# Table of Contents

**Chapter 1: Introduction** ................................................................................................................. 4

**Chapter 2: Getting Started** .............................................................................................................. 6
Getting to Know the Program Interface .......................................................................................... 6
Overview of MSI Creation Methods ............................................................................................... 6
How Installation Repackaging Works .............................................................................................. 6
Overview of Repackaging Best Practices ....................................................................................... 6
Demo: Repackage EXE to MSI Using Monitoring .......................................................................... 6
Demo: Create Customized Installations Using Monitoring ............................................................... 6
Demo: Repackage Silent Installations Using Wrapping .................................................................... 6
Installation Project Editing and Customizing .................................................................................. 6
Packages Testing and Troubleshooting ............................................................................................. 6

**Chapter 3: Program Interface Overview** .......................................................................................... 41
Welcome Screen .................................................................................................................................. 41
Projects View ..................................................................................................................................... 41
Project Details View .......................................................................................................................... 41
File System View ............................................................................................................................... 41
Registry View ...................................................................................................................................... 41
Environment Variables View .............................................................................................................. 41
Services View ..................................................................................................................................... 41
Assemblies View ................................................................................................................................. 41
Drivers View ....................................................................................................................................... 41
Printers View ...................................................................................................................................... 41
Custom Actions View ......................................................................................................................... 41
Wrapped Packages View .................................................................................................................. 41
Log View ............................................................................................................................................. 41
Operations View ................................................................................................................................. 41
Graphical User Interface features ....................................................................................................... 41

**Chapter 4: Installations Repackaging** ............................................................................................... 89
How to choose a Repackaging Mode? ............................................................................................... 89
Automated Monitoring ...................................................................................................................... 89
Advanced Monitoring ....................................................................................................................... 89
Repackaging via Wrapping ............................................................................................................... 89
Low-Level Repackaging .................................................................................................................... 89
How should I configure the repackaged installations to support an upgrade? ............................. 89

**Chapter 5: Installation Projects** ....................................................................................................... 105
Projects Management ....................................................................................................................... 105
Projects Preparation .......................................................................................................................... 105
File System Modifications ............................................................................................................... 105
Registry Modifications ....................................................................................................................... 105
Environment Variables Modifications ............................................................................................ 105
Services Modifications ..................................................................................................................... 105
Side-by-side Assemblies Deployment ............................................................................................. 105
Drivers Deployment .......................................................................................................................... 105
Printers Deployment ......................................................................................................................... 105
Using Custom Actions ...................................................................................................................... 105
Wrapping Existing Installations ....................................................................................................... 105
Capturing Changes ........................................................................................................................... 105
Creating MSI Package ...................................................................................................................... 105
MSI Package Language .................................................................................................................... 105
MSI Installation Context ................................................................................................................... 105
Signing Packages ............................................................................................................................... 105
Importing Package ............................................................................................................................ 105

**Chapter 6: Using Placeholders and Filters** ..................................................................................... 172
How should I correctly specify the filter condition? ......................................................................... 172
System Folder Definition Placeholders ............................................................................................. 172
Property Definition Placeholders ...................................................................................................... 172

**Chapter 7: Log** ................................................................................................................................. 177
Analyzing Log .................................................................................................................................... 177
Exporting Log ..................................................................................................................................... 177

**Chapter 8: Program Preferences** .................................................................................................... 180
MSI Package Builder Part .................................................................................................................. 180
Filters Part .......................................................................................................................................... 180
Miscellaneous Part ............................................................................................................................. 180
Chapter 9: Evaluation of the Program ........................................................................................................................................................... 198
  Evaluation Wizard .................................................................................................................................................................................. 198
  Where can I get my License Code? .................................................................................................................................................. 198
  How should I formulate the Extended License request? .................................................................................................................. 198

Chapter 10: Program Updates ...................................................................................................................................................................... 204
  Live Update ......................................................................................................................................................................................... 204
  Major Update ................................................................................................................................................................................... 204

Chapter 11: Main Program Actions .......................................................................................................................................................... 207
  Home Ribbon Page ........................................................................................................................................................................ 207
  Project Ribbon Page ....................................................................................................................................................................... 207
  View Ribbon Page .......................................................................................................................................................................... 207
  Program Ribbon Page ...................................................................................................................................................................... 207
  File System Contextual Ribbon Page ............................................................................................................................................... 207
  Registry Contextual Ribbon Page ........................................................................................................................................................ 207
  Environment Variables Contextual Ribbon Page .................................................................................................................................. 207
  Services Contextual Ribbon Page ................................................................................................................................................... 207
  Assemblies Contextual Ribbon Page ................................................................................................................................................ 207
  Drivers Contextual Ribbon Page ..................................................................................................................................................... 207
  Printers Contextual Ribbon Page .................................................................................................................................................. 207
  Custom Actions Contextual Ribbon Page ........................................................................................................................................ 207
  Wrapped Packages Contextual Ribbon Page ................................................................................................................................... 207

Chapter 12: Requirements ........................................................................................................................................................................ 225

Chapter 13: Edition Upgrade ........................................................................................................................................................................ 226

Chapter 14: How can I leave my Feedback? ........................................................................................................................................... 227

Chapter 15: About EMCO Software ............................................................................................................................................................. 229

Chapter 16: Contact Information ............................................................................................................................................................. 231
Chapter 1: Introduction

Welcome to EMCO MSI Package Builder. The program is designed to help you with repackaging of non-silent installations into silent MSI packages. This allows you to deploy repackaged installations remotely using any of software distribution tools. You can also use the program to create silent custom installations quickly and easily. This manual provides you with a detailed description of the program's features and shows you how to use these features in practice.

Downloading the Program

The program is available in thee editions with different sets of features. You can download evaluation versions of all editions from the Downloads page.

- The Professional edition allows you to repackage non-silent EXE installations into silent MSI packages using installation monitoring.

- The Enterprise edition allows you to use both installation monitoring and wrapping for repackaging. Installation monitoring works at an advanced level in this edition and allows you to repackage complex installations that deploy Windows services and drivers. You can also use wrapping if you need to repackaged installations that deploy Windows components.

- The Architect edition, in addition to installation repackaging and MSI generation features, allows you to create Microsoft App-V and MSIX/AppX packages. Using this edition, you can repackaged installations into different types of packages depending on your needs.
You can use an evaluation version during 30 days. After this period, you have to register the program to continue using it. MSI packages generated by an evaluation copy have some limitations: they can be deployed during 30 days following their generation date and are displayed with the evaluation mark in Windows Programs and Features after being deployed.

**Using Documentation**

The program is designed to provide you with an easy and straightforward approach to repackaging third-party installations and creating customized installations in the MSI format. To repack an installation, you just need to follow the steps of the repackaging wizard, and you will get an automatically generated MSI in the end. It means that you can use the program without referring to the documentation.

If you wish to get an overview of the program features, to understand how the program works and to see how to use the main features in practice, you can read the Getting Started guide. To get more information about the program features, you can refer to the corresponding chapters of this manual.

**Getting Help**

If you see that a generated MSI package doesn't work as required, first you need to check if the repackaging best practices were followed during the repackaging since the repackaging results depend on the compliance with the repackaging requirements and other aspects. You can learn more about this in the Overview of the Repackaging Best Practices chapter. You can also refer to MSI Packages Testing and Troubleshooting to learn about the general troubleshooting recommendations.

To get help on a problem, you can also contact support. Please send us the problem details including the name and the version of the installation you try to repack, the OS name and the platform (x86 or x64) of your repackaging environment, and the problem description. This information should help the support team to reproduce the problem and provide you with troubleshooting instructions.
Chapter 2: Getting Started

EMCO MSI Package Builder is a packaging tool that allows you to repackage non-silent installations into silent MSI packages and create custom silent installations quickly and easily. The program uses a unique repackaging technology that allows converting almost any non-silent installation into a silent MSI to enable its automatic deployment using Group Policy or other software distribution tools. The same technology can be used for automatic creation of customized MSI packages.

Using EMCO MSI Package Builder, you can benefit from the following features that are demonstrated in the course of this tutorial.

- **Multiple MSI packaging methods.** Depending on the project, you can use one of the provided packaging methods that is suitable for that project. You can convert non-silent installations into a silent MSI automatically using Live Monitoring. In the specific cases, when Live Monitoring cannot be used, you can wrap installations into an MSI package if those installations can be deployed silently.

- **Best-in-class repackaging technology.** The program allows you to repack a non-silent installation of any complexity into a silent MSI package. The program supports repackaging of installations that perform complex changes such as modifying file system and registry permissions, installing Windows services and drivers, changing Windows environment variables, etc.

- **Automatic MSI creation.** The program allows you to repackage EXE to MSI automatically: you just need to go through the deployment steps of the original EXE installation, and the program will capture all the performed changes on the fly and generate an MSI for you. You can also use the Live Monitoring technology to monitor any activity on a PC and generate an MSI reproducing the captured changes.
Easy MSI customization. Having created an MSI package, you can edit it, if required. You can have an access to all captured changes to modify them or add additional changes. You can also configure the MSI package to execute any scripts or executable files before and after the MSI installation.

In the following chapters, you will learn about the different MSI packaging methods offered by the program and will see how to use them in practice. In addition, you will get a brief overview of the installation repackaging internals and best practices. Finally, you will learn how to test generated MSI packages and how to troubleshoot MSI packaging problems.

What's Inside
- Getting to Know the Program Interface
- Overview of MSI Creation Methods
- How Installation Repackaging Works
- Overview of Repackaging Best Practices
- Demo: Repackage EXE to MSI Using Monitoring
- Demo: Create Customized Installations Using Monitoring
- Demo: Repackage Silent Installations Using Wrapping
- Installation Project Editing and Customizing
- Packages Testing and Troubleshooting

Getting to Know the Program Interface

When you start EMCO MSI Package Builder the first time, you can see the main screen of the program and the Repackage Installation wizard displayed at the top of it. The main goal of the program is installation repackaging, so the wizard is designed to guide you through the repackaging process. In the wizard, you can select the required repackaging method, choose the installations to be repackaged, configure the deployment package settings, etc. In the next chapter, you will learn how to select the right repackaging method in the wizard and you will see how to use the wizard in practice in the following chapters.

When the Repackage Installation wizard is closed and no installation projects are opened, you can see the starting main program screen [Pic 1]. The program has a Ribbon user interface, so the Ribbon menu is displayed at the top of the screen providing you with an access to the main actions of the program. Different types of the actions are located on different Ribbon tabs, so you can switch them, if required. Additional Ribbon tabs are displayed automatically when specific elements are selected and provide you access to the contextual actions.
Under the Ribbon, you can see opened projects on the left and the main area on the right. When you start the program, the **Projects** view is empty and the main area displays a list of the recent projects and the main actions of the program. You can click the main actions to initiate the corresponding repackaging wizards. You can open the **Repackage Installation** wizard to access all repackaging options by clicking **Repackage Installation** on the Ribbon.

![The starting main screen](image_url)
If you have completed repackaging or opened previously created installation projects, you can see the current projects in the **Projects** view. You can have more than one project opened to be able to copy data between the projects. A project consists of different types of resources. If you select a project node in the **Projects** view, you can see and edit the project settings. If you select, for example, the **File System** node, you can review and modify files and folders in an installation project using the file system editor [Pic 2]. There are different editors for managing different types of resources, so you need to select the required item in the **Projects** tree to review and modify its content using the corresponding editor.

![Pic 2. The program screen with an opened project](image)

At the bottom of the main screen, you can see the **Log** and **Operations** views. The **Log** view reports status information for deployment packages generation and other operations. You can use this view to find error messages and troubleshooting information in case of a problem. The **Operations** view displays the currently running operations and allows you to manage them.

You can learn more about the available views and editors in the **Program Interface Overview** chapter where you can find detailed explanations of features provided by every view.
Overview of MSI Creation Methods

EMCO MSI Package Builder is designed to create silent MSI packages, i.e. packages that can be deployed without interacting with a user. Such packages can be deployed manually on a local PC or remotely across a network using any remote software distribution tool.

The primary feature of EMCO MSI Package Builder is repackaging of non-silent installations into silent MSI packages. This allows you to convert non-silent installations, which cannot be deployed remotely, into MSI packages that are ready for remote deployment. You can use the program in many other cases, for example, to create custom silent installations or to combine multiple installations into a single MSI package. The program offers different approaches to creating an MSI to be used in different cases, so you need to know all available options and their features to select the right approach for every case.

What is Monitoring, Wrapping and Manual Creation?

The primary method of MSI creation is monitoring, which is available in both the Professional and Enterprise editions of the program. Using this method, you can repack non-silent installations into silent MSI packages and create custom silent installations according to your needs. It uses the Live Monitoring technology to capture all changes performed under Windows to create an MSI reproducing such changes. You can learn how it works in the How Installation Repackaging Works chapter.

The other MSI creation methods are wrapping and manual creation. Wrapping is available in the Enterprise edition of the program only. It can be used to repackage Windows system installations that cannot be re-packaged using monitoring. This method allows including one or multiple silent installations in the EXE or MSI format into an MSI package, so the included installations are deployed silently when you deploy the wrap MSI. Besides, you can create silent MSI packages manually through visual editors by specifying the changes to be included into an MSI.

How to use these MSI creation methods in the program? When you start the program for the first time, the Repackage Installation wizard is displayed automatically. In the Enterprise edition of the program, as the first step, you need to select if you would like to use monitoring or wrapping. In the Professional edition, you cannot use wrapping, so you need to select the monitoring mode that will be described later in this chapter. If you wish to create an MSI manually though editors, you can close the wizard and create a new project using the Ribbon options.
It is recommended that monitoring be used in most cases since it is a universal method, and wrapping or a manual MSI creation should be considered only if you have good reasons to do so. Below, you can find some examples demonstrating when using different types of the MSI creation methods makes sense.

**Monitoring usage examples:**

- Repackage one or multiple non-silent installations into a silent MSI.
- Create a silent MSI package that includes a repackaged non-silent installation and its customization (the installed product registration, configuration, etc.).
- Create a custom silent MSI that includes any captured changes.

**Wrapping usage examples:**

- Repackage an installation of a Windows component, such as .NET Framework, for example, that cannot be repackaged using monitoring. The repackaged installation should support silent deployment.
- Wrap one or multiple EXE/MSI installations that can be deployed silently into a silent MSI.

**Custom creation examples:**

- Create a simple MSI that deploys only a few resources.

Note that you can use monitoring not only for repackaging but also for creating custom installations by capturing the changes performed manually. It is much easier to use that approach than to create an MSI manually using visual editors. You can see how to use monitoring in practice in the Demos included in the Repackage EXE to MSI Using Monitoring and Create Customized Installations Using Monitoring chapters. Wrapping is demonstrated in the Demo contained in the Repackage Silent Installations Using Wrapping chapter.
How to Select the Monitoring Mode?

You can use monitoring in a number of cases, for instance, to repackage single or multiple non-silent installations, to repackage a non-silent installation and its customization, or to create any custom installation. To support all those cases, monitoring can work in two modes, which are **Automated Monitoring** and **Advanced Monitoring** that you can select in the **Repackage Installation** wizard.

When you select any of the above modes, EMCO MSI Package Builder uses the Live Monitoring technology to capture the file system and registry changes and create an MSI reproducing such changes. You will learn how it works in the next chapter. To understand which method you need to select, you can review their comparison below.

<table>
<thead>
<tr>
<th>How repackaging works</th>
<th>Automated Monitoring</th>
<th>Advanced Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>In the wizard, you specify a non-silent installation that you need to repackage, and EMCO MSI Package Builder automatically starts monitoring system changes and runs the specified installation. You follow the installation deployment steps, and after the deployment is finished, EMCO MSI Package Builder automatically stops monitoring and generates an MSI package that includes the captured changes performed by the monitored installation.</td>
<td>As the last step of the wizard, EMCO MSI Package Builder automatically starts monitoring system changes. You manually run one or several installations, apply the required software customization or perform any changes that should be captured and included into the MSI to be generated. Once you are done, you to switch back to EMCO MSI Package Builder and stop monitoring manually. EMCO MSI Package Builder automatically generates an MSI package that includes the captured changes reproducing the activity you performed during monitoring.</td>
<td></td>
</tr>
</tbody>
</table>
### Automated Monitoring

**Requirements**
- You provide an installation to be repackaged.
- You follow the deployment steps of the repackaged installation by selecting the install options manually.

**What it can be used for**
- To repackage one or multiple non-silent installations into a silent MSI package.

**Examples**
- Repackage a non-silent installation of *Open Office* into a silent MSI.
- Repackage non-silent installations of an antivirus and a firewall into a single silent MSI package.

### Advanced Monitoring

**Requirements**
- You run installations and/or perform any changes you need to capture manually.
- You manually stop monitoring in the program.

**What it can be used for**
- To repackage installations that require pre/post customizations, such as registration or tuning.
- To capture any custom changes performed manually or by any executable or script to create an MSI reproducing those changes.

**Examples**
- Repackage a non-silent installation and its registration after deployment.
- Create a silent MSI to install fonts.

In the next chapter, you can learn how monitoring works and what advantages and limitations it has. In addition, you can find recommendations that will help you to use monitoring successfully in the *Overview of the Repackaging Best Practices* chapter.
How Installation Repackaging Works

Installations monitoring is the primary repackaging method available in EMCO MSI Package Builder that allows you to convert almost any non-silent EXE installation into a silent MSI package. Usually, this kind of repackaging is required to deploy software on remote PCs, because only silent installations can be deployed remotely.

EMCO MSI Package Builder incorporates the best-in-class Live Monitoring technology used for repackaging. It integrates into Windows to capture file system, registry and permission changes performed by the repackaged installation. During repackaging, you just need to follow the installation steps of the repackaged installation manually. After you select the installation options, the monitored installer applies the corresponding file system, registry and permission changes that are captured by EMCO MSI Package Builder.
Once the installation is completed, EMCO MSI Package Builder automatically generates an MSI package that includes the captured changes, so the MSI installation is identical to the original installation and includes all the customizations you applied during the monitoring process. When you install the MSI, it applies the same changes that were applied when you deployed the original installation manually.

After the installation is repackaged, you can review the changes captured by EMCO MSI Package Builder. The program intellectually interprets raw-data file system and registry changes into Windows objects, so you can see, for example, Windows services and drivers that were deployed by the monitored installation and edit them through visual editors, if required, to generate a modified MSI.

**Advantages of the Live Monitoring Technology**

The Live Monitoring technology used in EMCO MSI Package Builder is the most capable installation repackaging technology available on the market today. It has the following features that allow you to perform repackaging quickly and easily.

- **Successful repackaging of simple and complex installations.** It does not matter what kind of changes are performed by the repackaged installation - EMCO MSI Package Builder can repackage anything, including complex Windows drivers, services, permissions, etc.

- **Changes are captured on the fly.** An MSI package is generated immediately after the monitored installation is finished. Using other repackaging solutions, you need to make Windows snapshots before and after the installation to compare them, so an MSI generation takes too long.

- **Monitoring is started and stopped automatically.** The program tracks the system information and knows when monitoring should be started and stopped without the risk of losing important data.

- **Repackaging is automated from start to finish.** You just follow a simple wizard and get a generated MSI at the end.

- **Unwanted changes are filtered out.** The program has a set of filters to avoid capturing changes performed by unrelated processes. Any captured change is associated with the process that generated it, so unwanted changes are removed automatically.

It is very easy to repackage an installation using the Live Monitoring technology. You can see how to use it in practice in the Repackage EXE to MSI Using Monitoring and Create Customized Installations Using Monitoring chapters.

**Repackaging Limitations**

EMCO MSI Package Builder allows you to repackage almost any installation, but still here are a few cases when repackaging using the Live Monitoring technology is not possible.

- **Windows cloning.** It is impossible to repackage Windows itself, because the repackaging technology is designed to be used for applications only. There are third-party tools that can clone Windows images, so you can use them if needed.

- **Installation of Windows components.** During repackaging, EMCO MSI Package Builder captures file system and registry changes, but when you install a system component such as .NET Framework, for example, it installs resources into protected areas of the file system and registry, so it is not possible to reproduce those changes. In this case, if the repackaged installation supports silent deployment, you can use the wrapping approach to repackage the installation.
Installation of system software that modifies Windows on reboot. When Windows is modified on reboot, it is not possible to capture and reproduce those changes, so use wrapping instead or repackaging for installations that apply such changes.

The limitations listed above are common for all repackaging tools, and there is no way to bypass them using the Professional edition of the program. The MSI Package Builder Enterprise edition allows you to use the wrapping method that should be used instead of repackaging to avoid the above limitations.

Overview of Repackaging Best Practices

The installation repackaging process is based on monitoring file system and registry changes performed by the monitored installation. Depending on the computer state before and during the repackaging process, you may obtain either correct or incorrect repackaging results. It is recommended that a special environment be used and the repackaging best practices explained below be followed to achieve correct repackaging results.

Recommended Repackaging Environment

When repackaging installations, you should always keep in mind that it is strongly advised:

- **Use a clean environment.** To avoid capturing unwanted changes produced by other software running on a PC, it is recommended that repackaging be performed on a clean environment where only the OS and EMCO MSI Package Builder are installed. For your convenience, you may create a virtual machine (VM) for repackaging, and having installed the OS and EMCO MSI Package Builder, create a snapshot with a clean VM configuration. Every time you need to perform repackaging, you should revert the VM to this snapshot in order to use a clean repackaging environment.

- **Repackage and deploy on the same platform.** When you deploy the same installation on different platforms (x86 and x64), it can create different resources, so you should repackage the installation on the same platform and, preferably, under the same OS that will be used for deployment of the package to be generated. If you need to deploy an installation on both the x86 and x64 platforms, you should create two separate packages for these platforms.

- **Disable anti-viruses, firewalls and Windows updates.** The system software working in the background may generate changes that will appear in the created package. To avoid that, it is recommended that Windows updates be disabled. Anti-viruses, firewalls and other security software may also produce unwanted changes and might cause an unstable or incorrect monitoring operation, so they should be disabled as well.
- **Prevent an auto-start of the repackaged application after its installation.** When you perform repackaging in the Automated Monitoring mode, EMCO MSI Package Builder automatically starts monitoring before running the repackaged installation and stops monitoring when the installation is finished. Some installations prompt you to run the installed application as the final step of the deployment, which prevents EMCO MSI Package Builder from detecting the end of the installation and stopping the monitoring process at the right place automatically. You have to skip running the installed application in the course of the installation process. If the application starts automatically after the installation, just close it to notify EMCO MSI Package Builder of the end of the installation process.

- **Skip reboots.** If the monitored installation requires a reboot, you should stop monitoring and generate a deployment package before rebooting.

Using a clean environment for repackaging is very important to get correct deployment packages, so it is strongly recommended that this rule be followed. If for some reason you cannot use a clean environment, there are some additional rules you should follow, but in this case, the monitoring results may not be as precise as those produced in a clean environment may.

- **Close all running applications and processes.** You should close all running applications and processes, except EMCO MSI Package Builder, before you start repackaging since they are not related to the repackaged installation but may cause changes that will appear in the repackaging results.

- **Uninstall the monitored application.** If an installation you need to repackage is already installed, uninstall it with all its components and dependencies.

Usually, repackaging problems are caused by non-compliance with one or more of the best practice recommendations, so if you get a deployment package that does not work properly, make sure you use a clean environment for repackaging and follow other best practices.
Demo: Repackage EXE to MSI Using Monitoring

Repackaging of existing non-silent EXE installations into silent MSI packages is one of the main features of EMCO MSI Package Builder, so let's see how to use it. Let's assume we need to distribute *Open Office* across our entire network. When you run the *Open Office* installation, it requires that you select the installation options and the packages to be deployed. You cannot use the original installation for remote deployment because it is not silent. To prepare the installation for remote deployment, we can repackage it into a silent MSI package using EMCO MSI Package Builder.

In this demo, we only need to repackage an *Open Office* installation without any pre- and post-install customization, so, as you learned in the Overview of the MSI Creation Methods chapter, we can perform repackaging in the Automated Monitoring mode by following the steps below.

**Step 1. Open the Repackage Installation wizard**

The Repackage Installation wizard is automatically opened when you start EMCO MSI Package Builder. If you do not see it on the screen, press the Repackage Installation button on the Home tab in the Ribbon. When the wizard is started, read the welcome information and press the Next button.

