactive, but you still can save a DSCN file and convert it on a machine with MS Office installed later.

# 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 |