**Step 2. Select the Monitor Installation option**

The wizard allows you to repackage an installation using either monitoring or wrapping. To repackage a non-silent installation into an MSI package, select the Monitor Installation option and press the Next button. [Pic 1](#).

![Pic 1. Select the Monitor Installation mode](#)
Step 3. Select the Automated Monitoring option
As you learned in the Overview of the MSI Creation Methods chapter, you can convert a non-silent EXE into a silent MSI package using the **Automated Monitoring** option, so select this option in the displayed dialog. In this case, the program will automatically stop the monitoring and generate an MSI once the monitored installation is completed. Press the **Next** button to proceed to the next step.

![Automated Monitoring option](image)

**Pic 2. Select the Automated Monitoring option**
Step 4. Check the repackaging best practices
The repackaging results depend on the environment where such repacking is preformed, so the wizard shows you a brief summary of the repackaging best practices. You can find more details on this topic in the Overview of the Repackaging Best Practices chapter. Make sure you perform repackaging in a clean environment and follow other recommendations, then press the Next button.

Step 5. Select the installation file
At this step, we need to specify the path to the original installation file to be monitored, so select the path to the downloaded Open Office installation. Once the path to the installation file is specified, you are prompted to enter the installation parameters. The parameters are optional and should be specified only if you need to run the monitored installation in a special way. In our case, we can leave the parameters empty and press the OK button to confirm the installation configuration. As you can see, the wizard allows you to specify multiple installations to be monitored, so the installations will be executed one-by-one and will be repackaged into a single MSI package. We need to create an MSI for the Open Office installation only, so we can press the Next button.

Pic 3. Specify the installation to be repackaged
Step 6. Specify the MSI properties

The wizard prompts you to specify the properties of the MSI package to be generated as a result of the repackaging, so you should enter the MSI path. In this dialog, you should also specify the required Manufacturer name, so you can enter your company name, for example. All other information, such as Product Name, Version and Icon is extracted automatically from the original installation, and you can edit it, if required. You can also manage other options in this dialog, but for most of the installations, including this one, the default options are optimal, so you do not need to change them. You can learn more about the options in this dialog in the Creating MSI Package chapter. Once you have specified the required settings in the dialog, you can press the Next button.

Pic 4. Specify the MSI properties
Step 7. Filter out the running applications/processes

When the monitoring is started, the program will be capturing all file system and registry changes. Since multiple applications/processes may be running at the same time during monitoring, the changes applied by all running Windows processes will be captured. If you have other applications/processes running during repackaging, they may create unwanted changes that will appear in the capturing results and in the generated MSI package, so it’s recommended that they be filtered out. If you use a clean environment for repackaging, the wizard will be displaying an empty list of running applications. In case the list is non-empty but the running applications do not relate to the installation to be monitored, select them to filter the changes made by them out of the monitoring results. Once you have finished configuring the filters, press the **Finish** button to start the monitoring.

Pic 5. Filter out the running applications
Step 8. Follow the installation steps
As you can see, the wizard has been closed and the specified original installation has been started. It means that monitoring is in progress now, and you need to follow the steps of the original installation. You can select the required installation options and, as a result, the original installation will create corresponding files and registry keys that are monitored by the program. Once the installation is completed, EMCO MSI Package Builder will automatically stop the monitoring and generate an MSI package. Note that if the installation prompts you to run the installed application, you should skip this step to let EMCO MSI Package Builder detect that the installation has been completed. If the application starts automatically after the installation, just close it to notify EMCO MSI Package Builder about the end of the installation process.

Step 9. Getting the generated MSI package
Once the installation is completed, you can return to EMCO MSI Package Builder that should automatically start generating an MSI package using the capturing results. Depending on the captured data size, the generation process may take some time. Once the generation is completed, you can see the generated MSI package in Windows Explorer.

Pic 6. Getting the generated MSI
If you use an evaluation version of EMCO MSI Package Builder to generate an MSI, you get a trial MSI package with a few limitations. The generated package can be deployed during 30 days only and after deployment is displayed in Windows Programs and Features with the evaluation mark. These limitations do not influence the functionality of the repackaged application, so you can test it to make sure it works as expected. You can learn how to test MSI packages in the MSI Packages Testing and Troubleshooting chapter.

When generation of an MSI package is finished, EMCO MSI Package Builder automatically opens the MSI project, so you can review its content and customize it, if required. You can see the file system, registry and other resources captured during repackaging, modify them and add new resources to generate a new custom MSI package using the data stored in the MSI project. Besides, you can add custom pre- and post-install actions to be executed before and after the MSI deployment. You can learn how to use those features in the Installation Project Editing and Customizing chapter.

**Demo: Create Customized Installations Using Monitoring**

In some cases, you may need not only to repackage a non-silent installation but also to customize it after installation to apply a license code or change the installed application settings and to get a deployment package that includes the installation and its customization. As you may have learned in the Overview of the MSI Creation Methods chapter, you can repackage such installations using the Advanced Monitoring mode of repackaging. This mode allows you not only to monitor an installation but also to monitor any changes performed in Windows and create a deployment package reproducing those changes. This very powerful feature of EMCO MSI Package Builder allows you to create any type of installation quickly and easily. Let's have a look at some examples of how you can use it.

**Create a Customized Firefox Installation**

Suppose we need to distribute Firefox across a local network. The standard Firefox installation doesn't work silently, so we need to repackage it into an MSI. An additional requirement here is that the Firefox installation should show our company home page at start and should have certain changed proxy settings. Those changes can be applied in Firefox options after installation, so we need to monitor the installation and application customization steps to create the required MSI.
**Step 1. Open the Repackage Installation wizard**
As the first step, you need to open the Repackage Installation wizard. If it isn’t opened automatically, press the Repackage Installation button in the Home Ribbon page, read the welcome information and press the Next button.

**Step 2. Select the Monitor Installation option**
We are going to use installation monitoring, so select the corresponding option and press the Next button to proceed [Pic 1].
Step 3. Select the Advanced Monitoring option
We need to monitor not only the Firefox installation but also the post-install customization, so we cannot use the automatic monitoring start and stop functions provided by EMCO MSI Package Builder by default. As you learned in the Overview of MSI Creation Methods chapter, we need to use Advanced Monitoring to start and stop monitoring manually, so select the corresponding option and press the Next button [Pic 2].

Pic 2. Select the Advanced Monitoring mode
Step 4. Check the repackaging best practices
When you use monitoring, you need to follow the repackaging best practices to get accurate repackaging results. Review the recommendations displayed by the wizard, make sure that you perform repackaging in a clean environment and follow other best practices, and then press the **Next** button. You can learn more about this topic in the *Overview of Repackaging Best Practices* chapter.

Step 5. Specify the project name
EMCO MSI Package Builder will automatically create an installation project when the monitoring is completed, so you need to enter the project name and press the **Next** button.

Step 6. Specify the MSI properties
At this step, you need to specify the properties of the MSI package to be generated. You need to specify the path to the MSI file that will be generated automatically after the monitoring ends. You also need to specify **Product Name** and **Version** that should be the same as in the repackaged *Firefox* installation. You can enter your company name as **Manufacturer** and click on **Icon** to pick it from the *Firefox* installation file [Pic 3]. To learn more about the options in this dialog, refer to the *Creating MSI Package* chapter. Press the **Next** button when you finish editing the MSI properties.

[Pic 3. Specify the MSI properties]
Step 7. Filter out the running applications/processes
To prevent any unwanted changes from appearing in the repackaging results, you need to close all applications that are not related to the monitored process. EMCO MSI Package Builder automatically detects all running applications before starting the monitoring, so you can select the application the changes initiated by which should be excluded from the monitoring results. If you follow the repackaging best practices and use a clean environment, you will see an empty list in the wizard and can start monitoring. Otherwise, if you see a list of running applications, choose the applications that should be excluded from the monitoring and press the **Finish** button to start the monitoring.

Pic 4. Filter out the running applications
Step 8. Follow the installation and customization steps
At this step, the monitoring is started, and any file system and registry changes are being captured. Note that you haven't specified the Firefox installation in the wizard, so it won't be started automatically, and you need to find the installation file in Windows Explorer and run it manually. You need to follow the Firefox installation process until Firefox is installed. The difference from repackaging in the default mode is that monitoring isn't stopped automatically when the installation is finished, so you can apply the required customization. After Firefox is deployed, you can run the Firefox application and specify your company website as the home page, then set the required proxy settings in the program options.

Step 9. Stop the installation and get an MSI package
Now that all the Firefox installation and customization steps are finished, you can go back to EMCO MSI Package Builder and press the button to stop the monitoring. As you can see, once the monitoring is stopped, EMCO MSI Package Builder generates an MSI package automatically, and you can see it in the Windows Explorer.

![Pic 5. Stop the monitoring](image)

After getting an MSI package, you need to test it before you can deploy it across a network. The testing approach is explained in the Packages Testing and Troubleshooting chapter. If you generate an MSI using the trial version of EMCO MSI Package Builder, you get a trial MSI package that can be deployed during 30 days only and that displays a trial message in Windows Programs and Features after the deployment. These limitations do not influence the functionality of the repackaged application, so you can fully test it to make sure it works as expected.

As you can see, the installation repackaging in the Advanced Monitoring mode works automatically and is identical to the Automated Monitoring except for the requirement to run the repackaged installation and stop the monitoring manually, but that allows you to repackage installations that require customization.
Create a Custom Font Installer

Using repackaging in the Advanced Monitoring mode, you can easily get custom installations for any software deployment or Windows administration task. For example, let's assume that you need to distribute a font across your network, but you only have a font file. Of course, you can create an installation from scratch manually using the visual editors of EMCO MSI Package Builder, but in this case, you need to know a lot of technical details like the folder where the font should be installed, how to register the font, etc. Using Advanced Monitoring, you can create the required MSI automatically and let EMCO MSI Package Builder capture all the required changes while you install the font manually. Let's see how it works.

Steps 1 - 7. Select the Advanced Monitoring mode and configure the MSI properties
Every time you use Advanced Monitoring, the initial steps are the same: you need to open the Repackage Installation wizard, select the Advanced Monitoring mode and make sure you follow the repackaging best practices. You also need to specify the MSI properties and to use the installation-specific properties such as the MSI path, Product Name and Version. You can see the details steps 1 to 7 in the example above. In the dialog where you need to provide the MSI properties, you should use the Always restart computer after the installation option for Restart Mode. It is required to make the installed font visible in the system after the deployment.

Step 8. Install the font
Now that EMCO MSI Package Builder monitors all the changes, you can select the font file to be installed in Windows Explorer and choose the option to install it from the context menu. Check the Windows Control Panel > Fonts to make sure the font has been installed correctly.

Step 9. Stop installation and get an MSI package
Once the font installation is finished, you can go back to EMCO MSI Package Builder and press the button to stop the installation and generate an MSI. As a result, the MSI generation is started automatically, and you get an MSI package that installs the font.

Using this approach, you can create an MSI automatically for any system administration task. For example, if you need to create an installation that deploys a set of files and registry keys, at Step 8, you can copy these files to the required target folders in Windows Explorer and create the required registry keys in Registry Editor manually, and EMCO MSI Package Builder will create an MSI that reproduces the captured changes.

Demo: Repackage Silent Installations Using Wrapping

As you learned in the Overview of the MSI Creation Methods chapter, you can use the wrapping method to repackage specific installations that install Windows components and cannot be repackaged using monitoring because it isn't possible to reproduce these installations by capturing their changes.

When you wrap an installation, the original installation file is included into the wrapped MSI package and is executed during the MSI deployment. Since the generated MSI package should be deployed silently, the included installation should be configured to run silently as well, so you can use this method if you know how to deploy an installation silently.

Let's see how to use wrapping. For example, we need to repackage a .NET Framework installation that is a Windows component, so we cannot repackage it using monitoring.
Step 1. Determine and test the silent installation parameters
Since we need to create a wrapping MSI package ready for silent deployment, we should ensure that the included installation could be deployed silently. Using the Internet search, we can find out that .NET Framework installation can be deployed silently if the installation file is executed with specific parameters. Before generating an MSI package, we need to ensure that the silent deployment of the original installation works as expected. Use a virtual machine with a clean environment to test the silent installation of .NET Framework, which should be completed automatically while you should not be prompted to select any installation settings. If the tested installation can be deployed silently, you can proceed to the next step.

Step 2. Open the Repackage Installation wizard
Run EMCO MSI Package Builder. By default, you should see the opened Repackage Installation wizard at the start of the program. If it doesn't open automatically, you may click the Repackage Installation button on the Home Ribbon page. Once the wizard is launched, read the welcome information and press the Next button.

Step 3. Select the Wrap Installation option
We are going to use installations wrapping, so select the Wrap Installation option in the wizard and press the Next button to proceed to the next step Pic 1.

Pic 1. Select the Wrap Installation option
Step 4. Specify the installation file and the installation options

You are prompted to enter the path to the installation. In the file picker dialog, specify the path to the .NET Framework executable installation file. You should specify the installation options in the New Wrapped Package dialog. You can change the package name to .NET Framework, if you wish. In this dialog, you also need to specify the silent command-line parameters to be passed to the installation. As we know, a .NET Framework installation can be deployed silently when executed with the /q /norestart parameters, so enter them into the Parameters field and press OK to close the dialog. Finally, press the Next button in the wizard to proceed to the next step.

Pic 2. Configuring the .NET Framework install options
Step 5. Specify the MSI properties
You should enter the path to the MSI package to be generated. You also need to enter the Manufacturer name and change other options, if required. Press the Finish button to start the MSI generation.

![Pic 3. Specifying the MSI generation options]

Step 6. Get the MSI package
EMCO MSI Package Builder generates an MSI package that wraps the specified silent installations and deploys them when you deploy the MSI. You can see the generated package in Windows Explorer that opens after an MSI is generated.

You need to test the generated MSI package before you can deploy it across a network to make sure it works as expected. Follow the testing recommendations provided in the MSI Packages Testing and Troubleshooting chapter.

If you use the trial version of EMCO MSI Package Builder, the generated MSI has the trial limitations, i.e. it can be deployed during 30 days only, and its entry displayed in Windows Programs and Features after the installation bears the trial mark.

Using wrapping, you can also create an MSI package that deploys multiple installations. The wrapped package may include both EXE and MSI installations, the only requirement for wrapping being the ability to support silent deployment.
Installation Project Editing and Customizing

Regardless of the repackaging method you use to create a deployment package, you can see that the main screen of EMCO MSI Package Builder changes when the package generation is finished. EMCO MSI Package Builder automatically creates an installation project when you use installation monitoring and wrapping and opens such a project after the deployment package is generated, so you can review and edit it. The projects are created in the program's storage, and you can open and edit them anytime. You can apply custom changes through the visual editors and generate a modified deployment package.

In most cases, you don't need to review and edit installation projects because you get deployment packages at the end of the Repackage Installation wizard, but you can get access to the advanced features if you know how to edit and customize installation projects. Let's see how it works in practice.
Review and Modify Project Properties

When a deployment package generation is finished, you can see that the Projects view on the left of the main screen contains a number of new items, which are the recently created installation project and a set of the installation changes located under it. If you have any previously created deployment packages, you may open them to add to the Projects view. If you select the root node of a project in the Projects view, you can see the project properties on the right of the main screen. Those are the properties you specified in the Repackage Installation wizard.

If you made a mistake when creating a deployment package and specified, for example, a wrong Product Name or Version, you can edit them in the project settings and press the Create MSI Package button on the Ribbon to generate a new package with the modified properties.
Configure an MSI to Update Another MSI

You can configure an MSI package to be installed as an update for an older version of the same product installed as another MSI package. Windows Installer supports updates automatically; so if it detects that you install an update, it uninstalls the old MSI package and then installs the new one.

To configure an MSI to be deployed as an update, you only need to modify the MSI package properties. First, you need to modify the properties displayed in the Identification and Upgrade section. Enable the Allow an installation upgrade option and specify the same Upgrade GUID as the one used in the MSI you wish to update. You also need to update the Version number to ensure that the new MSI has a higher version number than the old one. Learn more about the technical requirements for an upgrade in the How should I configure the repackaged installations to support an upgrade? chapter.

Digitally Sign MSI Packages

MSI packages apply Windows changes on deployment, so Windows Installer requires that they should be run under an account with administrative permissions. The Windows UAC automatically checks the validity of digital signatures when software is executed with administrative permissions. By default, MSI packages generated by EMCO MSI Package Builder aren't digitally signed, so you can see the signature warning in the Windows UAC when you deploy such MSI packages manually. If you need to deploy the generated MSI packages remotely, note that such deployment tools as Microsoft SCCM don't deploy unsigned MSI packages.

EMCO MSI Package Builder allows signing MSI packages if you have a digital code-signing certificate and specify it in the program Preferences dialog. You can enable and disable digital signing of the generated MSI packages by configuring the corresponding options in the Digital Signature section of the MSI package properties. You can learn more about digital MSI signing in the Signing Packages chapter.
Review and Edit Changes

Under the project node in the **Projects** view, you can find the list of changes applied by the installation. These changes are grouped by types and are displayed as separate nodes in a tree, so you can see **File System, Registry, Assemblies**, etc. If you select a node, on the right you can see a specific editor for the corresponding type of resources and can use it to review and modify the changes [Pic 2]. You can learn how to use every available editor in the **Installation Projects** chapter, but here you can get a quick overview of the available features.

If you select the **File System** node, for example, you can see the structure of folders on the left and the modified files on the right. The resource icons show the applied operation type, so you can see the files that were created, modified or deleted. If you created an installation project automatically using repackaging, you can notice that the file system paths are not absolute and include so-called System Folders. This is required to make installations portable from one PC to another, because different PCs may have different paths to standard Windows folders that store applications, their data and so on. EMCO MSI Package Builder can convert an absolute path to system folders automatically, so it's recommended that this conversion be performed if you edit a project manually. You can learn more about System Folders in the **System Folder Definition Placeholders** chapter.
If you wish, you can switch the editor to the flat representation mode by clicking the Flat button on the contextual Ribbon page. In this case, the changes are displayed as a grid and this grid is grouped by default by the Creator values. This mode is helpful if the project was created using monitoring, because it allows seeing the captured processes and the files they changed. If you need to troubleshoot a problematic package, you can use this mode to find and delete the changes performed by unrelated process monitored during the repackaging. The advanced package editing mode can only be used in special cases.

![Pic 3. File System changes grouped by processes that generated the changes](image)

You can modify the set of file system and other resources using the actions available in the editor's context menu and in the contextual Ribbon pages. If you need to review and modify other types of changes, you can select the required node in the Projects view and use the corresponding editors.
Define Custom Actions

EMCO MSI Package Builder allows you to configure a deployment package to execute custom actions before and/or after the installation or uninstallation and to perform system configuration actions. Such actions can be managed through a special editor displayed after selecting Custom Actions in an installation project. You can configure a package to run any executable commands, scripts or applications as pre and post actions. The pre-install and post-uninstall executables should be available on the deployment PC so they cannot use the project files; but for post-install and pre-uninstall, you can run the executables deployed by the generated package. You can learn more about configuring custom actions in the Using Custom Actions chapter.

Using EMCO MSI Package Builder, you can customize installations according to your needs. For example, you can automatically create an installation project by monitoring the required actions and then review and modify the project contents by adding additional resources or a configuration of custom actions.

Packages Testing and Troubleshooting

Once an MSI package is generated, it needs to be thoroughly tested before being deployed remotely across a network. To test an MSI, it’s recommended that it be installed manually on a virtual machine with a fresh OS installation. Following the deployment, you need to run the installed application to make sure it works as expected. Sometimes, the changes applied by an MSI aren’t visible for Windows after the MSI deployment. This may cause improper operation of the installed application. It’s recommended that the PC be rebooted to let Windows apply the changes if you see that the installed application works incorrectly.

If you notice a problem with the generated MSI, you can identify the reason for the problem and repeat the repackaging process. Many problems can be resolved simply by using a clean environment for repackaging and following the best practices explained in the Overview of the Repackaging Best Practices chapter. Below, you can find a list of the most common MSI problems and recommendations on how to resolve them.
A file is locked during MSI install
This problem usually happens when you don't use a clean environment for repackaging, and an unrelated process, such as an antivirus running in the background during the repackaging, generated a file change that is captured and included into the MSI. When you deploy the MSI, it tries to modify the file, but the file is in use by the running antivirus, therefore the MSI deployment cannot be finished. To resolve this problem, you need to repeat the repackaging using a clean environment as suggested in the repackaging best practices.

A file is locked during MSI uninstall
This problem is similar to the one described above. An MSI includes a change initiated by an unrelated application, which may be an antivirus, for example. When you deployed the MSI, the file change is accepted, but when you try to uninstall the MSI, the file is locked by the antivirus. To resolve this problem and uninstall the MSI, you need to close the program that locks the file and then uninstall the MSI. In any case, the generated MSI is incorrect and you need to create a new MSI using a clean environment for repackaging.

An MSI was installed successfully, but the installation does not appear in Windows Programs and Features
By default, EMCO MSI Package Builder creates MSI packages that are deployed per user, so if you didn't use any special options during the MSI deployment, the installation is visible for the current user only, and if you switch to another user, you won't be able find it in the Windows Start menu or in Programs and Features. To resolve this problem, you may provide special deployment options for the msiexec utility to be deployed to all user accounts during the MSI deployment. Alternatively, you may change the MSI project settings to use the All Users option instead of the Current User option and generate a new MSI package.

An MSI upgrade doesn't remove the old version of the same application
Windows Installer uninstalls the previous version of MSI automatically if a new version is being installed. Note that this only works for MSI packages, so you cannot rely on this feature if the previous version of the application was installed by an EXE installer. Besides, to use this feature, you need to properly configure the MSI version, Upgrade GUID and the installation scope as explained in the How should I configure the repackaged installations to support an upgrade? chapter.

An MSI doesn't replace some files
When you deploy an MSI package, Windows Installer checks the file properties for every file to be replaced. For files that contain version information in their properties, like DLL or EXE files, Windows Installer compares the versions of the new file and the existing one and replaces the existing file only if the new file version is greater than that of the existing one. This behavior is configured in the Install Options section of the MSI properties, so you can change it to replace all files unconditionally.

If you face a problem, first make sure that you follow all the repackaging best practices and repeat the repackaging in a clean environment. If you still have questions after that, you can contact support and provide information on the problem and the installation you try to repackage.
Chapter 3: Program Interface Overview

EMCO MSI Package Builder is a system program that needs to track file system and registry changes in the monitoring mode, so you need to have administrative permissions in Windows to run it. When you run the program on Windows Vista or a later OS, you need to allow the program to apply system changes in the User Account Control (UAC) prompt. If you use the evaluation version of the program, you need to press the **Evaluate** link to close the **Evaluation Wizard**, which appears at startup.

The program has a modern Ribbon-based interface. **Pic 1**: the **Ribbon** is available at the top of the main screen. It consists of four main tabs with items and can also display additional contextual tabs highlighted on the Ribbon when you use some of the available views and editors.

**Pic 1. The main program window**
Under the Ribbon, on the left side of the screen, you can find the Projects view. This view displays one or multiple installation projects that are currently opened in the program. For a project you can define the project settings, a set of file system and registry changes, changes to the environment variables configuration, the windows services deployment actions, the side-by-side assemblies configuration, drivers deployment actions, custom actions and wrapping settings.

On the right side of the Projects view, you can see the main area, which shows different views depending on your current selection. If there are no projects opened, you can see the Welcome Screen there with a set of available actions. If a project is selected in the Projects view, you can define the project build properties in the main area. To define a set of file system, registry or other changes to be performed by a deployment package, you should select the corresponding node under the project node in the Projects view. The content of the main area will be switched to the appropriate view. The available views for a regular installation project are the following: File System, Registry, Environment Variables, Services, Assemblies, Drivers, Printers and Custom Actions. As for a wrapped packages installation project, there is only the Wrapped Packages view.

At the bottom of the main screen, you can find the Log and Operations views. The Log view can be used to see all the events and error messages reported by the program. The Operations view can be used to manage the running operations.

To configure the program settings, you can open the Preferences dialog by selecting the corresponding option under the Application menu. The Preferences dialog allows you to configure the monitoring filters and other settings in the program.

**What's Inside**

Welcome Screen
Projects View
Project Details View
File System View
Registry View
Environment Variables View
Services View
Assemblies View
Drivers View
Printers View
Custom Actions View
Wrapped Packages View
Log View
Operations View
Graphical User Interface features
Welcome Screen

The **Welcome Screen** is displayed just when you start the program. It is located right in the middle of the program main window and is intended to help you start working with MSI Package Builder.

![Pic 1. The Welcome Screen](image)

The **Recent Projects** part of the **Welcome Screen** contains links to open recently used projects, open another project or create a new project. The **Get Help** part contains links to the information that is useful to anyone who is new to MSI Package Builder and would like to know more about the program. The main part of the **Welcome Screen** is the **Product Actions** one. It contains shortcuts to the actions that are used to perform installations repackaging.
Projects View

The Projects view is located by default on the left of the main program window. It is designed to display a set of currently opened MSI Package Builder projects and the nodes representing the changes to be performed by the deployment packagePic 1. On the project node, you can also see the current version of the deployment package to generate, and on each changes node, you can see how many entries are available in the corresponding set of changes.

![Projects View](image)

Pic 1. The Projects view

The icon next to every item represented in the Projects view is used to describe of the item's type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not, and if there are any processing problems. Below is the list of icons used:

- an MSI Package Builder project;
- an MSI Package Builder project is being operated;
- an MSI Package Builder project containing errors that should be resolved before creating a deployment package;
- an MSI Package Builder project based on monitoring results that is not prepared;
- an MSI Package Builder project based on monitoring results that is prepared;
- an MSI Package Builder project based on monitoring results that could not be fully prepared (not all of required files were available during the prepare operation).
The **Projects** view is the starting point for the changes management and creating a deployment package based on existing projects. It is also a master view for the main program area – the views displayed there are depending on the selection in the **Projects** one.

### Functions Overview

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Projects Management</strong></td>
<td>From the <strong>Projects</strong> view, you can create, rename, delete, open and close MSI Package Builder projects. To perform these actions, you can either use the pop-up menu of the <strong>Projects</strong> view or the buttons from the <strong>Project Management</strong> group on the <strong>Home</strong> Ribbon page. The actions for creating a new project, opening existing projects and closing all projects are available in the pop-up menu displayed on an empty selection, and to edit, rename, delete and close specific projects, you should firstly select those projects. The detailed information about the projects management abilities is available in the <strong>Projects Management</strong> section of this document.</td>
</tr>
<tr>
<td><strong>Projects Preparation</strong></td>
<td>Within the <strong>Projects</strong> view, it is possible to prepare the projects created on a basis of the monitoring results. The preparation steps consist of copying the file system resources to the projects storage, repairing missing links and rolling system folders. These actions are available in the pop-up menu and on the <strong>Project</strong> Ribbon page. To learn more about preparing the projects, refer to the <strong>Projects Preparation</strong> section of this document.</td>
</tr>
<tr>
<td><strong>Capturing Changes</strong></td>
<td>From the <strong>Projects</strong> view, it is possible to capture changes to the underlying operating system and create a project, containing the performed changes. To start a new session of changes capturing use either the <strong>Start Monitoring</strong> item from the pop-up menu displayed on an empty selection, or the <strong>Start Monitoring</strong> button from the <strong>Capture</strong> group on the <strong>Home</strong> Ribbon page. To stop the changes capturing session, use the <strong>Stop Monitoring</strong> items. The capturing abilities can be used for low-level repackaging and for simple monitoring of the application activity.</td>
</tr>
<tr>
<td><strong>Deployment Package Generation</strong></td>
<td>You can easily create a deployment package containing the changes represented by any project displayed in the <strong>Projects</strong> view. Just select that project and choose the <strong>Create MSI Package</strong> item in the pop-up menu – it is equivalent to using the <strong>Create MSI Package</strong> button from the <strong>Builder</strong> group on the <strong>Home</strong> Ribbon page. The detailed information about the deployment package generation abilities is available in the <strong>Creating MSI Package</strong> section of this document.</td>
</tr>
<tr>
<td><strong>Deployment Package Import</strong></td>
<td>It is possible to create a project by extracting any deployment package generated with MSI Package Builder in the <strong>Projects</strong> view. This feature is called a deployment package import. To perform import, use either the <strong>Import Package</strong> item from the pop-up menu displayed on an empty selection, or the <strong>Import Package</strong> button from the <strong>Project Management</strong> group on the <strong>Home</strong> Ribbon page.</td>
</tr>
<tr>
<td><strong>Search</strong></td>
<td>Within the Projects view, you can execute a search for specific project using the Find item from the pop-up menu.</td>
</tr>
</tbody>
</table>

From the **Projects** view pop-up menu, depending on the selection, it is possible to perform projects management, capture changes, import existing deployment packages, etc.
Project Details View

The **Project Details** view is displayed within the main program area when an MSI Package Builder project is selected in the **Projects** view. This view is used to define the data that is used to configure a deployment package being created on a basis of the project Pic 1. The settings are grouped by tabs.

![Pic 1. The Project Details view](image)
On the **General** tab in the **Product Details** group, you should define the generic package properties such as the product name, manufacturer, language, etc. The **Product Support Information** group can be used to override the common support configuration defined within the program preferences.

The **MSI** tab is used to configure the Windows Installer package generated base on the project. The **General Properties** group allows you to define the MSI installation context, the way the generated package behaves regarding the Programs and Features section of the Control Panel and if the PC should be restarted for the installation to complete. The **Product Information** group is used to define generic MSI package properties, such as the product name, manufacturer, language, etc. The **Identification & Upgrade** group is used to define the product GUID and upgrade GUID, if it is required. Within the **Digital Signature** group, it is possible to override the common package signing configuration for the selected project, if it is required. The **Install Options** part allows you to define the supported operating systems, the required Microsoft .NET Framework version and install mode. And the **Summary Info** part is responsible for defining summary information for the package.

This configuration is stored between sessions and is used to complete the corresponding properties during an MSI package creation. For a detailed description of each option, refer to the **Creating MSI Package** section of this document.
File System View

The File System view is displayed within the main program area when the File System node of any project is selected in the Projects view. This view is used to define and review the changes to be performed to the file system when installing a deployment package created on a basis of this project.

Pic 1. The File System view

The file system modifications can include modifications of folders, files and shortcuts. The location of each element is represented with a file system path, that consists of drive name or system folder and several folders included into each other. When monitoring existing installations, the required changes are created automatically and added to the corresponding project representing the installation process. It is also possible to create the changes manually or import files and folder from local file system.

Tree

The Tree button from the Presentation group of the contextual File System tab should be used to display the file system modifications in form of a tree, similar to Windows Explorer.
The Flat button from the Presentation group of the contextual File System tab allows you to display the file system modifications as a raw list, where each element is represented with its full path.

Information on the file system modifications can be represented both in form of a folder tree with data on each folder and in form of a resource list. When displayed as a resource list, it is by default grouped by processes responsible for the creation of each entry. For entries created manually, such a procedure is not applicable. You can clear the grouping using the table features, and it is possible to revert to the default layout at any time using the Reset Layout item from the configuration menu.

The icon next to every item represented in the File System view is used to describe the item's type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not, and if there are any processing problems.

Below is the list of icons used to represent the item type and state:

- a logical drive;
- a logical drive is being processed;
- a logical drive that contains a permanent path;
- a system folder;
- a system folder is being processed;
- a system folder that contains a permanent path;
- a folder;
- a folder is being processed;
- a folder is a part of a permanent path;
- a file;
- a file is being processed;
- a file is permanent;
- a shortcut;
- a shortcut is being processed.

The following overlays are used to represent the operation to be performed with an item:

- an item should be created;
- an item should be modified;
- an item should be deleted;
- permissions should be defined for an item.

As for the problematic situations, the following overlays are used:

- an item containing errors that should be resolved before creating a deployment package;
- an item containing changes that may lead to problems (e.g. removing of system-critical resources);
- an item is either a missing link or contains missing links.
The actions for adding, editing and deleting the file system changes as well as copying and moving those changes between projects are available in the **File System** view pop-up menu and on the contextual **File System** Ribbon page.

**Functions Overview**

| Changes Management | From the **File System** view, you can create, edit and delete the modifications to be performed by a deployment package to the file system. The actions for creating a new drive, system folder, folder, file and shortcut can be found within the **New** group on the contextual File System Ribbon page. The **System Folder** and **Drive** buttons from the **New** group on the **Project** page can also be used to create a new system folder and drive, respectively, together with the **New System Folder** and **New Drive** items in the pop-up menu displayed on an empty selection in the **File System** view. When a drive or folder is selected in the **File System** view, you can use the **New** sub-menu of the pop-up menu to create a new folder, file or shortcut in the selected location. To change any file system modification, you can use the **Edit** item from the pop-up menu or the **Edit** button from the **Management** group on the contextual **File System** Ribbon page, and to delete the modification, use the **Delete** items. |
| System Folders | Within the **File System** view, it is possible to replace the common system folders with their definitions and vice-versa. The system folder definition is expanded to the actual path when a deployment package is being installed on a target PC. To replace a path, representing a system folder, with its definition placeholder, you can use the **Roll System Folder** menu item from the pop-up menu or the **Roll** button from the **System Folders** group on the contextual **File System** Ribbon page. To expand a system folder definition to the path that is defined on a current PC, use the **Unroll System Folder** menu item from the pop-up menu or the **Unroll** button from the **System Folders** group on the contextual **File System** Ribbon page. It is also possible to roll and unroll all system folders within a project using the **Roll All** and **Unroll All** buttons from the **System Folders** group on the **Project** Ribbon page. |
| Copy/Move | You can easily copy and/or move the modifications to be performed by a deployment package to the file system from the **File System** view. You can use the drag/drop and copy/paste techniques as well as the **Cut**, **Copy** and **Paste** menu items to reach the goal. It is also possible to use the **Copy To** and **Move To** items available both in the pop-up menu and on the contextual **File System** Ribbon page to perform copy/move immediately choosing a target project in a dialog. |
| External Tools | From the **File System** view, it is possible to open a location of any file system item in the Windows Explorer using the **Jump to Explorer** item from the pop-up menu, when a file system item is selected. When a file is selected, you can open this file using the **Open File** item from the pop-up menu. You can also import changes from a file system of the PC MSI Package Builder is running on. To perform import, you should either use the items available in the **Import** sub-menu of the pop-up menu displayed on an empty selection and from the **Import** group on the contextual **File System** Ribbon page, or the **File System** button from the Import group on the **Project** Ribbon page. |
Filters' Settings  Any file system item from the **File System** view can be easily added either to monitoring or uninstall file filters using the **Apply As Filter** sub-menu from the **File System** view pop-up menu when a file or folder is selected.

Presentation  The changes to a file system performed by a deployment package can be displayed in the **File System** view in form of a tree or in form of a table. To switch between different data presentation models, you can use either the **Tree** and **Flat** buttons from the **Presentation** group on the contextual **File System** Ribbon page, or the **Presentation** sub-menu of the view pop-up menu displayed on an empty selection.

Search  Within the **File System** view, you can execute a search for specific changes to the file system performed by a deployment package using the **Find** item from the pop-up menu.

For detailed information on the changes to a file system that can be defined in a project, refer to the **File System Modifications** section of this document.
Registry View

The Registry view is displayed within the main program area when the Registry node of any project is selected in the Projects view. This view is used to define and review the changes to be performed to the Windows registry when installing a deployment package created on a basis of this project [Pic 1].

![Pic 1. The Registry view](image)

The registry modifications can include modifications of keys and values. The location of each element is represented with a registry path, that consists of root key and several keys included into each other. When monitoring existing installations, the required changes are created automatically and added to the corresponding project representing the installation process. It is also possible to create the changes manually or import changes from a registration entries (.reg) file.

| Tree | The Tree button from the Presentation group of the contextual Registry tab should be used to display the registry modifications in form of a tree, similar to Registry Editor. |
| Flat | The Flat button from the Presentation group of the contextual Registry tab allows you to display the registry modifications as a raw list, where each element is represented with its full path. |
Information on the registry modifications can be represented both in form of a keys tree with data on each key and in form of a resource list. When displayed as a resource list, it is by default grouped by processes responsible for the creation of each entry. For the entries created manually, such a procedure is not applicable. You can clear the grouping using the table features, and it is possible to revert to the default layout at any time using the Reset Layout item from the configuration menu.

The icon next to every item represented in the Registry view is used to describe of the item's type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not, and if there are any processing problems.

Below is the list of icons used to represent the item type and state:

- a root key;
- a root key containing permanent registry keys or values;
- a root key is being processed;
- a registry key;
- a registry key is a part of a permanent path;
- a registry key is being processed;
- a string registry value;
- a string registry value is permanent;
- a string registry value is being processed;
- a binary registry value;
- a binary registry value is permanent;
- a binary registry value is being processed;
- a DWORD registry value;
- a DWORD registry value is permanent;
- a DWORD registry value is being processed;
- a QWORD registry value;
- a QWORD registry value is permanent;
- a QWORD registry value is being processed;
- a multi-string registry value;
- a multi-string registry value is permanent;
- a multi-string registry value is being processed;
- an expandable string registry value;
- an expandable string registry value is permanent;
- an expandable string registry value is being processed.

The following overlays are used to represent the operation to be performed with a registry key or value:

- an item should be created;
- an item should be modified;
- an item should be deleted;
- permissions should be defined for a key.
As for the problematic situations, only the following overlay is used:

- an item containing changes that may lead to problems (e.g. removing of system-critical resources).

The actions for adding, editing and deleting the registry changes as well as copying and moving those changes between projects are available in the **Registry** view pop-up menu and on the contextual **Registry** Ribbon page.

### Functions Overview

<table>
<thead>
<tr>
<th>Changes Management</th>
<th>From the <strong>Registry</strong> view, you can create, edit and delete the modifications to be performed by a deployment package to the Windows registry. The actions for creating a new root key, registry key and registry values of different types can be found within the <strong>New</strong> group on the contextual <strong>Registry</strong> Ribbon page. The <strong>Root Key</strong> button from the <strong>New</strong> group on the <strong>Project</strong> page can also be used to create a new key together with the <strong>New Root Key</strong> item in the pop-up menu displayed on an empty selection in the <strong>Registry</strong> view. When a registry key is selected in the <strong>Registry</strong> view, you can use the <strong>New</strong> sub-menu of the pop-up menu to create a new key or a new value of a specific type within the selected key. To change any registry modification, you can use the <strong>Edit</strong> item from the pop-up menu or the <strong>Edit</strong> button from the <strong>Management</strong> group on the contextual <strong>Registry</strong> Ribbon page, and to delete the modification, use the <strong>Delete</strong> items. As for the default values for each key, you can also change the value type using the <strong>Change Type</strong> sub-menu from the <strong>Registry</strong> view when a default value is selected.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy/Move</td>
<td>You can easily copy and/or move the modifications to be performed by a deployment package to the Windows registry from the <strong>Registry</strong> view. You can use the drag/drop and copy/paste techniques as well as the <strong>Cut, Copy and Paste</strong> menu items to reach the goal. It is also possible to use the <strong>Copy To</strong> and <strong>Move To</strong> items available both in the pop-up menu and on the contextual <strong>Registry</strong> Ribbon page to perform copy/move immediately choosing a target project in a dialog.</td>
</tr>
<tr>
<td>External Tools</td>
<td>From the <strong>Registry</strong> view, it is possible to open a location of any key or value in the <strong>Registry Editor</strong> using the <strong>Jump to Registry Editor</strong> item from the pop-up menu, when a key or value is selected. You can also import the changes to a Windows registry from a registration entries (.reg) file. To perform import, you should either choose the <strong>Registry</strong> button available in the <strong>Import</strong> group on both the regular <strong>Project</strong> and contextual <strong>Registry</strong> Ribbon pages, or choose the <strong>Import</strong> item from the pop-up menu displayed on an empty selection.</td>
</tr>
<tr>
<td>Filters' Settings</td>
<td>Any key or value from the <strong>Registry</strong> view can be easily added either to monitoring or uninstall registry filters using the <strong>Apply As Filter</strong> sub-menu from the <strong>Registry</strong> view pop-up menu when a key or value is selected.</td>
</tr>
<tr>
<td>Presentation</td>
<td>The changes to a Windows registry performed by a deployment package can be displayed in the <strong>Registry</strong> view in form of a tree or in form of a table. To switch between different data presentation models, you can use either the <strong>Tree</strong> and <strong>Flat</strong> buttons from the <strong>Presentation</strong> group on the contextual <strong>Registry</strong> Ribbon page, or the <strong>Presentation</strong> sub-menu of the view pop-up menu displayed on an empty selection.</td>
</tr>
<tr>
<td>Search</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>Within the Registry view, you can execute a search for specific changes to the Windows registry performed by a deployment package using the Find item from the pop-up menu.</td>
<td></td>
</tr>
</tbody>
</table>

For detailed information on the changes to a registry that can be defined in a project, refer to the Registry Modifications section of this document.
Environment Variables View

The Environment Variables view is displayed within the main program area when the Environment Variables node of any project is selected in the Projects view. This view is used to define and review the changes to be performed to the environment variables when installing a deployment package created on a basis of this project [Pic 1].

[Pic 1. The Environment Variables view]
The environment variables modifications can include modifications to user and system variables. When monitoring existing installations, the required changes are created automatically and added to the corresponding project representing the installation process. It is also possible to create the changes manually.

The environment variables modifications are by default grouped by the variable type. You can clear the grouping using the table features, and it is possible to revert to the default layout at any time using the Reset Layout item from the configuration menu.

The icon next to every item represented in the Environment Variables view is used to describe of the item's type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not.

Below is the list of type icons used:
- a user environment variable;
- a user environment variable is being processed;
- a system environment variable;
- a system environment variable is being processed.

The following overlays are used to represent the operation to be performed with each variable when installing a generated deployment package:
- a variable should be created;
- a variable should be created, if it does not exist;
- a variable should be removed;
- a variable should be removed, if its value matches the defined value.

The actions for adding, editing and deleting the changes to environment variables as well as copying and moving those changes between projects are available in the Environment Variables view pop-up menu and on the contextual Environment Variables Ribbon page.

Functions Overview

| Changes Management | From the Environment Variables view, you can create, edit and delete the modifications to be performed by a deployment package to the environment variables. The New Environment Variable item from the Environment Variables view pop-up menu as well as the Environment Variable button from the New group on the contextual Environment Variables Ribbon page can be used to create a new modification to environment variables. It is possible to create a new variable, append/prepend data to existing variables and remove variables together with a deployment package deployment process. To change any environment variables modification, you can use the Edit item from the pop-up menu or the Edit button from the Management group on the contextual Environment Variables Ribbon page, and to delete the modification, use the Delete items. |

57
| **Copy/Move** | You can easily copy and/or move the modifications to be performed by a deployment package to the environment variables from the Environment Variables view. You can use the drag/drop and copy/paste techniques as well as the Cut, Copy and Paste menu items to reach the goal. It is also possible to use the Copy To and Move To items available both in the pop-up menu and on the contextual Environment Variables Ribbon page to perform copy/move immediately choosing a target project in a dialog. |
| **Search** | Within the Environment Variables view, you can execute a search for specific changes to the environment variables performed by a deployment package using the Find item from the pop-up menu. |

For detailed information about the changes to environment variables that can be defined in a project refer to the Environment Variables Modifications section of this document.
Services View

The Services view is displayed within the main program area when the Services node of any project is selected in the Projects view. This view is used to define and review the changes to be performed to the installed services when installing and/or uninstalling a deployment package created on a basis of this project [Pic 1].

MSI Package Builder allows you to create, delete and control Windows services. When monitoring existing installations, the required changes are created automatically and added to the corresponding project representing the installation process. It is also possible to create the changes manually.

The icon next to every item represented in the Services view is used to describe of the item’s type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not, and if there are any processing problems.

Below is the list of type icons used:

- a service;
- a service is being processed;
- a service containing errors that should be resolved before creating a deployment package.
The following overlays are used to represent the operation to be performed with each service:

- a service should be created;
- a service should be deleted;
- a service should be controlled (started and/or stopped);
- a service should be restarted;
- permissions should be defined for a service.

The actions for adding, editing and deleting the changes to Windows services as well as copying and moving those changes between projects are available in the Services view pop-up menu and on the contextual Services Ribbon page.

Functions Overview

<table>
<thead>
<tr>
<th>Changes Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>From the Services view, you can create, edit and delete the modifications to be performed by a deployment package to the Windows services configuration. The New Service item from the Services view pop-up menu and the Service button from the New group on the contextual Service Ribbon page can be used to create a new modification to Windows services. It is possible to create the service together with the package deployment process or control any service that is already installed when installing and/or uninstalling the deployment package. To change any service modification, you can use the Edit item from the pop-up menu or the Edit button from the Management group on the contextual Services Ribbon page, and to delete the modification, use the Delete items.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Copy/Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>You can easily copy and/or move the modifications to be performed by a deployment package to the Windows services configuration from the Services view. You can use the drag/drop and copy/paste techniques as well as the Cut, Copy and Paste menu items to reach the goal. It is also possible to use the Copy To and Move To items available both in the pop-up menu and on the contextual Services Ribbon page to perform copy/move immediately choosing a target project in a dialog.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Within the Services view, you can execute a search for specific changes to the Windows services configuration performed by a deployment package using the Find item from the pop-up menu.</td>
</tr>
</tbody>
</table>

For detailed information on the changes to Windows services that can be defined in a project, refer to the Services Modifications section of this document.
Assemblies View

The Assemblies view is displayed within the main program area when the Assemblies node of any project is selected in the Projects view. This view is used to configure a set of side-by-side .NET and/or Win32 assemblies that are installed by the generated deployment package [Pic 1].

This view is used to configure a set of side-by-side .NET and/or Win32 assemblies that are installed by the generated deployment package.

The assemblies are displayed in form of a table, where each row represents a single assembly. The icon next to every assembly is used to describe of the assembly's type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not and if there are any problems.

Below is the list of icons used to represent the item type and state:

- a .NET assembly;
- a .NET assembly is being processed;
- a Win32 assembly;
- a Win32 assembly is being processed.
As for the problematic situations, the following overlays are used:

- an assembly item containing errors that should be resolved before creating a deployment package;
- an assembly item contains missing links.

The actions for adding new assemblies, editing and deleting existing ones and others are available in the **Assemblies** view pop-up menu and on the contextual **Assemblies** Ribbon page.

### Functions Overview

| **Assemblies Management** | From the **Assemblies** view, you can add, edit and delete the .NET and/or Win32 side-by-side assemblies to be installed by a generated deployment package. To add a new assembly, you can either choose the **New Win32 Assembly/New .NET Assembly** item from the pop-up menu, or press the **Win32 Assembly/.NET Assembly** button from the **New** group on the contextual **Assemblies** Ribbon page and on the **Project** Ribbon page. The **Edit** item from the pop-up menu, as well as the **Edit** button from the **Management** group on the contextual **Assemblies** Ribbon page can be used to change the selected assembly, and to delete any assembly, you can use the **Delete** items. |
| **Copy/Move** | You can easily copy and/or move the .NET and/or Win32 side-by-side assemblies to be installed by a generated deployment package from the **Assemblies** view to another project. You can use the drag/drop and copy/paste techniques as well as the **Cut**, **Copy** and **Paste** menu items to reach the goal. It is also possible to use the **Copy To** and **Move To** items available both in the pop-up menu and on the contextual **Assemblies** Ribbon page to perform copy/move immediately choosing a target project in a dialog. |
| **Search** | Within the **Assemblies** view, you can execute a search for specific assembly using the **Find** item from the pop-up menu. |

For detailed information on the side-by-side assemblies and their management process, refer to the **Side-by-side Assemblies Deployment** section of this document.
Drivers View

The Drivers view is displayed within the main program area when the Drivers node of any project is selected in the Projects view. This view is used to define and review the changes to be performed to the device drivers when installing and/or uninstalling a deployment package created on a basis of this project [Pic 1].

MSI Package Builder allows you to create and delete basic drivers as well as install and pre-install driver packages. Both basic drivers and drivers from packages can be registered as device class filters by MSI Package Builder if it is required. When monitoring existing installations, the required changes are created automatically and added to the corresponding project representing the installation process. It is also possible to create the changes manually.

The icon next to every item represented in the Drivers view is used to describe of the item's type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not, and if there are any processing problems.

Below is the list of type icons used:

- a basic driver;
- a basic driver is being processed;
- a driver package;
- a driver package is being processed;
- a DIFx driver package;
- a DIFx driver package is being processed.

As for the problematic situations, the following overlays are used:

- a basic driver containing errors that should be resolved before creating a deployment package;
- a driver package containing errors that should be resolved before creating a deployment package;
- a driver package contains missing links;
- a DIFx driver package contains missing links.

The following overlays are used to represent the operation to be performed with each driver:

- a basic driver should be created;
- a basic driver should be deleted;
- permissions should be defined for a basic driver;
- a driver package should be installed;
- a driver package should be pre-installed;
- a DIFx driver package should be installed;
- a DIFx driver package should be pre-installed.

The actions for adding, editing and deleting the changes to device drivers as well as copying and moving those changes between projects are available in the Drivers view pop-up menu and on the contextual Drivers Ribbon page.

Functions Overview

<table>
<thead>
<tr>
<th>Changes Management</th>
<th>From the Drivers view, you can create, edit and delete the modifications to be performed by a deployment package to the device drivers configuration. The New Basic Driver, the New Driver Package and the New DIFx Driver Package items from the Drivers view pop-up menu, as well as the Basic Driver, the Driver Package and the DIFx Driver Package buttons from the New group on the contextual Drivers Ribbon page can be used to create a new modification to device drivers. To change any device drivers modification, you can use the Edit item from the pop-up menu or the Edit button from the Management group on the Drivers contextual Ribbon page, and to delete the modification, use the Delete items.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy/Move</td>
<td>You can easily copy and/or move the modifications to be performed by a deployment package to the device drivers configuration from the Drivers view. You can use the drag/drop and copy/paste techniques as well as the Cut, Copy and Paste menu items to reach the goal. It is also possible to use the Copy To and Move To items available both in the pop-up menu and on the contextual Drivers Ribbon page to perform copy/move immediately choosing a target project in a dialog.</td>
</tr>
<tr>
<td><strong>Search</strong></td>
<td>Within the <strong>Drivers</strong> view, you can execute a search for specific changes to the device drivers configuration performed by a deployment package using the <strong>Find</strong> item from the pop-up menu.</td>
</tr>
</tbody>
</table>

For detailed information on the changes to device drivers that can be defined in a project, refer to the **Drivers Deployment** section of this document.
Printers View

The **Printers** view is displayed within the main program area when the **Printers** node of any project is selected in the **Projects** view. This view is used to define and review the changes to be performed to the printing system when installing and/or uninstalling a deployment package created based on this project [Pic 1].

![Pic 1. The Printers view](image)
MSI Package Builder allows you to perform the following changes to the printing system: create, modify and delete printers; install and uninstall printer drivers; add and remove print processors, print monitors and printer ports. When monitoring existing installations, required changes are created automatically and added to the corresponding project representing the installation process. It is also possible to create changes manually.

By default, items in the view are grouped by the entry type. You can change the grouping using the **table features**, and it is possible to revert to the default layout at any time using the **Reset Layout** item from the configuration menu.

The icon next to every item represented in the **Printers** view is used to describe the item's type and state. The state icons are provided to help you understand what is happening in the program at the moment. You can always see if the item is being processed now by some operation or not, and if there are any processing problems.

Below is the list of the type icons used:

- a printer;
- a printer is being processed;
- a printer driver;
- a printer driver is being processed;
- a print processor;
- a print processor is being processed;
- a print monitor;
- a print monitor is being processed;
- a local printer port;
- a local printer port is being processed;
- a TCP/IP printer port;
- a TCP/IP printer port is being processed.

In case of problems, the following overlays are used:

- a printer containing errors that should be resolved before creating a deployment package;
- a printer driver containing errors that should be resolved before creating a deployment package;
- a print processor containing errors that should be resolved before creating a deployment package;
- a print monitor containing errors that should be resolved before creating a deployment package;
- a printer driver contains missing links;
- a print processor contains missing links.

The following overlays are used to represent the operation to be performed with each item:

- a printer should be created;
- a printer should be modified;
- a printer should be deleted;
- a printer driver should be installed;
- a printer driver should be uninstalled;
- a print processor should be added;
- a print processor should be removed;
- a print monitor should be added;
- a print monitor should be removed;
- a local printer port should be created;
- a local printer port should be deleted;
- a TCP/IP printer port should be created;
- a TCP/IP printer port should be modified;
- a TCP/IP printer port should be deleted.

The actions for adding, editing and deleting the changes to the printing system as well as copying and moving those changes between projects are available in the Printers view pop-up menu and on the Printers contextual Ribbon page.

Functions Overview

<table>
<thead>
<tr>
<th>Changes Management</th>
<th>From the Printers view you can create, edit and delete the modifications to be performed by a deployment package to the printing system configuration. The actions for creating changes in printers, printer drivers, print processors, print monitors and printer ports can be found within the New group on the Printers contextual Ribbon page and in the New group of the Printers view pop-up menu. Alternatively you can use the Printer Entries drop-down button from the New group on the Project Ribbon page. To change any modification to printing system, you can use the Edit item from the pop-up menu or the Edit button from the Management group on the Printers contextual Ribbon page, and to delete the modification, use the Delete items.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy/Move</td>
<td>You can easily copy and/or move the modifications to be performed by a deployment package to the printing system configuration from the Printers view. You can use the drag/drop and copy/paste techniques as well as the Cut, Copy and Paste menu items to reach the goal. It is also possible to use the Copy To and Move To items available both in the pop-up menu and on the Printers contextual Ribbon page to perform copy/move immediately choosing a target project in a dialog.</td>
</tr>
<tr>
<td>Search</td>
<td>Within the Printers view you can execute a search for specific changes to the printing system configuration performed by a deployment package using the Find item from the pop-up menu.</td>
</tr>
</tbody>
</table>

For detailed information on changes to the printing system that can be defined in a project, refer to the Printers Deployment section of this document.
Custom Actions View

The **Custom Actions** view is displayed within the main program area when the **Custom Actions** node of any project is selected in the **Projects** view. This view is used to provide the actions to be executed together with the generated deployment package before and/or after the package install and uninstall process as well as specific system actions [Pic 1].

![Pic 1. The Custom Actions view](image)

The pre & post actions are displayed in form of a table, where each row represents a single action. The icon next to every action is used to describe of the action's type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not. Below is the list of icons used:

- an action to be executed before a deployment package installation;
- an action to be executed after a deployment package installation;
- an action to be executed before a deployment package uninstallation;
- an action to be executed after a deployment package uninstallation;
- an action is being processed;
- a SAM licenses installation;
- a SAM licenses installation is being processed;
- an action to pin an application to the Task Bar and/or Start Menu;
- an action to unpin an application from the Task Bar and/or Start Menu;
- a Pin Application action is being processed.

The system actions are also displayed in form of a table. With the help the system actions, you can pin applications to or unpin them from the Start Menu and the Task Bar. Another available system action is the SAM License Installation action. It allows you to install Software Assets Management (SAM) licenses to the Software Licensing Service (SLS) when deploying MSI packages.

The actions for creating new actions, editing and deleting existing ones and copying/moving actions to another project are available in the Custom Actions view pop-up menu and on the contextual Custom Actions Ribbon page.

**Functions Overview**

<table>
<thead>
<tr>
<th>Actions Management</th>
<th>From the Custom Actions view, you can create, edit and delete the actions to be performed before and after a generated deployment package deployment. To create a new action, you can either choose the New Action item from the pop-up menu, or press the Action button from the New group on the contextual Custom Actions Ribbon page and on the Project Ribbon page. The Edit item from the pop-up menu, as well as the Edit button from the Management group on the contextual Custom Actions Ribbon page can be used to change the selected action, and to delete any action you can use the Delete items.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy/Move</td>
<td>You can easily copy and/or move the actions to be performed before and after a generated deployment package deployment from the Custom Actions view to another project. You can use the drag/drop and copy/paste techniques as well as the Cut, Copy and Paste menu items to reach the goal. It is also possible to use the Copy To and Move To items available both in the pop-up menu and on the contextual Custom Actions Ribbon page to perform copy/move immediately choosing a target project in a dialog.</td>
</tr>
<tr>
<td>Search</td>
<td>Within the Custom Actions view, you can execute a search for specific action using the Find item from the pop-up menu.</td>
</tr>
</tbody>
</table>

For detailed information on the custom actions management process, refer to the Using Custom Actions section of this document.
Wrapped Packages View

The Wrapped Packages view is displayed within the main program area when the Wrapped Packages node of any project of the corresponding type is selected in the Projects view. This view is used to configure a set of packages that are deployed together with the generated MSI package Pic 1.

The wrapped packages are displayed in form of a table, where each row represents a single package. The icon next to every package is used to describe of the package’s type and state. The state icons are provided to help you understand what is currently happening in the program. You can always see if the item is being processed now by some operation or not and if there are any problems. Below is the list of icons used:

- an executable package;
- an executable package is being processed;
- an executable package containing files that cannot be accessed;
- a Windows Installer package;
- a Windows Installer package is being processed now;
- a Windows Installer package containing files that cannot be accessed.
The actions for adding wrapped packages, editing and deleting existing ones, changing the deployment order and others are available in the **Wrapped Packages** view pop-up menu and on the contextual **Wrapped Packages** Ribbon page.

**Functions Overview**

<table>
<thead>
<tr>
<th>Packages Management</th>
<th>From the <strong>Wrapped Packages</strong> view, you can create, edit and delete the installation packages to be deployed together with a generated MSI package. To create a new package, you can either choose the <strong>New Wrapped Package</strong> item from the pop-up menu, or press the <strong>Wrapped Package</strong> button from the <strong>New</strong> group on the contextual <strong>Wrapped Packages</strong> Ribbon page and on the <strong>Project</strong> Ribbon page. The <strong>Edit</strong> item from the pop-up menu, as well as the <strong>Edit</strong> button from the <strong>Management</strong> group on the contextual <strong>Wrapped Packages</strong> Ribbon page can be used to change the selected wrapped package, and to delete any package, you can use the <strong>Delete</strong> items. To change the packages deployment order, you can use the <strong>Move Up</strong> and <strong>Move Down</strong> items available in the pop-up menu and in the <strong>Order</strong> group from the contextual <strong>Wrapped Packages</strong> Ribbon page.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy/Move</td>
<td>You can easily copy and/or move the packages to be deployed together with a generated MSI package from the <strong>Wrapped Packages</strong> view to another project. You can use the drag/drop and copy/paste techniques as well as the <strong>Cut</strong>, <strong>Copy</strong> and <strong>Paste</strong> menu items to reach the goal. It is also possible to use the <strong>Copy To</strong> and <strong>Move To</strong> items available both in the pop-up menu and on the contextual <strong>Wrapped Packages</strong> Ribbon page to perform copy/move immediately choosing a target project in a dialog.</td>
</tr>
<tr>
<td>Search</td>
<td>Within the <strong>Wrapped Packages</strong> view, you can execute a search for specific installation package using the <strong>Find</strong> item from the pop-up menu.</td>
</tr>
</tbody>
</table>

For detailed information on the wrapped packages and packages management process, refer to the **Wrapping Existing Installations** section of this document.
Log View

The **Log** view is designed to display information on the events taking place during the program execution. The larger part of this information consists of events generated by the operations and the operations results.

![Image of Log view](Image)

The **Log** view is located by default at the bottom of the MSI Package Builder main window and displays the log in form of a tree. The description for any logged event is by default wrapped, so that you can easily read it. If you would like to have more events visible at the same time, you can configure the **Log** view to display only one line per event by disabling the **Wrap Text** option from the **Configuration** menu.

Every event in the **Log** is assigned a severity level represented by a certain icon. The icon allows you to see if the operation has finished successfully without reading the description. The following icons are available:

- the blue icon with an 'i' character means that everything is OK;
- the brown circle icon with a cross-cut line is used to identify that the task was canceled by user or due to shutdown of the underlying system;
- the yellow icon with an exclamation mark is the warning sign: it tells you that some errors have occurred, but they are not critical. In such a case, there is no guarantee that the operation has actually succeeded;
- the red icon with a white cross is the error sign: it means that the operation execution has failed.

Analyzing the Log can help you a lot in your everyday work with MSI Package Builder, because this log contains all the information on the executed operations and provides you with troubleshooting recommendations in case any problems are detected.

Functions Overview

<table>
<thead>
<tr>
<th>Export All</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Export All button should be used to export the log to the CSV file.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clear</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Clear button should be used to remove all the logged events from the program.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Full Expand</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Full Expand button from the Log view toolbar should be used to expand all nodes in the table of logged events.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Full Collapse</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Full Collapse button from the Log view toolbar should be used to collapse all nodes in the table of logged events.</td>
</tr>
</tbody>
</table>

The options of clearing the log, expanding nodes in the tree of logged events and collapsing them are also available from the pop-up menu of the Log tree. The layout of the Log view, including the visible columns, the column widths, the sorting settings and the text wrapping, is saved between sessions.
Operations View

The Operations view [Pic 1] shows the detailed progress of each operation being performed at the moment and allows cancelling a particular operation or all running operations. By default, it is located at the bottom of the MSI Package Builder main window.

Pic 1. The Operations view

Progress information for every operation is shown in the pane with the progress bar, the operation information text and the Cancel button. The Cancel button is used to cancel individual running operations, whereas if the grouping operation is canceled, all the sub-operations are also canceled.

Cancel All

The Cancel All button from the Operations view toolbar can be used to cancel all the operations running in the application.

You can cancel all the running operations by clicking the Cancel All button on the toolbar of the Operations view.
Graphical User Interface features

EMCO Software provides you with a modern and intuitive graphical user interface, because we appreciate the users of our products and would like them to feel glad that they have EMCO programs installed on their PCs. Lots of resources were involved in creating this kind of an interface for you, and now we are proud we have done it. Custom DPI settings are fully supported, so that you can use EMCO programs on any display with any resolution you like. The *Microsoft User Interface Guidelines on Layout, Icons and Sizing* have been a powerful base for this work, and we are glad to tell you that they are fully complied with and supported. With the help of the skinning support and the Ribbon UI interface, every customer can configure the program UI to feel comfortable during each working day. EMCO also provides you with the High Contrast skin along with the bonus skins pack, which is an accessibility feature designed for people with vision impairment. The High Contrast color scheme can increase legibility for some users by heightening the screen contrast with alternative color combinations.

This chapter gives you a detailed description of how to fully enjoy the graphical user interface features, the skinning mechanism and the Ribbon bar features.
Skinning

MSI Package Builder provides you with a wide range of custom skins with unique look and feel, so that you can choose any skin you like most. If you are a fan of the Microsoft Office interface, you have no reason to complain either, since MSI Package Builder also gives you an option of choosing this type of skin. There are not only formal skins but also some informal ones.

All the skins can be divided into four groups: Office Skins, Custom Design Skins, Bonus Skins and Skins for Fun. The following skins are available:

<table>
<thead>
<tr>
<th>Skin Type</th>
<th>Skins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skins for Fun:</td>
<td>&quot;Christmas&quot;, &quot;Valentine&quot;, &quot;Summer&quot;, &quot;Springtime&quot;.</td>
</tr>
</tbody>
</table>

Let us take a brief look at some of the skins:

Office 2010 Skin look and feel example
Custom Design Skin look and feel example
Skins for Fun look and feel example
High Contrast Skin look and feel
Ribbon

Ribbon is a tool that presents commands organized into a set of tabs. The tabs on the Ribbon represent commands that are most relevant for each of the task areas in the program. For example, in Office Word the tabs group commands by activities such as inserting objects like pictures and tables, doing page layout, working with preferences, doing mailings, and reviewing. The Home tab provides an easy access to the most frequently used commands. Office Excel has a similar set of tabs that make sense for spreadsheet work including tabs for working with formulas, managing data, and reviewing. Those tabs simplify access to the program features, because they organize the commands in a way that reflects the tasks people perform in those programs.

Pic 1. The Ribbon bar

We are delighted to let you know that we fully conform with Microsoft® Ribbons Guidelines and would like to introduce some Ribbon features to you. To learn more about Ribbon, the story of its development and its usability features, you may visit 'The Story of the Ribbon' article from the MSDN blogs.
Application Menu

The **Application Menu** invoked using the **Application** button is somewhat similar to the File menu in most programs that use a classic user interface, but it gives you more.

It contains links to the most helpful actions located under different tabs but made available from one access point. Also you can open the recently used projects using the links from the Recent Projects part.

Quick Access Toolbar

The **Quick Access Toolbar** is an end-user customizable bar located near the **Application Menu** or below the Ribbon bar depending on the configuration. It can contain links to both Ribbon items and Ribbon groups.

To add an action link to the quick access toolbar, right click this action and select **Add to Quick Access Toolbar** from the pop-up menu. The groups can be added in the same way, the only difference being that to add a group you should right click its caption.
Representation and Navigation Features

The representation of the Ribbon bar can also be configured to make your work more comfortable. You can minimize Ribbon so that the tab's content is only shown when the tab is clicked on, thus extending the program workspace. Also, if it is not convenient for you to have the Quick Access Toolbar next to the Application Menu, you may place it below the Ribbon bar, so that it will look just like a simple toolbar. This configuration can be accessed from the pop-up menu of the Ribbon bar Pic 4.

Navigation between the Ribbon tabs can be performed not only with a mouse click on a tab but also with the help of the mouse wheel. Just place the cursor over any tab and scroll the wheel – scrolling up will switch the tabs from right to left, and scrolling down will switch the tabs in the opposite direction.
UI Elements' main features

The graphics shell used for MSI Package Builder is aimed at providing a high level of usability to everyone. This topic covers main features of the graphical elements used in this program, and here you can find what puts EMCO GUI a step ahead of the others.

Docking

The MSI Package Builder user interface is built using the ultimate docking technology which provides for the maximum use of the program working area. It allows docking the windows that are used less often than the main one to the sides, auto hide them or even close and then open again when required. The dock panels can be docked both to the main window and to each other, thus enabling you to build such a subsidiary window layout that makes you feel comfortable while working with MSI Package Builder [Pic 1].

To change the position of any dock window, you should click its header and move the mouse pointer while holding the left mouse button down. Hint windows are shown to help you understand where you can drop the window dragged. When you are dragging it over another dock window, it is possible to dock both windows to each other or display them in different tabs of the same dock window.

To enable the auto hide feature for a window attached to any side of a main window, click the pin button in the dock window header. Clicking the cross button results in closing of the dock window. Each view can also be closed and opened again using the checkboxes in the Show Ribbon group accessible from the Application page.
Grouping and Filtering Data

The MSI Package Builder user interface is designed so as to make its usage as flexible as possible. The tables available in every EMCO program provide you with an easy-to-use data filtration and grouping mechanism. To group data by one of the columns, you should drag its header to the grouping box displayed over the table or choose an appropriate item from the column header pop-up menu [Pic 2].

To group or ungroup data by any column when Group By Box [Pic 2] is not visible, you can have it displayed by selecting the Show Group By Box item from the pop-up menu of any column header.

Data filtration can be performed in two ways: by using the quick filter or the filter editor. To use the quick filter feature, just click on the glyph in the right top corner of any column header. A drop-down list appears offering you to choose one of the predefined filters or select the custom one from the filtering dialog [Pic 3].
The **Filter Editor** shipped with MSI Package Builder is easy to use and allows you to build your own complex filters quickly and easily [Pic 4]. To open the filter editor, choose the **Filter Editor** item from the column's pop-up menu.

**Pic 4. Using the filter editor**

You can enable and disable the currently applied filter condition using the checkbox displayed next to the filter condition in the bottom of the view, inside the filter info pane [Pic 5].

**Pic 5. The filter info pane**

To reset the currently applied filter use the **X** button from the filter info pane, and to customize it use the **Customize** button from this pane.
Managing Columns in Trees and Tables

You can customize almost every table and tree in EMCO programs by moving and removing columns to make the control most informative for you. To move a column, drag it over the control's header and drop between other columns to its new position [Pic 6].

To remove a column that is of no use for you, right click the control's header and select the Remove This Column item from the pop-up menu. Also, you can control columns availability using the column chooser [Pic 7].

To show the column chooser, right click the control's header and select the Column Chooser menu item. After that, you can drag and drop columns from the header to the column chooser and backwards.
Automatic Saving and Restoring of Windows Layout

One of the service functions of MSI Package Builder user interface is its ability to save and restore the windows layout. All the changeable parameters like the windows sizes and positions; the table columns order, sizes and positions; the grouping and filtering options; the dock windows configuration, etc. are saved between sessions. Thus, you do not need to configure the program's user interface layout every time you start this program.
Chapter 4: Installations Repackaging

The main goal of MSI Package Builder is to deliver to you an easy-to-use installations repackaging tool. We have designed and developed two different repackaging models, which are monitoring operating system changes and wrapping existing installations. You can choose between smart repackaging, which allows you to easily create a deployment package based on existing installations by simply following the steps of the **Repackage Installation** wizard, and low-level repackaging, which allows you to control each aspect of the repackaging process.

The **Repackage Installation** wizard was designed to make the repackaging process as intuitive as it is possible. By default, it is displayed on the program startup, so you can repackage existing installation with no additional actions required. If you do not need it to be displayed on each startup, you can disable this option other on the wizard welcome page, or on the **User Interface** preference page.

![Repackage Installation Wizard](image)

**Pic 1. Choosing a repackaging mode**
While going through the steps of the **Repackage Installation** wizard, you are choosing the repackaging mode you would like to use, configure a deployment package to be generated and a project to be created. The project containing the repackaging results is saved to the projects storage and can be used in the future to upgrade the installation package.

In this chapter we'll go through the available repackaging modes and take a closer look at each one, providing you with some tips and tricks on using the repackaging abilities shipped with MSI Package Builder.

**What's Inside**

- How to choose a Repackaging Mode?
- Automated Monitoring
- Advanced Monitoring
- Repackaging via Wrapping
- Low-Level Repackaging
- How should I configure the repackaged installations to support an upgrade?
How to choose a Repackaging Mode?

MSI Package Builder is shipped with the different repackaging modes to be used for different situations. It is possible to use the smart repackaging technique with the help of the **Repackage Installation** wizard to be able to create an MSI package immediately, or use the low-level repackaging to benefit from advanced repackaging abilities. As for smart repackaging, you can choose between automated installations monitoring, advanced monitoring of changes made to the operating system and wrapping of existing installations into an MSI package for simultaneous deployment. The low-level repackaging allows you to combine different repackaging techniques and to control each aspect of the repackaging process.

The repackaging through monitoring of file system and registry changes is the easiest mode for configuration, but it has a set of limitations. In this mode the deployment package is created on a basis of changes performed to the operating system by existing installations. You should choose this mode when the repackaged installations do not support silent execution or you cannot find out the parameters to be used for silent execution. Also this mode is to be used when you are going to create an MSI package based on generic changes, performed manually rather than those performed by existing installations.

While monitoring existing installations, it is recommended that the **Automated Monitoring** mode be used. Using this mode guaranties that the monitoring process will stop only after the installation is fully complete, thus all the changes performed by the installation are included into the resulting deployment package. To use this mode, the installation setup should be designed in such a manner, that all processes performing the application and operating system configuration exit on installation completion. If it is not possible to fulfill these requirements, or it is not enough to monitor only those changes that are performed by existing installations, you should use the **Advanced Monitoring** mode. In such a case, you are responsible for stopping the monitoring process when all required changes are performed.

The main prerequisite for repackaging via changes monitoring, is the fact that the monitoring should be performed in a clean environment. The main disadvantage of this mode is that the resulting deployment will not be fully cross-platform, thus the repackaging results will be valid only for the same operating system as used for monitoring.

The **Repackaging via Wrapping** mode is the recommended one for experienced users for creating MSI packages. Although it is a little more complicated for creating silent MSI packages, it allows you to avoid all the limitations of the monitoring technology. The installations wrapped into an MSI package will be distributed in the original state, thus deployed correctly to any operating system, regardless the deployment process particularities. It is the mode you should always choose if it is not required to generate the MSI package for silent deployment, but just distribute a package containing a set of existing installations. As for the packages for silent deployment, while using this mode, you must specify the parameters to be passed to the original installer application to be executed in a silent mode.

The low-level repackaging abilities allow you to reach the maximum flexibility in repackaging, but should be used by experienced users only. In this mode, you can change the monitored data and provide the required changes manually. This mode is the most complex, because you are responsible for preparing the correct changes configuration for creating a deployment package. Because of the complexity of generic repackaging process it is required to fully understand each aspect of every step to be performed for successful repackaging.

You are now introduced to the repackaging modes included into MSI Package Builder and should be able to choose an appropriate one during the everyday work.
Automated Monitoring

MSI Package Builder allows you to automatically monitor the changes performed by a single installation or a set of installations and create a deployment package based on these changes. When multiple installations are monitored, they are launched one by one in the specified order. To perform automated monitoring you should choose the Automated Monitoring option in the Repackage Installation wizard.

![Pic 1. Providing installations to monitor](image-url)
Let us take a look at the automated monitoring process configuration. Firstly, you should specify a name for the project to be created on a basis of the monitored changes, provide a set of installations to be monitored and define the installation order [Pic 1]. You can define command-line parameters to be passed to the installer application for each installation while adding it. It is possible to monitor Windows Installer packages, Microsoft Software patches and executable installation packages.

Pic 2. Configuring MSI package
The next step is to provide a path to save a resulting MSI package to and configure the package. The detailed information about an MSI package configuration is available in the Creating MSI Package section of this document. After configuring the package itself you can already press **Finish** to proceed with the repackaging procedure. Optionally you can continue following the wizard to configure process filters used for monitoring.

As for the monitoring filters, you can define additional processes to be filtered choosing them from those currently running [Pic 3].

![Pic 3. Extending monitoring process filters](image)

A wrong processes monitoring filters configuration may lead to improper installation monitoring if the activity of the process required by the installation. You must not filter the processes when you are not sure if their activity can affect the installation results. Otherwise, the deployment package built as a result of such monitoring will not be able to perform the same installation.
It is also possible to add/remove any process from those currently running to/from the persistent monitoring filters, such as change the persistent monitoring process filters configuration using a corresponding hyperlink.

After you press **Finish** the monitoring process is started and as soon as it completes a deployment package is created containing the monitored changes in the specified location.

**When does the monitoring process stop?**

When a monitoring process is running, you can see the monitoring progress on a screen. It displays how many installations from those defined are already complete.

As soon as all the installations are complete, the monitoring process is stopped automatically. The installation is treated as completed when the installer process and all the processes daughterly to the installer process exit.

*Please note that if the setup wizard suggests running the installed application upon the installation completion, you should decline this suggestion. Otherwise, the monitoring module will wait for the started application to exit.*
You can abandon the monitoring process using the **Stop** button at any time, although it is not recommended.

**The automated monitoring process does not stop. Which option should I choose?**

You can face the situation, when you have passed through all the steps of the setup wizard, but the automated monitoring process does not stop. To choose further steps, it is required to understand what leads to this situation. Let us take a closer look at the list of possible reasons and appropriate solutions for each one.

Firstly, the monitoring process may not be stopping because the installation is still performing the application configuration in background. Thus, it is recommended to be patient and always wait for some time after the installer application user interface has closed.

Another possible reason is that the monitored installation has launched the installed application or one of its modules. This can happen if you have not unchecked an appropriate option on the installation wizard finish page, or if the installation always performs the step of launching the installed program. Please note that those executable modules can be run in the background or minimized to tray. In such a case, you should close the launched applications and continue waiting for the monitoring to stop automatically. The monitoring module will switch to next installation, if any, or create the deployment package, if this installation is the last one from those monitored, as soon as those applications are closed. In case if it is not possible to close the launched applications, but you are sure that the installation is fully complete, you can stop the monitoring process manually and create the deployment package based on the changes currently monitored by pressing the **Stop** button and then choosing the option to proceed with creation.

It is not recommended that the automated monitoring process be stopped manually to proceed with the deployment package creation unless you are positively sure the installation is fully complete. In case the setup is still in progress, the repackaging will be incomplete.

The last reason is that the installation does not fulfill the requirements for automated monitoring (the process performing the installation does not exit after the setup is complete). If you are not sure that the installation process is fully complete and cannot find out the processes that are preventing the automated monitoring from being completed, you can abort the monitoring process by pressing the **Stop** button and then choosing the **Abort Monitoring** option. All the monitored changes will be irreparably lost. In such a case it may be convenient for you to use repackaging via wrapping.
Advanced Monitoring

With MSI Package Builder you can track changes made to your operating system and create a deployment package based on these changes. With the help of the monitoring technology, you can easily create an MSI package from any installation you want and even on any changes you performed manually. Just start monitoring, perform the changes and stop monitoring – the new project created will contain all these changes. You can launch the advanced monitoring with an ability of creating a deployment package on completion choosing the Advanced Monitoring option within the Repackage Installation wizard.

Within the scope of the monitoring process configuration, first you are proposed to provide a name for the project to be created as the monitoring results. This project will contain all changes performed by the running processes to the operating system during the monitoring session.

Pic 1. Configuring MSI package
The next step is to provide a path to save a resulting MSI package to and configure the package. The detailed information about an MSI package configuration is available in the Creating MSI Package section of this document. After configuring the package itself, you can already press Finish to proceed with the repackaging procedure. Optionally you can continue following the wizard to configure process filters used for monitoring.

As for the monitoring filters, you can define additional processes to be filtered choosing them from those currently running Pic 2.

The wrong processes monitoring filters configuration may lead to improper installation monitoring if the activity of the process required by the installation to succeed is ignored. You must not filter the processes when you are not sure if their activity can affect the installation results. Otherwise, the deployment package built as a result of such monitoring will not be able to perform the same installation.
As soon as you press the Finish button, the monitoring process is started, and since that moment until the moment when you stop it, all changes except those skipped in view of the monitoring filters configuration are recorded and saved to the specified project. The deployment package based on these changes is created at the specified location.

**When should I stop the monitoring process?**

While using an advanced mode for monitoring changes, the monitoring process cannot be stopped automatically, because there is no trigger for that. Therefore, it is up to you to choose the moment when it should be stopped. Before stopping the monitoring process, you should make sure that all required changes are performed. It means that the installations you have monitored have exited, and all the operations have been fully completed, thus the required files are fully copied, moved, created or deleted, and required registry keys and values are fully saved to the windows registry, etc.

As soon as you stop the monitoring process, the project is created containing the recorded changes. If you are using the Repackage Installation wizard this project is automatically prepared and a deployment package is created. For the manual capturing mode, you should prepare the created project before building a deployment package. See the Projects Preparation section for the information about the required preparation steps.
Repackaging via Wrapping

One of the installations repackaging methods used by MSI Package Builder is wrapping existing installations into a generated MSI package. Wrapping means that those installations are distributed as a part of the MSI package and are installed, uninstalled and repaired simultaneously in the specified order. The advantage of this mode is that the original installation is always executed on each PC, so there should be no chance that a generated MSI package will be installed incorrectly on specific target machines. The disadvantage of this mode is that you need to know the silent installation parameters and possibly provide installation scenario files to generate an MSI package that can be used for silent deployment.

To perform repackaging via wrapping, either choose the Wrap Installation option in the Repackage Installation wizard or click the Wrap Installation shortcut in the Product Actions group on the Welcome Screen. In any case, you’ll be able to provide a set of installations to be wrapped and their deployment order Pic 1.
MSI Package Builder allows you to wrap both Windows Installer packages and executable installation packages. For each package, you can define if it should be repaired and uninstalled while repairing and uninstalling a generated MSI package. You can also specify a list of additional files, if required, and the parameters to be passed to the installer to perform install, uninstall and repair. The provided packages are included into a created MSI package and saved into a project with the specified name for future usage. The detailed information about the wrapped packages configuration is available in the Wrapping Existing Installations section of this document.

The next step is to provide a path to save a resulting MSI package to and configure the package. The detailed information about an MSI package configuration is available in the Creating MSI Package section of this document.

As soon as the package is configured, press Finish to proceed with its creation. The created package is saved to the specified location and is ready for deployment. The project representing the current repackaging process is also saved to the projects storage.

That is all you need to do to get an MSI package for deploying existing installations in their original state simultaneously.
Low-Level Repackaging

MSI Package Builder comes with a wide range of features available for creating standalone deployment packages and repackaging existing installations. Most of repackaging tasks can be easily solved using the smart repackaging modes available in the Repackage Installation wizard, but the program does not put any restrictions on the ways used to for building a deployment package. All the features available in the smart mode can be used both as standalone ones and in a combination with each other while performing a low-level repackaging. Let us take a closer look on what does the low-level repackaging mean and how is it possible to benefit from it.

The low-level repackaging concept should be used by experienced users only. While using a low-level repackaging you are responsible for performing each and every step required for the successful repackaging, so it is required that you fully understand those steps.

The advantage of the low-level repackaging is the fact that you are not limited by any single repackaging technique and can combine those techniques to reach the desired goal. For example, you can monitor the installations you want to repackage and then either provide additional changes to be performed to a target Machine during an MSI package deployment, or correct the monitoring results, if required.

When you are using the feature of capturing changes to the underlying operating system in a scope of advanced monitoring you must prepare the created project before creating a deployment package. Preparation consists of several important steps. Firstly, MSI Package Builder should copy all file system resources to a project storage. If there are any problems during the copying process, the problematic files and folders are marked as missing links. The problems with missing links must be resolved before proceeding with the package generation. For a deployment package to be correctly installed on different operating systems with different environment configurations, it is required to replace all paths that represent the system folders with a special system folder object. All those steps are described in detail within the Projects Preparation section of this document.

When providing the required changes manually, you just create a project to contain those changes, and then provide the changes to the project. The detailed information about providing and modifying the changes is available in the Installation Projects section of this document.

If you want to use wrapping, the packages to be deployed together with an MSI package, those are wrapped packages, are provided in the Wrapped Packages view in the same manner as when performing repackaging via wrapping. For detailed information on wrapping existing installations, refer to the Wrapping Existing Installations section of this document.

As soon as all required changes are provided or the wrapped packages are configured, depending on the project type, you can proceed with a deployment package creation. To learn more about the options available while creating deployment packages, see the Creating MSI Packages section of this document.

In general, the low-level repackaging is a set of steps to be performed for creating a deployment package that include all required changes and actions. It is really flexible and easy-to-use for those, who fully understand the steps required for performing successful repackaging. Using the available abilities it is possible to perform repackaging of any complexity, but you must always be particularly accurate while using the low-level repackaging in your everyday work.
How should I configure the repackaged installations to support an upgrade?

MSI Package Builder allows you to perform installation repackaging fast and easy. But a single instance repackaging is not enough in a real life. It is obvious, that when a new version of the repackaged software is released, you are going to create a new MSI package, containing this version. In this chapter, we will show you how to create an MSI package for repackaging a new version of any application.

The information below must be read by everyone who is going to repackage the upgraded installations and then deploy a resulting MSI package to Machines where the previous version of the package is installed. If the conditions described below are not met, the upgrade won't be performed correctly and the Programs and Features section of the Control Panel will display both versions of the repackaged installation.

For MSI packages to be deployed and upgraded correctly, they should be properly identified. For each MSI package there is a Product GUID, that uniquely identifies the product, and an Upgrade GUID that is used for matching upgrades. Windows Installer maintains its database and handles its consistency, depending on these identifiers. These identifiers are defined on a project scope while creating and an MSI package or in the Project Details view.

The Product GUID value for the different versions of the installations being repackaged should differ, and the Upgrade GUID value should match. So, for example, if we are using the GUIDs displayed on Pic 1 while repackaging the Abode Reader 10.1.0 installation, the same Upgrade GUID in combination with another Product GUID, must be used for the Abode Reader 10.3.1 installation.

If you have lost the project used to create a previous version of the MSI package, thus cannot find GUIDs used to identify that project, you can import that package. You'll find the required GUIDs in the Project Details view when the package created on a basis of the import result is selected.
It is also required that at least one of the first three version figures of the MSI package used for upgrade differs from the version of the previously generated package. For example, if we have used the 1.0.0 version of the MSI package while repackaging the *Abode Reader 10.1.0* installation, then we can use 1.1.0 version for the *Abode Reader 10.3.1*, but we cannot use 1.0.0.3.

If you are going to repack only a single installation into an MSI package, but not a suit of applications, it is recommended that the application version be used as the MSI package version. It will make the configuration process easier.

For the successful upgrade it is also required that both MSI packages should have the same installation context (Current User/All Users) defined.

Before performing a mass deployment of the upgraded package, it is strongly recommended that the upgrade process be checked for correct work on a test PC. First, install a previous version of an MSI package, and then install an upgraded version. If you have configured the packages correctly, you'll be able to see that the new version of the application is installed and there is a single entry in the *Programs and Features* section of the *Control Panel* representing this package.

Following the above-stated recommendations, you'll be able to create MSI packages that fully support upgrade, thus be able to upgrade the applications installed by the repackaged setups.
Chapter 5: Installation Projects

MSI Package Builder is a project based program – this means that all data is grouped by the projects, that are displayed in the Projects view. The interesting feature of the projects structure is that you can close the project if it is not needed right now then reopen it again. This allows you to keep the Projects view in an actual state during the everyday work.

Each project contains the information on file system, registry and other modifications – the deployment package based on the project will perform these modifications during the installation on the computer. The project stores all configuration options, required to create a deployment package between sessions. That makes it easier to manage updates. So, for example, if you have upgrade enabled in the project settings, all you need to create an updated package is to define the required modifications, change the version and provide another product GUID – everything else is already configured.

In this chapter we will take a closer look on the project structure, will show you how you can define the changes, specify actions to be performed during the package deployment, provide installations to be deployed together with the package and will guide you through the process of manual MSI package creation. This part is for those who want to fully understand the concept of MSI Package Builder and be able to use all available advanced features.

What's Inside

Projects Management
Projects Preparation
File System Modifications
Registry Modifications
Environment Variables Modifications
Services Modifications
Side-by-side Assemblies Deployment
Drivers Deployment
Printers Deployment
Using Custom Actions
Wrapping Existing Installations
Capturing Changes
Creating MSI Package
MSI Package Language
MSI Installation Context
Signing Packages
Importing Package
Projects Management

MSI Package Builder is a project based program, so a project is a main structure unit. While repackaging installations the required projects are created automatically, but it is also possible to create projects manually to be used for generating deployment packages. The projects that are not currently in use are closed, thus are not available in the Projects view. On each program start up, all projects are closed, and then you can open any project from those available to start working with it. Let us take a closer look on the projects management process, including creating, renaming, closing, opening and deleting projects.

Project Setup

The Project Setup button from the Builder group on the Home Ribbon page opens the project setup wizard, that can be used as an entry point both for managing MSI Package Builder projects and creating deployment packages on the basis of those projects.

To make it easier to create new projects (both empty and bases on an existing deployment packages), open existing projects, and create a deployment package based on an opened project we have designed a Project Setup Wizard [Pic 1]. To open this wizard, just press the Project Setup button from the Builder group on the Home Ribbon page.

Pic 1. The Project Setup Wizard
The first page of the wizard is used to introduce its abilities to you, and on the next page you are to choose the action you would like to perform. It is possible to choose between the following options: **Create New Project, Open Existing Projects, Create Package** and **Import Package**. The projects creation and opening process will be described later in this chapter. The **Create Package** option allows you to generate a deployment package using any project from those currently opened. For detailed information on the MSI package creation, refer to the **Creating MSI Package** section of this document. The **Import Package** option should be used if you would like to create a new project, containing all the changes to be performed by an existing deployment package created with MSI Package Builder. This can be useful, if you need to change the package you have generated earlier but the original project is no longer available. The detailed information on the package import is available in the **Importing Package** section of this document.

To create a new project in MSI Package Builder you can choose the **New Project** item from the **Projects** view pop-up menu or press the **New** button from the **Project Management** group on the **Home** Ribbon page. Alternatively, you can click the **New Project** link in the **Recent Projects** group on the **Welcome Screen**. The **New Project** dialog will appear on the screen. The same pane is also displayed if you choose the **Create New Project** option in the **Project Setup Wizard**.

You are proposed to specify a name for the project to be created and the project type. You should pay enough attention to a project name; it will help you to identify the project in future. As for the project type, you are proposed to choose between the regular repackaging project and the project for wrapping existing installations into a single MSI package.

The **Open** button from the **Project Management** group on the **Home** Ribbon page allows you to open projects available in the projects storage.
When you close MSI Package Builder all the projects you were working with are automatically closed too. It is also possible to close projects manually – this will be described further. If you want to continue working with those projects that were closed, you should open them again.

The links to the projects you have used recently are available in the Application Menu and in the Recent Projects group of the Welcome Screen. To open any of those projects, simply click on a link.

To open projects, choose the Open Project item from the Projects view pop-up menu or press the Open button from the Project Management group on the Home Ribbon page. Alternatively, you can click the Open Project link in the Recent Projects group on the Welcome Screen. The Open Project dialog will appear on the screen. The same pane is also displayed if you choose the Open Existing Projects option in the Project Setup wizard.

You can select both single and multiple projects in the Available Projects table to be opened at the same time. It is also possible to delete a project from the projects storage, if it is no longer needed by using the Delete Projects button from the toolbar.

The Close button from the Project Management group on the Home Ribbon page should be used to close the selected projects, thus remove them from the Projects view.

The Close All button from the Project Management group on the Home ribbon page allows you to close all currently opened projects, thus remove them from the Projects view.
MSI Package Builder provides you with an ability of always keeping the Projects view in an actual state. You can always close the projects you are not going to work with right now and they will be removed from the view, but they still exist so you can reopen any of them to continue working with them anytime you want. To close the project select it in the Projects view and choose the Close item from the pop-up menu, or press the Close button from the Project Management group on the Home ribbon page. It is also possible to close all the currently opened projects at once using the Close All button.

### Rename

The Rename button from the Project Management group on the Home Ribbon page should be used to provide a new name for the selected project.

You can face a situation when the name you have given to a project is no longer actual or you have mistakenly provided a wrong name during the project creation. It is not a problem – with MSI Package Builder you can rename the project easily any time you want. Just double click the project, or select it and press the Rename button from the Project Management group on the Home Ribbon page. The Rename Project dialog will appear on the screen, where you can provide a new name for the project.

![Pic 4. Renaming a project](image)

The important limitation for a project name is that it must be unique. Thus, it is not possible to rename the project, if the project with the same name as specified already exists (even if it is now closed).

### Delete

The Delete button from the Project Management group on the Home Ribbon page allows you to delete the selected projects from the projects storage.

The projects created in MSI Package Builder can require much space on the hard drive, but it is always possible to delete the project from the storage if it is no longer needed. To delete the project select it in the Projects view and choose the Delete item from the pop-up menu or press the Delete button from the Project Management group on the Home Ribbon page.

After you have deleted a project it is not possible to restore it any more. Make sure that the project is no longer needed before deleting it.

Now you are introduced to the project management features and should be able to organize your MSI Package Builder projects the way that will fit your needs best.
Projects Preparation

As the result of the monitoring file system, registry and other changes the MSI Package Builder project is created and filled with those changes. If the monitoring was performed via Repackage Installation wizard, all the preparation steps are performed automatically, so this chapter is interesting only to those users who are using advanced changes monitoring through the capturing abilities of MSI Package Builder.

Prepare

The Prepare button from the Links group on the Project Ribbon page should be used to prepare the selected projects.

After the changes monitoring is complete, the files created and modified during the monitoring process are not available in the projects structure. The projects containing those files are marked with a red arrow overlay (🟥), which means they are unprepared. All external files should be copied to the project structure before creating a deployment package. This is performed using the Prepare button from the Links group on the Project Ribbon page, when the project is selected.

As soon as the preparing operation is completed, you can see if it has actually succeeded by taking a look at the overlay arrow icon. If the arrow is green (🟢), than the whole projects is prepared successfully, and if it is yellow (🟡), than there were problems during the preparation process. In most cases, this means that some files were missing when the preparation was executed. The detailed information about errors is available in the Log view.

Missing Links

The Missing Links drop-down button from the Links group on the Project Ribbon page allows you to choose the method for resolving the problems with missing files in the selected projects. You can either remove those files, or try to check if they became available.

If the file was not accessible during the preparation process, thus not copied to the project storage, it is marked as missing. For the easier identification of the missing links a broken chain overlay icon (🟥) is displayed in the left top corner of the items in the File System view. The problems with missing links should be resolved before creating a deployment package. There are two possible scenarios of solving the missing links problem: repair and remove.

Before choosing a resolution mode, you should analyze the problems. If the missing files are temporary or not required to be created by the installer during a deployment of a generated package, they can be painlessly removed. Otherwise, you should provide access to those files and repair the missing links. You should use the Missing Links drop-down button from the Links group on the Project Ribbon page to resolve problems with missing links.

Roll All

The Roll All button from the System Folders group on the Project Ribbon page should be used to replace all system folders with their definitions in the selected projects.

Unroll All

The Unroll All button from the System Folders group on the Project Ribbon page allows you to expand the system folder definitions to their absolute local paths in the selected projects.
For a deployment package to be installed correctly on different operating systems with different environment configurations, it is required to replace all paths that represent the system folders (such as My Documents) with a special system folder object. This object will be expanded to the full path on the destination PC during the package deployment. You can perform this step for all folders in a project at once, using the Roll All button from the System Folders group on the Project Ribbon page.

After performing those steps, the packages are fully prepared to be used for creating a deployment package.
File System Modifications

The file system modifications are the changes to be performed by a generated package during its deployment to the file system on a target PC. Those changes can include files and folders creation or deletion, file content modification and shortcuts creation. The file system modifications are displayed in the File System view when the File System node of a project is selected in the Projects view. The changes performed during the monitoring process are added to the file system modifications automatically and can be edited only after the project is prepared. Let us take a closer look at configuring file system modifications.

**System Folder**

The System Folder button from the New group on the Project Ribbon page and on the contextual File System Ribbon page should be used to add a new system folder object to the currently configured project.

**Drive**

The Drive button from the New group on the Project Ribbon page and on the contextual File System Ribbon page should be used to add a new logical drive to the currently configured project.

Each file system modification is defined with a full path of the resource, so to add a file you should also define the whole file path. Each path is routed either with a logical drive or with a system folder object. The logical drive simply represents the drive letter, and the system folder object is used to define one of the available system folders. You can review a list of system folders in the System Folder Definition Placeholders section of this document.

To add a new logical drive definition, you should either choose the New Drive item from the pop-up menu, or click the Drive button from the New group on the regular Project and contextual File System Ribbon pages. Within the New Drive dialog, you are proposed to choose a drive letter.

![Pic 1](image1.png)

Pic 1. Adding a logical drive definition to a project
As soon as the drive is added, you can define modifications to files and folders on this drive.

Mostly the programs are installed to specific folders, and the shortcuts are created within the specific folders, that are operating system specific. To reach the same behavior you should use the system folders. To add a new system folder definition to the file system modifications, either choose the **New System Folder** menu item, or click the **System Folder** button from the **New** group on the regular **Project** and contextual **File System** Ribbon pages. Within the **New System Folder** dialog you are proposed to choose a system folder to add to the project.  

![Pic 2. Adding a system folder to a project](image)

As soon as the system folder is added, you can define modifications to files and folders in this folder. Each system folder is expanded to a full path during a generated package deployment.

<table>
<thead>
<tr>
<th><strong>Folder</strong></th>
<th>The <strong>Folder</strong> button from the <strong>New</strong> group on the contextual <strong>File System</strong> Ribbon page should be used to create a new folder modification within the selected path.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File</strong></td>
<td>The <strong>File</strong> button from the <strong>New</strong> group on the contextual <strong>File System</strong> Ribbon page should be used to create a new file modification within the selected path.</td>
</tr>
<tr>
<td><strong>Shortcut</strong></td>
<td>The <strong>Shortcut</strong> button from the <strong>New</strong> group on the contextual <strong>File System</strong> Ribbon page allows you to create a new shortcut to any target from this project in the selected path.</td>
</tr>
</tbody>
</table>
MSI Package Builder allows you to define which files and folders should be created, modified or deleted during a deployment package installation. To add a new folder modification, select the drive, folder or system folder to add modification to, and either choose the **New > Folder** item from the pop-up menu, or press the **Folder** button from the **New** group on the contextual **File System** Ribbon page. The **New Folder** dialog will appear on the screen to configure the folder modification.

![New Folder dialog](image)

Pic 3. Adding a folder modification to a project
While configuring a folder modification, you should specify a folder name and an operation to be performed over this folder during a deployment package installation. You can choose between the Create, Modify and Delete operations. For the folder creation and modification, you can attach content to this folder, using the file system import. Just check the Attach content to this folder option and provide a path to the folder to import its content. On the Security tab, you can specify the permissions to be explicitly assigned to this folder.

To add a new file modification, select the drive, folder or system folder to add modification to, and either choose the New > File item from the pop-up menu, or press the File button from the New group on the contextual File System Ribbon page. The New File dialog will appear on the screen to configure the file modification.

While configuring a file modification, you should specify a file name and an operation to be performed over this file during a deployment package installation. You can choose between Create, Modify and Delete operations. For the file creation and modification, you should define file content in the Content field, providing a path to an existing file to get content from.

While specifying files and folders modifications you can provide a relative path to the Name field – the missing path elements will be automatically created in the file system modifications.

It is also possible to override file system attributes to be used for the file and specify if this file should be treated as persistent, thus not removed during a deployment package uninstall process. These options are available on the Advanced Options tab. On the Security tab, you can specify the permissions to be explicitly assigned to this file.

The advanced options are provided for the experienced users. Please do not modify these options until you are positively sure that this modification is required for the correct package deployment.
MSI Package Builder allows you to create shortcuts to any item defined in the file system modifications during a deployment package installation. To add a shortcut definition, select the path you would like to create shortcut in and choose the **New > Shortcut** item from the pop-up menu. Alternatively, you can use the **Shortcut** button from the **New** group on the contextual **File System** ribbon page. The **New Shortcut** dialog will appear on the screen to let you configure the shortcut.

![Pic 5. Adding a shortcut to a project](image)

While adding a shortcut, you should provide a name for the shortcut to be created to the **Name** field and choose the shortcut target in the **Target** field. Optionally you can configure a set of additional shortcut settings. You can choose a working directory for a shortcut, choose a required window state, provide command-line arguments to be passed to the shortcut target and define a comment for the shortcut. The **Arguments** field supports the property definition placeholders, so you can use the standard MSI properties while specifying the arguments. See the **Property Definition Placeholders** section of this document for the list of available placeholders.

<table>
<thead>
<tr>
<th>Edit</th>
<th>The <strong>Edit</strong> button from the <strong>Management</strong> group on the contextual <strong>File System</strong> Ribbon page should be used to change the selected file system modification.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>The <strong>Delete</strong> button from the <strong>Management</strong> group on the contextual <strong>File System</strong> Ribbon page allows you to delete the selected file system modification from the project.</td>
</tr>
<tr>
<td>Copy To</td>
<td>The <strong>Copy To</strong> button from the <strong>Management</strong> group on the contextual <strong>File System</strong> Ribbon page should be used to copy the selected file system modifications to another location.</td>
</tr>
<tr>
<td>Move To</td>
<td>The <strong>Move To</strong> button from the <strong>Management</strong> group on the contextual <strong>File System</strong> Ribbon page allows you to move the selected file system modifications to another location.</td>
</tr>
</tbody>
</table>
The file system modifications can be changed, deleted and transferred to different location. To change the modification, select it in the File System view and either choose the Edit item in the pop-up menu or click the Edit button on the contextual File System Ribbon page. While editing a file system modification, it is possible to configure the same options as during its creation. To delete specific modifications from the project, select them and choose the Delete item from the pop-up menu or click the Delete button on the contextual File System Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving file system modifications within a single project and between projects. You can also use the Copy To and Move To buttons from the Management group on the contextual File System Ribbon page to transfer the selected file system modifications to a different location. While using these buttons you are proposed to select a target location in a dialog and confirm your selection.

When a container (drive, system folder or folder) is transferred to another location, all files and folders within the container are transferred as well. Also all files and folders within a container are deleted irreparably when a container is deleted, so be careful while transferring and deleting file system modifications.

**Importing Existing Files and Folders**

MSI Package Builder allows you to ease the file system modifications definition by using an ability of importing the local file system objects to the project. You can import both files and folders with or without their contents. While importing file system elements, a folder structure is created and files contents are attached automatically. To perform import, you can choose the Import > Folder or Import > File items from the pop-up menu. Alternatively, you can use the File System, Folder and File buttons from the Import groups on the regular Project and contextual File System Ribbon pages.

The File System drop-down button from the Import group on the Project Ribbon page should be used to add files and folders from the file system to the selected project.
Folder
The **Folder** button from the **Import** group on the contextual **File System** Ribbon page allows you to add a folder from the file system with or without its content to the selected project.

File
The **File** button from the **Import** group on the contextual **File System** Ribbon page allows you to add a file from the file system to the selected project.

While importing a folder, you are proposed to provide the path to the folder to be added to the selected project and choose if the folder content should also be imported. You can import files and folders from the specified folder, only files from the specified folder or ignore the folder content. As for importing a file, you should provide the path to the file to be added to the selected project. The file content will be attached automatically.

Roll
The **Roll** button from the **System Folders** group on the contextual **File System** Ribbon page should be used to replace the selected system folders with their definitions in the selected project.

Unroll
The **Unroll** button from the **System Folders** group on the contextual **File System** Ribbon page allows you to expand the selected system folders to their absolute local paths in the selected project.

Together with rolling and unrolling all system folders in a project, that is a recommended way, it is possible to roll and unroll only the specific folders. To roll a folder select it and choose the **Roll** item from the pop-up menu, and to perform unroll you should choose the **Unroll** item. Alternatively, you can use the **Roll** and **Unroll** buttons from the **System Folders** group on the contextual **File System** Ribbon page. Rolling and unrolling only specific folders is an advanced feature of MSI Package Builder and should be used very carefully.

Now you are introduced to the features used to manage the file system modifications with MSI Package Builder and should be able to define the modifications to be included into a deployment package without any misunderstanding.
Registry Modifications

The registry modifications are the changes to be performed by a generated package during its deployment to the registry on a target PC. Those changes can include keys and values creation or deletion and values modification. The registry modifications are displayed in the Registry view when the Registry node of a project is selected in the Projects view. The changes performed during the monitoring process are added to the registry modifications automatically. Let us take a close look at configuring registry modifications.

**Root Key**

The Root Key button from the New group on the Project Ribbon page and on the contextual Registry Ribbon page should be used to add a new root key to the currently configured project.

Each registry modification is defined with a full path of the resource, so to add a value you should also define the complete key path. Each path is routed with a root registry key, also known as a registry hive. To add a new root key, you should either choose the New Root Key item from the pop-up menu, or click the Root Key button from the New group on the regular Project and contextual Registry Ribbon pages. Within the New Root Key dialog, you are proposed to choose a registry hive.

![Pic 1. Adding a root key to a project](image)

As soon as the root key is added, you can define modifications to keys and values in this key.

**Key**

The Key button from the New group on the contextual Registry Ribbon page should be used to create a new registry key modification within the selected registry key.

**Value**

The Value drop-down button from the New group on the contextual Registry Ribbon page should be used to create a new registry value modification within the selected registry key.
MSI Package Builder allows you to define which registry keys and values should be created, modified or deleted during a deployment package installation. To add a new registry key modification, select the root key or key to add modification to, and either choose the New > Key item from the pop-up menu, or press the Key button from the New group on the contextual Registry Ribbon page. The New Key dialog will appear on the screen to configure the registry key modification.

While configuring a registry key modification, you should specify a key name and an operation to be performed over this key during a deployment package installation. You can choose between the Create, Modify and Delete operations.

While specifying registry modifications you can provide a path to the Name field, and the missing path elements will be created in the registry modifications automatically.

The Name field supports the property definition placeholders, so you can use the standard MSI properties while specifying the arguments. See the Property Definition Placeholders section of this document for the list of available placeholders.

It is also possible to specify if the registry key should be treated as persistent, thus not removed during a deployment package uninstall process, on the Advanced Options tab. On the Security tab, you can specify the permissions to be explicitly assigned to this key.

The advanced options are provided for the experienced users. Please do not modify these options until you are positively sure that this modification is required for the correct package deployment.
To add a new registry value modification, select the key you want to add modification to and choose the appropriate value type from the New item in the pop-up menu or from the drop-down of the Value button from the New group on the contextual Registry Ribbon page. The New Value dialog will appear on the screen [Pic 3] This dialog is almost the same for all value types; it differs only with a value data editor.

![New Value dialog]

Pic 3. Adding a registry value modification to a project

While configuring a registry value modification, you should specify a value name and an operation to be performed over this value during a package deployment. You can choose between Create, Modify and Delete operations. For the value creation and modification, you can define a data to be written to this value. For the string, expandable string and multi-string values, in case if the value already exists, you can choose if it should be overridden by the specified data or the specified data should be appended/prepended to the existing.

To define a modification to a default registry value, you can select it and choose the Edit item from the pop-up menu. If you want to edit it as a type, different from REG_SZ, you should use the Change Type item from the pop-up menu.

To define a modification to a default registry value, you can select it and choose the Edit item from the pop-up menu. If you want to edit it as a type, different from REG_SZ, you should use the Change Type item from the pop-up menu.

It is also possible to specify if the registry value should be treated as persistent, thus not removed during a deployment package uninstall process, within the Advanced Options group.

The advanced options are provided for experienced users. Please do not modify these options until you are positively sure that this modification is required for the correct package deployment.

The Value Name and Value Data fields support the property definition placeholders, so you can use the standard MSI properties while specifying the arguments. See the Property Definition Placeholders section of this document for the list of available placeholders.

<table>
<thead>
<tr>
<th>Edit</th>
<th>The Edit button from the Management group on the contextual Registry Ribbon page should be used to change the selected registry modification.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>The Delete button from the Management group on the contextual Registry Ribbon page allows you to delete the selected registry modification from the selected project.</td>
</tr>
</tbody>
</table>
The **Copy To** button from the **Management** group on the contextual **Registry** Ribbon page should be used to copy the selected registry modifications to another location.

The **Move To** button from the **Management** group on the contextual **Registry** Ribbon page allows you to move the selected registry modifications to another location.

The registry modifications can be changed, deleted and transferred to different location. To change the modification, select it in the **Registry** view and either choose the **Edit** item in the pop-up menu or click the **Edit** button on the contextual **Registry** Ribbon page. While editing a registry modification, it is possible to configure the same options as during its creation. To delete specific modifications from the project, select them and choose the **Delete** item from the pop-up menu or click the **Delete** button on the contextual **Registry** Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving registry modifications within a single project and between projects. You can also use the **Copy To** and **Move To** buttons from the **Management** group on the contextual **Registry** Ribbon page to transfer the selected registry modifications to a different location. While using these buttons you are proposed to select a target location in a dialog and confirm your selection (Pic 4).

![Copy Key To]

Pic 4. Copying registry modifications
When a registry key is transferred to another location, all keys and values within the key are transferred as well. Also all keys and values within a registry key are deleted irreparably when a registry key is deleted, so be careful while transferring and deleting registry keys.

**Importing Changes from Registration Entries (.reg) File**

MSI Package Builder allows you to ease the registry modifications definition by using an ability of importing changes from a registration entries (.reg) file. To perform import, you can choose the **Import** item from the pop-up menu. Alternatively, you can use the **Registry** button from the **Import** groups on the regular **Project** and contextual **Registry** Ribbon pages.

While importing changes, you are proposed to provide the path to the file to use as a definition for the changes to be included into a project.

Now you are introduced to the features used to manage the registry modifications with MSI Package Builder and should be able to define the modifications to be included into a deployment package without any misunderstanding.
Environment Variables Modifications

With MSI Package Builder you can easily add to a project any system and user environment variables so that they will be registered in or unregistered from the operating system when the deployment package is installed on the computer.

The environment variables modifications are displayed in the Environment Variables view when the Environment Variables node of a project is selected in the Projects view. Let us take a close look at configuring environment variables modifications.

**Environment Variable**
The Environment Variable button from the New group on the regular Project and contextual Environment Variables Ribbon pages should be used to add a new environment variable modification to the currently configured project.

MSI Package Builder allows you to add both modifications to system and user environment variables. To add a modification choose the New Environment Variable item from the pop-up menu or click the Environment Variable button from the New group on regular Project and contextual Environment Variables Ribbon pages. The New Environment Variable dialog will appear on the screen [Pic 1].

![Pic 1. Configuring an environment variable modification](image-url)
During the configuration process, you are proposed to specify the environment variable name and the operation to be performed over this variable during a deployment package installation. You can choose between the Create, Create if does not exist, Remove and Remove if value equals operations. For creating an environment variable you should provide a variable value to the Value field. If variable already exists during a generated package installation, you can choose if its value should be overridden with the defined one, or the defined value should be appended/prepended to the existing one. The System Variable checkbox is used to identify if you are going to modify system or user environment variable.

It is also possible to specify if the environment variable should be treated as persistent, thus not removed during a deployment package uninstall process, within the Advanced Options group.

---

**The advanced options are provided for the experienced users. Please do not modify these options until you are positively sure that this modification is required for the correct package deployment.**

---

The Value field supports the property definition placeholders, so you can use the standard MSI properties while specifying the value for an environment variable. See the Property Definition Placeholders section of this document for the list of available placeholders.

---

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Edit" /> <strong>Edit</strong></td>
<td>The Edit button from the Management group on the contextual Environment Variables Ribbon page should be used to change the selected environment variables modification.</td>
</tr>
<tr>
<td><img src="image" alt="Delete" /> <strong>Delete</strong></td>
<td>The Delete button from the Management group on the contextual Environment Variables Ribbon page allows you to delete the selected environment variables modifications from the selected project.</td>
</tr>
<tr>
<td><img src="image" alt="Copy To" /> <strong>Copy To</strong></td>
<td>The Copy To button from the Management group on the contextual Environment Variables Ribbon page should be used to copy the selected environment variables modifications to another location.</td>
</tr>
<tr>
<td><img src="image" alt="Move To" /> <strong>Move To</strong></td>
<td>The Move To button from the Management group on the contextual Environment Variables Ribbon page allows you to move the selected environment variables modifications to another location.</td>
</tr>
</tbody>
</table>
The environment variables modifications can be changed, deleted and transferred to different location. To change the modification, select it in the **Environment Variables** view and double-click it. Alternatively, you can either choose the **Edit** item in the pop-up menu or click the **Edit** button on the contextual **Environment Variables** Ribbon page. While editing an environment variables modification, it is possible to configure the same options as during its creation. To delete specific modifications from the project, select them and choose the **Delete** item from the pop-up menu or click the **Delete** button on the contextual **Environment Variables** Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving environment variables modifications between projects. You can also use the **Copy To** and **Move To** buttons from the **Management** group on the contextual **Environment Variables** Ribbon page to transfer the selected environment variables modifications to a different location. While using these buttons you are proposed to select a target location in a dialog and confirm your selection ([Pic 2].

![Pic 2. Copying environment variables modifications](image)

Now you are introduced to the features used to manage the environment variables modifications with MSI Package Builder and should be able to define the user and system variables modifications to be included into a deployment package without any misunderstanding.
Services Modifications

The services modifications are the changes to be performed by a generated package during its deployment to the installed services on a target PC. Those changes can include services creation, deletion and control.

The services modifications are displayed in the Services view when the Services node of a project is selected in the Projects view. The changes performed during the monitoring process are added to the services modifications automatically. Let us take a close look at configuring service modifications.

Service

The Service button from the New group on the regular Project and contextual Services Ribbon pages should be used to add a new service modification to the currently configured project.

To add a new service modification, you should either choose the New Service item from the pop-up menu, or press the Service button from the New group on the regular Project and contextual Services Ribbon pages. Within the New Service dialog, you are proposed to configure the service modification in detail [Pic 1].

Pic 1. Adding a service to be created and started during a deployment package installation
While adding a service modification, you should firstly specify the service name and the operation to be performed over the service. You can choose between the **Create**, **Delete**, **Control** and **Restart** operations. The scope of the settings available for configuring depends on the selected operation. Let us describe each operation in detail.

The **Create** operation is used if the service is a part of the installation. So, it is created together with the generated deployment package installation. For the **Create** operation, the scope of options is the widest. Together with the service name, used to install the service, you must also provide a display name used for the service and a path to the file from the current project to be used as the service executable on the **General** tab. It is also recommended that a service description be provided, which is displayed in a services section of the computer management console for each installed service. The other options available for service creation on the **General** tab are the execution options; those are the service type, startup type and the arguments to be passed to the service on startup.

On the **Log On** tab for the service being created, you can choose it should log on as a local system, or as a specific user. For the local system account, it is possible to define if the service should interact with desktop. On this tab, you can also configure an error control level used for this service. You can choose from the following:

- **Normal** - the startup program logs the error in the event log but continues the startup operation.
- **Critical** - the startup program logs the error in the event log, if possible. If the last-known-good configuration is being started, the startup operation fails. Otherwise, the system is restarted with the last-known good configuration.
- **Ignore** - the startup program ignores the error and continues the startup operation.

The **Dependencies** tab is also used when configuring the service creation and allows defining the services this service depends on. The service name is provided to the **Dependency** field and you can add, edit and remove dependencies on-line using the corresponding buttons built into this field.

On the **Security** tab, you can specify the permissions to be explicitly assigned to this service.

While specifying a service name, display name, description, startup arguments and dependencies you can use the property definition placeholders, standing for the standard MSI properties. See the **Property Definition Placeholders** section of this document for the list of available placeholders.

For the other operations, except **Create**, the **Log On**, **Dependencies** and **Security** tabs are not used and only the **Name** field from the top part of the **General** tab is used. As for the **Delete** operation, the service name is the only field to be specified. The service with the matching name will be deleted from a target PC during a generated deployment package installation. For the **Control** operation, after providing a service name, you should define triggers to start and stop the service on. You can choose between **Install**, **Uninstall**, **Both** and **None** options to start and stop the service on. As for the **Restart** operation, you should choose if it should be restarted on install, uninstall or both events.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edit</strong></td>
<td>The <strong>Edit</strong> button from the <strong>Management</strong> group on the contextual <strong>Services</strong> Ribbon page should be used to change the selected services modification.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>The <strong>Delete</strong> button from the <strong>Management</strong> group on the contextual <strong>Services</strong> Ribbon page allows you to delete the selected services modifications from the selected project.</td>
</tr>
</tbody>
</table>
The **Copy To** button from the Management group on the contextual Services Ribbon page should be used to copy the selected services modifications to another location.

The **Move To** button from the Management group on the contextual Services Ribbon page allows you to move the selected services modifications to another location.

The services modifications can be changed, deleted and transferred to different location. To change the modification, select it in the Services view and double-click it. Alternatively, you can either choose the Edit item in the pop-up menu or click the Edit button on the contextual Services Ribbon page. While editing a service modification, it is possible to configure the same options as during its creation. To delete specific modifications from the project, select them and choose the Delete item from the pop-up menu or click the Delete button on the contextual Services Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving services modifications between projects. You can also use the Copy To and Move To buttons from the Management group on the contextual Services Ribbon page to transfer the selected services modifications to a different location. While using these buttons you are proposed to select a target location in a dialog and confirm your selection.

Now you are introduced to the features used to manage the services modifications with MSI Package Builder and should be able to create and control services by a generated deployment package without any misunderstanding.
Side-by-side Assemblies Deployment

MSI Package Builder provides you with an ability to deploy shared libraries. A shared library is a file that is intended to be shared by executable files and further shared objects files. Modules used by a program are loaded from individual shared objects into memory at load time or run time. Side-by-side assemblies can safely share assemblies among multiple applications and can offset the negative effects of assembly sharing. Such effects, known as 'DLL hell', include version conflicts, missing DLLs, duplicate DLLs, and incorrect or missing registration. Instead of a single version of an assembly that assumes backward compatibility with all applications, side-by-side assembly sharing enables multiple versions of an assembly to run simultaneously on the system. As for .NET assemblies, the side-by-side concept is implemented with a help of the Global Assembly Cache. In case of Win32 assemblies, in side-by-side, Windows stores multiple versions of a DLL in the WinSxS folder of the Windows directory. SxS is also the technological basis for registration-free COM activation.

For the side-by-side assemblies to be deployed, they are specified on a project level and displayed in the Assemblies view when the Assemblies node of a project is selected in the Projects view. The assemblies installed during the monitoring process are added to the project automatically and can be edited only after the projects containing those changes are prepared. Let us take a closer look at configuring side-by-side assemblies.

<table>
<thead>
<tr>
<th>Win32 Assembly</th>
<th>.NET Assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Win32 Assembly button from the New group on the Project Ribbon page and on the contextual Assemblies Ribbon page should be used to add a new Win32 assembly to the currently configured project.</td>
<td>The .NET Assembly button from the New group on the Project Ribbon page and on the contextual Assemblies Ribbon page should be used to add a new .NET assembly to the currently configured project.</td>
</tr>
</tbody>
</table>
To add a new assembly to a project, you can choose the **New Win32 Assembly** and **New .NET Assembly** items from the **Assemblies** view pop-up menu. Alternatively, you can use the **Win32 Assembly** and **.NET Assembly** button from the **New** group on the **Project** and **Assemblies** Ribbon pages. The dialog will appear on the screen to let you configure the assembly to be added.

![Pic 1. Configuring a Win32 Assembly](image)

While configuring a Win32 assembly you should provide a path to the assembly manifest and security catalog files. If it is required, you should also add assembly files to the **Assembly Files** table. As for a .NET assembly, it is required that a path to the assembly file be provided. Optionally, it is possible to add configuration files, required for the assembly installation, to the **Assembly Files** table.

| **Edit** | The **Edit** button from the **Management** group on the contextual **Assemblies** Ribbon page should be used to change the selected assembly. |
| **Delete** | The **Delete** button from the **Management** group on the contextual **Assemblies** Ribbon page allows you to delete the selected assemblies from the selected project. |
| **Copy To** | The **Copy To** button from the **Management** group on the contextual **Assemblies** Ribbon page should be used to copy the selected assemblies to another project. |
| **Move To** | The **Move To** button from the **Management** group on the contextual **Assemblies** Ribbon page allows you to move the selected assemblies to another project. |
The side-by-side assemblies can be changed, deleted and transferred between projects. To change the assembly, select it and either double-click, or choose the Edit item in pop-up menu or on the contextual Assemblies Ribbon page. While editing an assembly, you can define the same scope of properties as during its creation. To delete the assembly from the project, select it and choose the Delete item in the pop-up menu or on the contextual Assemblies Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving assemblies between projects. You can also use the Copy To and Move To buttons from the Management group on the contextual Assemblies Ribbon page to transfer the selected assemblies to a different project. While using these buttons you are proposed to select a project from those available in a dialog and confirm your selection.

Now you are introduced to the side-by-side assemblies concept and an ability to deploy such assemblies with MSI Package Builder, thus you should be able to define the assemblies to be installed by a deployment package without any misunderstanding.

Drivers Deployment

MSI Package Builder is capable of deploying hardware and software drivers. It is possible to create or delete basic drivers and install or pre-install driver packages in scope of an MSI package installation.

Both basic drivers and drivers from packages can be registered as device class filters by MSI Package Builder, if required. Changes to the operating system drivers are displayed in the Drivers view when the Drivers node of a project is selected in the Projects view. Changes performed during the monitoring process are added automatically to the drivers deployment actions. Let us take a closer look at configuring the drivers deployment actions.

Basic Driver

The Basic Driver button from the New group on the regular Project and contextual Drivers Ribbon pages should be used to add a new basic driver deployment action to the currently configured project.
Driver Package
The **Driver Package** button from the **New** group on the regular **Project** and contextual **Drivers** Ribbon pages should be used to add a new driver package to the currently configured project to be either installed or pre-installed by the resulting MSI package.

DIFx Driver Package
The **DIFx Driver Package** button from the **New** group on the regular **Project** and contextual **Drivers** Ribbon pages should be used to add a new **Driver Installation Framework** package to the currently configured project to be either installed or pre-installed by a resulting deployment package.

The basic driver is a simple driver that is shipped as a single dynamic link library (usually as the *.sys file) and is managed like a windows service. A driver package consists of all the software components that you must supply for your device to be supported under Windows. The following components are necessary to install and support a device on a Windows operating system: the driver file (*.sys), the installation files (*.inf and *.cat) and other files (the device installation application, the device icon, the device property pages, etc).

To add a new basic driver deployment action, you should either choose the **New Basic Driver** item from the pop-up menu or press the **Basic Driver** button from the **New** group on the regular **Project** and contextual **Drivers** Ribbon pages. Within the **New Basic Driver** dialog, you are proposed to configure the basic driver deployment action in detail (Pic 1).

![Pic 1. Adding a basic driver to be created and started during an MSI package installation](image-url)
While adding a basic driver deployment action, you should first specify the driver name and the operation to be performed on the driver. You can choose between the Create and Delete operations. The scope of settings available for configuring depends on the selected operation. Let us describe each operation in detail.

The Create operation is used if the driver is a part of an installation. Thus, it is created together with the generated MSI package installation. Along with the driver name used to install the driver, you should also provide the display name used for the driver and the path to the file from the current project to be used as the driver binary on the General tab. You can also provide the driver security catalog file, if required. It is also recommended that the driver description be provided, which is displayed in the computer management console for each driver. Other options displayed on the General tab are the advanced driver loading settings, which are the driver type, the startup type, the load order group and the error control.

The Driver Type field is used to choose the type of the basic driver to be installed: either a kernel driver or a system driver.

As for the Startup Type, the following values can be used:

- **Boot** - the device driver required to start the computer;
- **System** - the non-boot-start driver that detects device(s) that are not PnP-enumerable;
- **Auto** - the Non-PnP driver that must be started by the service control manager;
- **On Demand** - the PnP driver;
- **Disabled** - the disabled driver.

The Load Order Group is used to define the group this driver should be added to for loading by the operating system.

You can choose from the following Error Control levels:

- **Normal** - the startup program logs error in the event log but continues the startup operation.
- **Critical** - the startup program logs error in the event log, if possible. If the last-known-good configuration is being started, the startup operation fails. Otherwise, the system is restarted with the last-known good configuration.
- **Ignore** - the startup program ignores error and continues the startup operation.
The **Dependencies** tab is also used while configuring the driver creation. It allows defining the drivers this driver depends on. The driver name is provided to the **Dependency** field, and you can add, edit and remove dependencies on-line using the corresponding buttons built into this field.

On the **Security** tab, you can specify the permissions to be explicitly assigned to this driver.

While specifying the driver name, display name, description and dependencies you can use the property definition placeholders standing for the standard MSI properties. See the [Property Definition Placeholders](#) section of this document for the list of available placeholders.

For the **Delete** operation, the **Dependencies** tab is not used and only the **Name** field from the **General** tab is applicable. The driver with the matching name will be deleted from the target PC while installing the generated MSI package.

If you would like to register the basic driver as a filter for one or more device classes, you should enable the **Register this driver as a device class filter** option on the **Install Options** tab. The **Install Options** tab contains a list of device class filters the configured driver is used as. You can add and remove the filter configuration entries using the corresponding buttons on the toolbar. Let us take a closer look at the device class filter configuration **Pic 2**.

![Pic 2. Registering a driver as an upper-level device class filter](image)
When registering a driver as a device class filter, you should provide the device class to register the driver to into the **Device Class** field. Within the **Filter Type** drop-down, you should choose the type of the filter to be registered. A **Lower-Level** filter driver monitors and/or modifies I/O requests to a particular device. Typically, such filters redefine the hardware behavior to match the expected specifications. For example, a lower-level class filter driver for mouse devices could provide acceleration performing a nonlinear conversion of the mouse movement data. An **Upper-Level** filter driver adds value to a particular device. For example, an upper-level device filter driver for a keyboard could enforce additional security checks. The **Dependency** and **Order** fields are responsible for the filter driver position in the scope of all filters for the specified device class.

To add a new driver package deployment action, you should either choose the **New Driver Package** item from the pop-up menu or press the **Driver Package** button from the **New** group on the regular **Project** and contextual **Drivers** Ribbon pages. If you are going to deploy the package that requires deployment via the DIFx (Driver Installation Framework), use the **New DIFx Driver Package** menu item and the **DIFx Driver Package** button respectively. The file system browser will be opened to let you choose the driver information file for the package. As soon as it is chosen, you will see the dialog where you are proposed to configure the driver package deployment action in detail. Let us use the **New Driver Package** one as the example ![Pic 3](image).

**Pic 3.** Adding a driver package to be pre-installed during an MSI package installation
Common information on the driver package being configured is displayed on the General tab. As you can see, the Driver Information and Security Catalog fields, as well as the Driver Package Files table, are filled automatically with the files provided by the selected installation file. You can change any of these aspects in the future, if required.

The next step is providing the driver name and configuring the deployment. The name is provided to the Name field. The next step is to specify the target platform in the Platform field. The platform can be specified only during the driver package creation. As for the deployment options, you can choose if the driver should be installed or pre-installed within the Operation drop-down. The Install operation should be chosen if the device with an appropriate ID is already present in the system or it is a virtual device created by the installer. The Pre-Install operation allows you to install a package to the driver storage. The operating system will use the installed driver as soon as an appropriate device is installed to the system.

The content of the Install Options tab differs depending on the type of the driver package. It may be possible to choose the devices or sections from the package to be installed, the hardware component ID and/or register a driver from the configured package as a filter for one or more device classes.

| Move Up | Move Up button from the Order group on the contextual Drivers ribbon page should be used to move the selected drivers up the deployment order. |
| Move Down | Move Down button from the Order group on the contextual Drivers ribbon page should be used to move the selected drivers down the deployment order. |

By default, the drivers deployment order is the same as the addition order, but you can reorder the drivers using the Move Up and Move Down items from the pop-up menu or the Order group on the contextual Drivers Ribbon page.

| Edit | Edit button from the Management group on the contextual Drivers Ribbon page should be used to change the selected driver deployment action. |
| Delete | Delete button from the Management group on the contextual Drivers Ribbon page allows you to delete the selected driver deployment actions from the selected project. |
| Copy To | Copy To button from the Management group on the contextual Drivers Ribbon page should be used to copy the selected driver deployment actions to another location. |
| Move To | Move To button from the Management group on the contextual Drivers Ribbon page allows you to move the selected driver deployment actions to another location. |
Driver deployment actions can be changed, deleted or transferred another location. To change the driver deployment action, select it in the **Drivers** view and double-click it. Alternatively, you can either choose the **Edit** item in the pop-up menu or click the **Edit** button on the contextual **Drivers** Ribbon page. While editing a driver deployment action, it is possible to configure the same options as those available during the action creation. To delete specific actions from a project, select them and choose the **Delete** item from the pop-up menu or click the **Delete** button on the contextual **Drivers** Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving driver deployment actions between projects. You can also use the **Copy To** and **Move To** buttons from the **Management** group on the contextual **Drivers** Ribbon page to transfer the selected driver deployment actions to another location. While using these buttons you are proposed to select the target location in the dialog and confirm your selection Pic 4.

Now that you have been introduced to the features used to manage driver deployment actions with MSI Package Builder, you should be able to deploy drivers by the generated MSI package with full understanding.

![Pic 4. Copying driver deployment actions](image-url)
**Printers Deployment**

MSI Package Builder comes with an option of performing changes to the printing system configuration. It is possible to create, delete and modify printers; install and uninstall printer drivers; create and delete print processors and monitors; create, modify and delete printer ports, both local and TCP/IP. All those actions will be performed within the scope of the generated MSI package installation if defined in the project.

The set of changes to the printing system described in the project is displayed in the **Printers** view when the **Printers** node is selected in the **Projects** view. The changes performed during the monitoring process are automatically added to the printing system configuration actions. Let us take a closer look at the possible types of actions.

| **Printer Entries** | The **Printer Entries** drop-down button from the **New** group on the **Project** Ribbon page can be used to create modifications to the printing system to be performed by the resulting MSI package.
| **Printer** | The **Printer** button from the **New** group on the **Printers** contextual Ribbon page should be used to add a new deployment action for creating, modifying or deleting a printer during the package deployment.
| **Printer Driver** | The **Printer Driver** button from the **New** group on the **Printers** contextual Ribbon page should be used to add a new deployment action for installing or removing a printer driver during the package deployment.
| **Print Processor** | The **Print Processor** button from the **New** group on the **Printers** contextual Ribbon page should be used to add a new deployment action for adding or removing a print processor during the package deployment.
| **Print Monitor** | The **Print Monitor** button from the **New** group on the **Printers** contextual Ribbon page should be used to add a new deployment action for adding or removing a print monitor (either a language monitor or a port monitor) during the package deployment.
| **Printer Port** | The **Printer Port** button from the **New** group on the **Printers** contextual Ribbon page should be used to add a new deployment action for creating, modifying or deleting a printer port during the package deployment.
Let us start with adding a printer deployment action. As it has already been mentioned, it is possible to create, modify and delete printers during a package deployment. To add a printer deployment action, you should use the **Printer** item either from the **Printers** view pop-up menu or from the **New** group on the **Printers** contextual Ribbon page. Alternatively you can use the **Printer Entries** drop-down button located on the **Project** Ribbon page in the **New** group. The **New Printer** dialog will appear on the screen to let you configure the required deployment action. 

![Pic 1](image.png)

**Pic 1.** Adding a printer to be created during installation of the resulting package
The common printer deployment action configuration is represented on the General tab of the dialog. If you would like the resulting MSI package to delete one of the existing printers, you should choose the Delete value in the Operation drop-down list and provide the name of the printer to be deleted to the Name field. For creation and modification of printers, there is a set of other fields to be specified beside the name. Along with the name, it is required to choose a driver to be used by the printer device and a port for the printer to reside on in the Driver and Port fields respectively. Two more required fields are Processor and Data Type. The Processor field should contain the processor to be used by the printer. The processor modifies the job as required for it to be printed properly. We will take a closer look at print processors later in this section of the documentation. The processor parameters can be provided to the Parameters field. The bitwise attributes mask can be specified by means of an easy-to-use editor using the ellipses button of the Attributes field. The current and default priorities of printers are specified in the Priority and Default Priority fields respectively. The page separator file is chosen from those already available in the project using the ellipses button of the Separator field. You can also specify the printer location, the share name, the custom spooler directory and a comment for the printer in the corresponding fields of this dialog.

On the Registry tab of the configuration dialog it is possible to configure the changes applied to printer-specific sections of Windows registry while creating or modifying the printer during the MSI package deployment process.

If you want to make a specific printer as default after installing it, you can select it in the modifications list and choose the Set as Default Printer item from the Printers view pop-up menu.

The next printer system layer available for configuration is the printer drivers. Printer drivers contain information specific to the printer begin used. Printer drivers reside on user's computers and are used by the GDI to render print jobs. A printer driver is a software program that understands how to communicate with printers and plotters. Printer drivers translate information a user sends from his/her computer into commands that a printer understands. A printer driver sends printer-setting data, including the specifications needed to produce each character of the document, to the GDI. It also transmits helper services or utilities required to make the output print correctly. Each printer driver consists of the driver binary, data type and configuration files, an optional help file and dependent files. MSI Package Builder allows you to install and/or uninstall printer drivers during the resulting package deployment.

To add a deployment action to be performed for a printer driver, you should use the Printer Driver item either from the Printers view pop-up menu or from the New group on the Printers contextual Ribbon page. Alternatively you can use the Printer Entries drop-down button located on the Project Ribbon page in the New group. The New Printer Driver dialog will appear on the screen to let you configure the required deployment action.
Pic 2. Adding a printer driver to be installed with the resulting package
The **Name** and **Environment** fields are used to provide a driver name and specify the target environment. These fields must be filled for both **Create** and **Delete** operations. As for the **Create** operation, you should also define the print monitor for the driver, the driver version, the default data type and driver files as described above.

MSI Package Builder allows you to manage print processors when deploying the resulting package. Print processors are user-mode DLLs that are responsible for converting a print job's spooled data into a format that can be sent to a print monitor. They are also responsible for handling application requests to pause, resume and cancel print jobs. You can add and delete print processors during a package deployment.

To add a print processor to be managed during a package deployment, you should use the **Print Processor** item either from the **Printers** view pop-up menu or from the **New** group on the **Printers** contextual Ribbon page. Alternatively you can use the **Printer Entries** drop-down button located on the **Project** Ribbon page in the **New** group. The **New Print Processor** dialog will appear on the screen to let you configure the required deployment action [Pic 3].
Within the dialog, you can define a name for the print processor to be managed and the target environment. If you are going to create a processor, it is also required to provide a path to the processing library in the **Processor Binary** field.

With MSI Package Builder, it is also possible to manage print monitors. Print monitors are responsible for directing a print data stream from the print spooler to an appropriate port driver. Two types of print monitors are defined: language monitors and port monitors. Both can be created or deleted with a package generated with MSI Package Builder.

To add a print monitor entry, you should use the **Print Monitor** item either from the **Printers** view pop-up menu or from the **New** group on the **Printers** contextual Ribbon page. Alternatively you can use the **Printer Entries** drop-down button located on the **Project** Ribbon page in the **New** group. The **New Print Monitor** dialog will appear on the screen to let you configure the required deployment action [Pic 4].

![Pic 4. Adding a print monitor](image)
When adding a print monitor, you should define its name, the environment to be affected and the operation to perform over this monitor. For the monitors being installed, you should also choose the monitor binary from the file available in the configured project.

The last layer available for management with a package generated via MSI Package Builder is the printer ports. There are two types of ports, namely local and TCP/IP. Packages generated with MSI Package Builder can create and delete both local and TCP/IP printer ports as well as modify TCP/IP ports.

To add a printer port operation, you should use the **Printer Port** item either from the **Printers** view pop-up menu or from the **New** group on the **Printers** contextual Ribbon page. Alternatively you can use the **Printer Entries** drop-down button located on the **Project** Ribbon page in the **New** group. A dialog will be displayed for choosing the port type, and as soon as it is chosen, you will get to either the **New Local Port** dialog [Pic 5] or the **New TCP/IP Port** dialog [Pic 6].

When adding a modification to local ports, you should simply provide the local port name to the **Name** field and specify the operation to be performed with this port (either **Create** or **Delete**) in the **Operation** drop-down.
The TCP/IP ports configuration process is a bit more complex. You can choose if you would like to create, modify or delete the port being added using the Operation drop-down. For deleting a port, it is only required to provide its name. For both adding and modifying a port, you should also provide the port properties. Beside the name, the port is identified with its version, the MIB index and the host address. The values should be provided to the Version, MIB Index and Host Address fields respectively. If the port is meant for raw bytes communication, then the Raw value should be selected in the Protocol drop-down list, and the port number should be specified in the Port Number field. In case the LPR protocol is used, the corresponding value should be selected in the Protocol drop-down, and the LPR queue name should be specified in the Queue Name field. In case the LPR byte counting is enabled for communication, the corresponding check box should be ticked. If the printer device residing on the port can report the SNMP status, the SNMP Status Enabled check should be ticked, and the required values should be provided to the Community Name and SNMP Device Index fields.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>The Edit button from the Management group on the Printers contextual Ribbon page should be used to change the selected printing system configuration action.</td>
</tr>
<tr>
<td>Delete</td>
<td>The Delete button from the Management group on the Printers contextual Ribbon page allows you to delete the selected printing system configuration actions from the selected project.</td>
</tr>
<tr>
<td>Copy To</td>
<td>The Copy To button from the Management group on the Printers contextual Ribbon page should be used to copy the selected printing system configuration actions to another location.</td>
</tr>
<tr>
<td>Move To</td>
<td>The Move To button from the Management group on the Printers contextual Ribbon page allows you to move the selected printing system configuration actions to another location.</td>
</tr>
</tbody>
</table>
The printing system configuration actions can be changed, deleted and transferred to different location. To change the printing system configuration action, select it in the **Printers** view and double-click it. Alternatively you can either choose the **Edit** item in the pop-up menu or click the **Edit** button on the **Printers** contextual Ribbon page. While editing a printing system configuration action, it is possible to configure the same options as during its creation. To delete specific actions from the project, select them and choose the **Delete** item from the pop-up menu or click the **Delete** button on the **Printers** contextual Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving printing system management actions between projects. Also you can use the **Copy To** and **Move To** buttons from the **Management** group on the **Printers** contextual Ribbon page to transfer the selected printing system configuration actions to a different location. While using these buttons you are proposed to select a target location in a dialog and confirm your selection ![Pic 7](image).

![Pic 7. Copying printing system configuration actions](image)

Now that you have been introduced to the features used for managing the printing system configuration actions with MSI Package Builder, you should be able to set up a printing system with the help of a generated MSI package without any difficulties.
Using Custom Actions

The custom actions are the actions that can be performed together with the MSI package install and/or uninstall process. They can be used, for example, to prepare the system for the installations, to check prerequisites, to start the application on installation completion, etc. These actions are defined on a project level and are displayed in the Custom Actions view when the Custom Actions node of a project is selected in the Projects view.

There are two kinds of custom actions, namely pre & post actions and system actions. The pre & post actions are generic commands to be performed, and the system actions are predefined commands.

Custom Actions
The Custom Actions drop-down button from the New group on the Project Ribbon page allows creating a new action of a specific type and add it to the selected project.

Pre & Post Action
The Pre & Post Action button from the New group on the Custom Actions Ribbon page should be used to create a new generic action and add it to the selected project.

To create a new action, choose the New Pre & Post Action item in the Custom Actions view pop-up menu from the Pre & Post Actions part. Alternatively, you can use the Pre & Post Action option within the Custom Action button from the New group on the Project Ribbon page and the Pre & Post Action button on the contextual Custom Actions Ribbon page. In any case the New Pre & Post Action dialog will appear on the screen to let you configure the action to be created.

Pic 1. Creating a new pre & post action
Each pre & post action can be represented with a command (such as an executable or script file) or a simple action to open a file or an URL. The actual action to be executed should be provided to the **Command** field. You can input the action command manually and choose a file to be opened either from the project, or from the local file system. Use the **Select File** and **Select Project File** buttons build into the edit box to select required files. The parameters to be passed to the command are to be defined in the **Parameters** field. Both the **Command** and **Parameters** fields support the property definition placeholders, so you can use the standard MSI properties while configuring the actions. See the **Property Definition Placeholders** section of this document for the list of available placeholders.

The Windows Installer does not allow installing, configuring and uninstalling packages in parallel, though it is impossible to execute installation, repair or uninstallation of another MSI package within the custom actions execution scope.

As for the start type, you can choose between **Before Install**, **After Uninstall**, **Before Uninstall** and **After Uninstall**. Thus, you can, for example check some prerequisites and perform preparation steps before the package installation and perform a kind of clean-up after uninstalling the package. For each action, you can choose the execution context, if to run it as administrator or as invoker in case of current user account context, if the installer should wait for the action to complete and if it should analyze the completion result.

The **Administrator** value from the **Run As** drop-down should be selected only for executing the actions that required administrative privileges to function. It is insecure to run all actions as administrator.

If the successful action completion is required for the installation to complete successfully, than you can check both the **Wait until this action is completed** and **Interrupt the install/uninstall process if this action is completed with an error** options. However, be aware of the fact, that if the action is implemented incorrectly, the whole installation will fail.

By default, the pre & post actions execution order is the same as the addition order, but you can reorder the actions using the **Move Up** and **Move Down** items from the pop-up menu or the **Order** group on the contextual **Custom Actions** Ribbon page.

**SAM Licenses Installation**

The **SAM Licenses Installation** button from the **New** group on the **Custom Actions** contextual Ribbon page should be used to create a new action for installing Software Assets Management (SAM) licenses to the Software Licensing Service (SLS) and add it to the selected project.
To create a new action for installing Software Assets Management (SAM) licenses, choose the **New SAM Licenses Installation** item in the **Custom Actions** view pop-up menu from the **System Actions** part. Alternatively, you can use the **SAM Licenses Installation** option within the **Custom Action** button from the **New** group on the **Project** Ribbon page and the **SAM Licenses Installation** button on the **Custom Actions** contextual Ribbon page. In each case, the **New SAM Licenses Installation** dialog will appear on the screen to let you configure the action to be created [Pic 2].

![Pic 2. Creating a new SAM Licenses Installation action](image)

For each SAM licenses installation action, you can define a name and a set of licenses to be installed to the Software Licensing Service (SLS) during the installation of the generated MSI package.

### Pin Application Action

The **Pin Application Action** button from the **New** group on the **Custom Actions** contextual Ribbon page allows you to create a new action to pin any application to or unpin it from the Start Menu and/or the Task Bar.
If you would like to pin any application to or unpin it from the Start Menu and/or the Task Bar, you can use the Pin Application action. To create such an action, choose the **New Pin Application Action** in the **Custom Actions** view pop-up menu from the **System Actions** part. Alternatively, you can use the **Pin Application Action** option within the **Custom Action** button from the **New** group on the **Project** Ribbon page and the **Pin Application Action** button on the **Custom Actions** contextual Ribbon page. The **New Pin Application Action** dialog will appear on the screen in any case (Pic 3).

![New Pin Application Action dialog](image)

For each action, you should define its name, the application to operate, whether the application should be pinned or unpinned, and if the operation should be performed for the Start Menu, the Task Bar or both.

**Edit**
- The **Edit** button from the **Management** group on the contextual **Custom Actions** Ribbon page should be used to change the selected action configuration.

**Delete**
- The **Delete** button from the **Management** group on the contextual **Custom Actions** Ribbon page allows you to delete the selected actions from the selected project.

**Copy To**
- The **Copy To** button from the **Management** group on the contextual **Custom Actions** Ribbon page should be used to copy the selected actions to another project.

**Move To**
- The **Move To** button from the **Management** group on the contextual **Custom Actions** Ribbon page allows you to move the selected actions to another project.
The custom actions can be changed, deleted and transferred between projects. To change the action, select it and either double-click, or choose the *Edit* item in pop-up menu or on the contextual **Custom Actions** contextual Ribbon page. While editing an action, you can define the same scope of properties as during its creation. To delete the action from the project, select it and choose the **Delete** item in the pop-up menu or on the contextual **Custom Actions** contextual Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving custom actions between projects. You can also use the **Copy To** and **Move To** buttons from the **Management** group on the contextual **Custom Actions** contextual Ribbon page to transfer the selected actions to a different project. While using these buttons you are proposed to select a project from those available in a dialog and confirm your selection.

Now you are introduced to the custom actions concept and the abilities the custom actions provide you with, thus you should be able to use this feature of MSI Package Builder to resolve appropriate tasks.

**Wrapping Existing Installations**

One of the methods of installations repackaging supported by MSI Package Builder is wrapping existing installations into a generated MSI package. Wrapping means that those installations are distributed as a part of the MSI package and are installed, uninstalled and repaired simultaneously. If you want to combine monitored or manually defined changes with the wrapped packages, you must take into account the wrapped packages install, uninstall and repair process executes after processing the defined changes. The wrapped packages are defined on a project level and displayed in the **Wrapped Packages** view when the **Wrapped Packages** node of a project is selected in the **Projects** view.

The **Wrapped Package** button from the **New** group on the **Project** Ribbon page and on the contextual **Wrapped Packages** Ribbon page should be used to create a new wrapped package and add it to the selected project.
If the project was created in a scope of the **Repackage Installation** wizard, the packages defined during repackaging are automatically added to the project. You can also add any number of wrapped packages to any project manually. To add a new wrapped package, choose the **New Wrapped Package** item from the **Wrapped Package** view pop-up menu. Alternatively, you can choose the **Wrapped Package** button from the **New** group on the **Project** Ribbon page and on the contextual **Wrapped Packages** Ribbon page. The browse dialog will be displayed to let you choose the installation package to be wrapped. As soon as the installation is chosen, you are proposed to configure the package [Pic 1].

![Pic 1. Configuring a wrapped package](image)

While configuring a package, you should provide a package name – this name is also used as the installation package name while deploying a generated MSI package. If additional files are required to install a wrapped package, they should be added to the **Additional Files** table, using the **Add File** and **Add Folder** buttons on the toolbar. You can add only files and folders that are located in the same folder as the setup file. In case the additional parameters are required to be passed to the installer during a generated MSI package installation, they should be provided to the **Parameters** field within the **Install Parameters** group.
It is possible to use the property definition placeholders when configuring both the installer parameters and maintenance parameters. See the Property Definition Placeholders section of this document for the list of available placeholders. Together with the common properties, you can also use the \${PackageSetupFile}\$ and the \${PackageSetupDir}\$ placeholders. The \${PackageSetupFile}\$ placeholder stands for the path to the setup file during the package deployment and the \${PackageSetupDir}\$ is replaced with the path to the setup file location.

To be able to repair and uninstall this wrapped package, you should enable the Allow Repair and Allow Uninstall options within the Maintenance Parameters group. In case if the wrapped package is an MSI, you can define the Windows installer parameters used for uninstall and repair, if required. Otherwise, if the wrapped package is an executable installer, you should define a command line to be executed to uninstall and/or repair the package.

**Move Up**
The Move Up button from the Order group on the contextual Wrapped Packages ribbon page should be used to move the selected packages up the deployment order.

**Move Down**
The Move Down button from the Order group on the contextual Wrapped Packages ribbon page should be used to move the selected packages down the deployment order.

By default, the wrapped packages deployment order is the same as the addition order, but you can reorder the packages using the Move Up and Move Down items from the pop-up menu or the Order group on the contextual Wrapped Packages Ribbon page. Please take into account, that the install, repair and uninstall order for the wrapped packages is the same.

**Edit**
The Edit button from the Management group on the contextual Wrapped Packages Ribbon page should be used to change the selected wrapped package configuration.

**Delete**
The Delete button from the Management group on the contextual Wrapped Packages Ribbon page allows you to delete the selected wrapped packages from the selected project.

**Copy To**
The Copy To button from the Management group on the contextual Wrapped Packages Ribbon page should be used to copy the selected wrapped packages to another project.

**Move To**
The Move To button from the Management group on the contextual Wrapped Packages Ribbon page allows you to move the selected wrapped packages to another project.

The wrapped packages can be changed, deleted and transferred between projects. To change the wrapped package, select it and either double-click, or choose the Edit item in pop-up menu or on the contextual Wrapped Packages Ribbon page. While editing a wrapped package, you can define the same scope of properties as during its creation. To delete the wrapped from the project, select it and choose the Delete item in the pop-up menu or on the contextual Wrapped Packages Ribbon page. The well-known drag/drop and copy/paste techniques are fully supported for copying and moving wrapped packages between projects. You can also use the Copy To and Move To buttons from the Management group on the contextual Wrapped Packages Ribbon page to transfer the selected wrapped packages to a different project. While using these buttons you are proposed to select a project from those available in a dialog and confirm your selection.
As you can see, the process of wrapping existing installations is fast and easy, but if you are going to deploy the generated MSI package in a silent mode, you should pay a special attention to the parameters passed to the wrapped package installer.

**Why is it important to supply installer command line parameters?**

By default, most of the installation setup programs guide the end user through the setup wizard, thus the interaction with the user is required to complete the setup. So if you are going to deploy a generated MSI package in a silent mode, it is required the wrapped installation is performed without any interaction with the user. If the package is configured incorrectly, the deployment process may pause to wait for user input, and if there is no user during the remote deployment process, it will hang indefinitely. Installation setup programs mostly support silent execution, but the execution parameters may vary depending on the installation vendor.

Now let us take a look at the importance of an answer file. In some cases, the installer simply does not provide a default installation sequence without any user input, so a silent deployment without any additional configuration is impossible. Such installers commonly provide a technique for recoding an answer file to perform the deployment saving the user answers to that file. Another example is setups that use the same command for uninstall and repair and simply ask you what you would like to do on the first step of the installation wizard. If you launch an uninstall using this command with a default scenario, the installation setup may simply perform a repair, and that is not what you are expecting. Therefore, the required answer files, if any, must be added to the wrapped package and used while specifying the installer command line parameters [Pic 1].

As a conclusion, we should emphasize that providing the parameters to be passed to the installer setup and the answer file, if required, is always a must. You should contact the installation package vendor or should do Internet search to get the proper parameters to be passed to each executable installer setup.
Capturing Changes

MSI Package Builder comes with unique monitoring technology that provides you with an ability of tracking changes made to your operating system and create a project containing those changes. This feature can be used in a scope of the low-level repackaging process for creating a deployment package based on those changes. You can also use it to simply review the changes performed to the underlying operating system during the monitoring session.

### Start Monitoring

The Start Monitoring button from the Capture group on the Home Ribbon page should be used to start a new session for tracking changes performed to the system.

### Stop Monitoring

The Stop Monitoring button from the Capture group on the Home Ribbon page should be used to stop the currently running monitoring session.

To start a new monitoring session to capture the changes performed to the underlying operating system, you can press the Start Monitoring button from the Capture group on the Home Ribbon page or choose the Start Monitoring item from the Projects view pop-up menu. The wizard will appear on the screen, where you can provide a name for the project to be created on a basis of the changes monitored and choose additional processes to be filtered during the monitoring process. The monitoring modules will ignore the activity of the filtered processes, thus the changes performed by those processes will not be included into the resulting set. As soon as the monitoring is started, you can perform the changes to be monitored, thus run the applications you are going to track activity of and/or perform the changes manually.

To stop the capturing procedure, press the Stop Monitoring button from the Capture group on the Home Ribbon page or choose the Stop Monitoring item from the Projects view pop-up menu. When the monitoring process is stopped, the project with the specified name is created, containing all the changes performed to the underlying operating system while the monitoring was active. If you want to create a deployment package based on these changes, you should first prepare the created project. For detailed information about the projects preparation steps, refer to the Projects Preparation section of this document.

This feature is also used both in automated and advanced monitoring repackaging modes, but the preparation steps, as well as the MSI package creation steps, are automated. As you have now been introduced to the capturing abilities, you should be able to use those abilities in your everyday work.
Creating MSI Package

MSI Package Builder was designed to repackage existing installations into MSI packages, so one of the main aspects of the program is a flexible configuration of the package being created. The package can be created during the installation repackaging process, or manually from an existing MSI Package Builder project.

Create MSI Package

The Create MSI Package button from the Builder group on the Home Ribbon page should be used to generate an MSI package based on the selected project.

To create an MSI package manually, you can use the Create MSI Package item from the Projects view pop-up menu or from the Builder group on the Home Ribbon page. While creating an MSI package based on a project the options for package created are filled automatically using the values defined in the Project Details view, and during the repackaging process they are defined on-line and then saved to the created project. Let us take a look at the available options and describe each one, starting with the general product information Pic 1.

![Pic 1. Specifying general MSI package information](image)
The general information consists of the **Product Name**, **Platform**, **Manufacturer**, **Icon**, **Language** and **Version**. The product name, manufacturer and icon are used to display the product in the **Programs and Features** section of the Windows control panel. The platform is used to define if the installer is deploying an x86 or an x64 application. The **Language** field is responsible for defining a language and encoding used by an installer package.

The version is the most important field of the general information, because it is responsible for the correct package installation and upgrade. That means that when you are preparing an updated package, using the same project, you must change the version to a bigger. This is also displayed in the **Programs and Features** section of the Windows control panel.

The other aspect of the general product information is an installation context. You can choose if the package when launched without additional parameters should be installed only for a current user account, or for all users. This is defined by switching between the **Install MSI to Current User** and **Install MSI to All Users** radio buttons. Please note, that the administrative privileges are required to install the MSI package for all users. You can also choose if the generated MSI package should be displayed in the **Programs and Features** section of the Windows control panel, and if the repackaged installations, if any, should also be displayed. If the reboot is required to complete the package installation, an appropriate options should be selected in the **Restart Mode** drop-down list.

![Pic 2. Package Identification & Upgrade](image-url)
The next step is configuring the package identification and upgrade abilities. For MSI packages to be deployed and upgraded correctly, they should be properly identified. For each MSI package there is a **Product GUID**, that uniquely identifies the product, and **Upgrade GUID** that is used for matching upgrades. Windows Installer maintains its database and handles its consistency, depending on these identifiers, so you must be very careful with them. You may not provide the upgrade identifier, but in this case, the upgrade will not be supported. If you are going to support upgrades, you can define if the upgrade is language specific and if downgrade is allowed. In case you choose that the upgrade process is language specific, then it will be allowed to install a new version of an MSI package only if its language is the same as the language of the previous version installed. If you disable the protection from downgrade it will be possible to install a previous version of the product even when the newer version is installed, replacing the newer version with the previous one.

For the successful upgrade it is also required that the installed MSI package should have the same installation context (Current User/All Users) defined as the MSI used for update. It is also required that at least one of the first three version figures of the MSI package used for update differs from the version of the installed MSI package.

![Image of Create MSI Package window](image)

**Pic 3. Overriding the common package signing options**
Before generating the MSI package, you can choose if the common package signing options should be used, or they should be overridden for this specific project [Pic 3]. As when defining the common settings, you can choose if the digital signature should be added to the package, select the signing certificate from the certificate storage and choose the time server used to generate a digital signature time stamp.
The next aspect of an MSI package configuration is install options. You can specify the list of supported operating systems and choose the reinstall mode used if the files deployed by an MSI package already exist in the target location. You can also define the Microsoft .NET Framework versions required by deployed software packages – the Windows Installer will check for those versions and interrupt the deployment process if the required version is missing on a target PC.

![Image of MSI Package Builder Enterprise 7 interface]

The last thing to define, that is also optional, is a set of support and summary properties. Those are available in the package properties and are displayed in the Programs and Features section of the Windows control panel. As for the support information, you can use the properties defined in the program preferences, or override them for this package.

When all the required settings are configured, either within the Project Details view or within the package creation wizard, all you need is to provide a path to save the resulting MSI package to and press Finish. As soon as the package is created, it is ready for deployment.
MSI Package Language

The fact that any MSI package can target only one language is a well-known limitation of the Windows Installer technology. Within MSI Package Builder, the installer language is selected during the project configuration and can be changed within the Project Details view. As any MSI package is language specific, for a successful deployment, all the characters that may be present in any project entry should be available in the code page of the chosen language when the package is being generated.

To help you avoid language mismatches that are really hard to detect on your own, MSI Package Builder performs the code page verification before generating each MSI package and warns you if the test has not been passed successfully allowing you to choose the resolution mode for the detected conflicts [Pic 1].

![Pic 1. Code page verification failure]
There are actually two types of conflicts: those that can be resolved automatically and those that cannot. As for the conflicts that cannot be resolved automatically, you can either resolve them manually or exclude the entries causing such conflicts from the resulting MSI package. As you can see, the displayed warning allows you to choose among three options.

The first options, which is **Process All Conflicts Automatically**, means that it is up to MSI Package Builder to choose what to do with each entry causing a code page conflict. It is the easiest and the most transparent choice for the user. If his option is used, the program performs necessary changes to the entries containing conflicts that can be resolved automatically. The entries containing conflicts that cannot be resolved automatically are excluded from the resulting package. This options is suitable in most cases, but you should always check the resulting package for correctness in case you are letting MSI Package Builder to perform automatic conflicts resolution.

The next option, which is **Process Each Conflict Manually**, is used to let you decide what to do with each conflicting entry on your own. This option provides the maximum flexibility but requires a great responsibility. Using this method, you can choose for each entry whether you would like to edit it to resolve the conflict, or to apply automatic conflicts resolution to the entry (if applicable), or to exclude the entry from the resulting package. You can also choose to automatically resolve or skip all conflicts using the buttons on the toolbar above the entries list. For each entry in the list, you can see its status, which is either **Conflicting**, or **Resolved**, or **Skipped**.

The last method of resolving code page conflicts is changing the target language of the resulting MSI package. The **Change MSI Package Language** option is used in such a case. As soon as you select the required language from the drop-down list, you are proposed to revalidate the project. If the validation is passed, the resulting MSI package is generated.

We have done our best to help you avoid problems caused by the language limitation of the Windows Installer technology and hope you will find the code page verification process with different conflict resolution methods useful.
MSI Installation Context

Windows Installer can install a package on a computer within two installation contexts: per-machine and per-user. A per-machine package installation is required if all users of the computer need to access and use the application. Based on the context, Windows Installer automatically redirects the values of the folder properties and registrations to the locations for a per-user or per-machine installation. The installation context is specified when configuring the MSI Package Builder project.

When a package is installed for a current user, it is not visible in the Programs and Features section of the control panel for other users. In most cases, the deployed content is also accessible only for the user that deployed the package, though it is not always true. To install a package per-user, you should choose the Install MSI to Current User radio button during the project configuration.

To create a package targeting a per-machine installation, you should choose the Install MSI to All Users radio button. As for the per-machine installation, there is some specifics in handing the user-specific folders and registry keys defined in the package. By default, when a per-machine installation is deployed and the package contains user-specific files and registry entries, such files and entries are created only for the user running the deployment. For other users, only the common files and registry entries are deployed. MSI Package Builder allows you to configure the package in such a way that user-specific entries will also be installed for other users on their first log-on after the package deployment. This feature is enabled by means of the Smart All Users Mode box.

For the smart per-machine installation to work correctly, the MSI package should be accessible to users logged in future via the same path as used during the initial deployment. In case the package is not available, the user-specific files and registry entries won't be created for other users.

To obtain proper deployment results, it is recommended that the default installation context be chosen carefully taking into account the specifics of each and every package.
Signing Packages

Digitally signing files helps protect against changes to a file by validating that a hash of the current file matches the hash stored in the digital signature. Digital signatures also help verify that a package came from a particular publisher by encrypting the hash with the publisher's private key. Verifying the signature using the publisher's public key or a trusted certificate authority that signed their public key validates the publisher.

You can sign Windows Installer packages to help guarantee that users know if your packages have been modified and that they came from you, the publisher. Windows Installer validates that a package hasn't been changed if it contains a digital signature when attempting to install it. Properly signed MSI packages can be installed via GPO even within the environments with strict security policies.

About Digital Signatures

A digital signature is based on a signing certificate. A certificate is a set of data that completely identifies an entity, and is issued by a certification authority only after that authority has verified the entity's identity. The data set includes the entity's public cryptographic key. When the publisher of a package signs the package with its private key, the installer can use the publisher's public key to verify the publisher's identity.

In order to perform a package signing operation, both private key and signer identification information must be supplied. The digital certificate used in the signature usually supplies the signer identification information, however. Thus, the private key must be supplied through some other means. Additionally, the signature must include the certificate chain for the cryptographic service provider (CSP), up to a root certificate trusted by the user, in order for the signed file to be authenticated. So in all, there are several items that need to be provided in order to generate a digital signature.

Microsoft has developed a certificate store technology to reduce the above complexity. Using this technology, when a user enrolls to obtain a certificate, they specify the private key information, the CSP information, and the certificate store name for the certificate. The certificate will then be stored in the certificate store and be associated with the other items. When the user wants to sign a package, they only need to identify the certificate in the certificate store. The code-signing tool will retrieve the certificate, the private key, and the certificate chain for the CSP, all based on the specified certificate.

When signing a package, a trusted time server is used to generate a time stamp for a digital signature. This is performed, to guarantee that the package is signed with the certificate that is neither expired nor revoked.

Requirements for a Digital Certificate

For a digital certificate to be used by MSI Package Builder for signing the generated packages, the following set of requirements must be met:

1. The certificate must include the **Code Signing (1.3.6.1.5.5.7.3.3)** within its Intended Purpose.

2. The certificate's **Valid From** date must be less and the **Valid To** date must be greater than the package signing date.
If you would like to use the certificate from certificates storage:

1. The digital certificate must be placed in the Current User certificates storage.
2. Both the private key and the signer identification information must be supplied.
3. The private key must be available together with the signing certificate in the certificate storage. In case you have a private key in a separate file, please use the tool provided by Microsoft for preparing the a private & public key pair for importing into the certificates storage as described here: Pvk2Pfx, Combine PVK + SPC to PFX.

In case the digital certificate does not meet the above-stated requirements, MSI Package Builder will not suggest that you use it for signing packages.

**Configuring Packages Signing**

MSI Package Builder allows you both to define the common packages signing configuration to be used for adding digital signatures to generated MSI packages and to override those settings for specific projects. The common digital signing options are specified on the Packages Signing preference page [Pic 1](#), and the overriding feature can be used either when creating an MSI package or in the Project Details view.

![Pic 1. Configuring the package signing options](image)
In any case, if you enable the packages signing, you are proposed to select the signing certificate to be used for creating a digital signature and choose the time server for generating a digital signature time stamp.

The required certificate can be selected from those installed to the above-mentioned certificate storage. To select the certificate from store, check the Select certificate from store radio button and press the button within the following field. The dialog will be displayed to let you choose the certificate from those available. When choosing the certificate, you can press the View Certificate button on the toolbar to view the detailed information on the selected certificate. This information dialog can also be reached via the button from the certificate choosing field when the certificate is already specified. To reset the certificate, you can use the button.

If you would like to use the certificate stored within the private information exchange (pfx) file, check the Select certificate from file radio button press the button within the following field. The file browser dialog will appear on the screen to let you provide the path to the file. Normally, private keys in pfx files are password encrypted, so you’re going to be prompted for the password to use the certificate from the specified file for signing packages. The same as for the certificates chosen from store, you can open the certificate information dialog via the button from the certificate choosing field and reset the certificate via the button.

If you do not have a required certificate, as a workaround it is possible to generate a self-signed certificate for the package being generated. In such a case the pfx file containing the automatically generated certificate will be placed next to the generated package and must be installed on the client machines before deploying the package. Please refer to the How to install the auto-generated self-signed certificate? section for details. To use the described approach, check the Use auto-generated self-signed certificate radio button and provide the password to protect private key to the Certificate password field.

As for the time server, you can either choose the one from those predefined in the Time Server field or provide the address of another trusted server that can generate a time stamp for digital signatures. A time stamp should always be added when signing a package. Although it is strongly recommended that a digital signature time stamp be added immediately when signing packages with MSI Package Builder, you can leave the Time Server field empty, thus skipping the time stamping. In case a time stamp is not added, it is possible to time stamp a signed package in future with the help of the sign tool.

If the package signing is enabled, MSI Package Builder will add a digital signature using the specified signing certificate and chosen time server when a package is generated. If there are any problems occurred during the signing process, they are added to the Log view.

### How to install the auto-generated self-signed certificate?

When using auto-generated self-singed certificate, it should be installed on client Machines before deploying generated packages. The following steps should be followed to install the certificate:

2. Select the **Local Machine** store and press **Next**.

3. Double check that you are importing the certificate from the appropriate pfx file and press **Next**. [PIC]
4. Provide the password you have specified in the **Certificate password** field to the **Password** field and press **Next**.
5. Select the **Place all certificates to the following store** radio button, press the **Browse** button and choose the **Trusted People** store. Then press **Next**.

![Certificate Import Wizard](image)

6. After checking the import settings on the final wizard page, press **Finish**.

As soon as the certificate is imported to the **Trusted People** store, the packages signed with the certificate are considered trusted thus fully secure, so you can successfully install MSI packages signed with such certificate via GPO.

As you can see, using a self-signed certificate as a workaround has a set of limitations, but allows you to get a signed packaged that can be trusted within your organization or by specific PCs only.

**Importing Package**

With MSI Package Builder you can not only create new deployment packages but also modify existing ones using the embedded packages import capability. Just choose a deployment you want to modify, import it to a new project, perform the changes needed and build a new one. This feature is designed to handle the situation, when you have generated a deployment package, deleted the project, but need to make some changes to the package or upgrade it.

Though it is guaranteed that the import of the deployment package built with any previous version of MSI Package Builder is fully supported, some MSI features are not available in the MSI Package Builder import technology. So if you are importing a generic deployment package, please check the import results and a newly generated deployment package carefully.

The feature of creating a new deployment package on a basis of an existing deployment package is available in the **Project Setup** wizard. Alternatively, you can use the **Import Package** button from the **Project Management** group on the **Home** Ribbon page or the **Import Pakcage** link in the **Product Actions** group on the **Welcome Screen**.
The **Import Package** button from the **Project Management** group on the **Home** Ribbon page should be used to create a new MSI Package Builder project based on a changes performed by an existing deployment package.

In any case, you will reach the page where you are proposed to provide the path to the existing deployment package to import and a name for the project to be created as a result **Pic 1**.

![Import Package](image)

**Pic 1. Importing an existing deployment package**

When the import is complete, you can see the project that contains the changes performed by the specified deployment package created in the **Projects** view.
Chapter 6: Using Placeholders and Filters

With MSI Package Builder, it is possible to define a set of monitoring filters that allow skipping the activity of the defined processes or the changes made to the defined file system items and registry keys during the Live Monitoring process and a set of uninstall filters that allow leaving joint resources on a PC after an MSI is uninstalled. These filters can be defined both as a simple filter value and as a regular expression. The filter condition can contain system folder definition placeholders.

The system folder definition placeholders, as well as the property definition placeholders, can also be used while defining custom actions, wrapped packages, registry changes, etc. In this chapter, we will discuss the ways of specifying filter conditions and provide a detailed description for system folder and property definition placeholders.

What's Inside

How should I correctly specify the filter condition?
System Folder Definition Placeholders
Property Definition Placeholders
How should I correctly specify the filter condition?

With MSI Package Builder, it is possible to define a set of monitoring filters that allow skipping the activity of the defined processes or the changes made to the defined file system items and registry keys during the Live Monitoring process and a set of uninstall filters that allow leaving joint resources on a PC after an MSI is uninstalled. In this chapter, we will show you how to specify a filter condition. There are two methods of specifying a filter condition – those are using a simple filter condition and using a regular expression. Let us take a close look at each method.

Simple Filter Condition

Simple filter condition is a string to be used for matching elements to be filtered. In case of a processes filter this string should be a process name, e.g. explorer.exe. For the file system filters this string should be a file system path to filter – all the sub-directories and files in the path specified are also filtered, for example C:\Windows\ filter value will lead to filtering all data inside Windows directory. The registry filter string should match a path to the registry key – all the sub-keys along with registry values in the path specified are filtered, e.g. choosing HKEY_LOCAL_MACHINE as root key and setting Software\Microsoft\Windows NT as key will filter all data inside Windows registry key.

While specifying a simple filter condition you can use wildcard characters same as for file system search, for example setting a file system filter value to ?:\Temp* will filter all the folders that start with Temp from all logical drive, such as C:\Temp, D:\Temporary Data, etc.

Regular Expressions

Using regular expressions is a more advanced way of a filter condition specification. Regular expression is a string that is used to match a set of strings, particular characters, words, or patterns of characters according to certain syntax rules that are described below. Regular expressions usage is a more flexible way of specifying a filter value than a simple filter condition, because one regular expression can cover lots of items to be filtered. For example:

^${CommonUserFolder}$\[^\]+\Local Settings\!(Application Data\).+$

This expression will filter all the folders from the Local Settings folder of each user except the Application Data folder.

The following syntax should be used to define a filter condition with a help of regular expression.

<table>
<thead>
<tr>
<th>Metacharacter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Matches any single character.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Indicates a character class. Matches any character inside the brackets (for example, [abc] matches a, b, and c).</td>
</tr>
<tr>
<td>^</td>
<td>If this metacharacter occurs at the start of a character class, it negates the character class. A negated character class matches any character except those inside the brackets (for example, [^abc] matches all characters except a, b, and c).</td>
</tr>
<tr>
<td></td>
<td>If ^ is at the beginning of the regular expression, it matches the beginning of the input (for example, ^[abc] will only match input that begins with a, b, or c).</td>
</tr>
<tr>
<td>-</td>
<td>In a character class, indicates a range of characters (for example, [0-9] matches any of the digits 0 through 9).</td>
</tr>
</tbody>
</table>
Using Placeholders and Filters

<table>
<thead>
<tr>
<th>Metacharacter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Indicates that the preceding expression is optional: it matches once or not at all (for example, <code>[0-9][0-9]?</code> matches <code>2</code> and <code>12</code>).</td>
</tr>
<tr>
<td>+</td>
<td>Indicates that the preceding expression matches one or more times (for example, <code>[0-9]+</code> matches <code>1</code>, <code>13</code>, <code>456</code>, and so on).</td>
</tr>
<tr>
<td>*</td>
<td>Indicates that the preceding expression matches zero or more times.</td>
</tr>
<tr>
<td>??, +?, *?</td>
<td>Non-greedy versions of ?, +, and *. These match as little as possible, unlike the greedy versions that match as much as possible (for example, given the input <code>&lt;abc&gt;&lt;def&gt;</code>, <code>&lt;.*?&gt;</code> matches <code>&lt;abc&gt;</code> while <code>&lt;.*&gt;</code> matches <code>&lt;abc&gt;&lt;def&gt;</code>).</td>
</tr>
<tr>
<td>( )</td>
<td>Grouping operator (for example <code>(\d+,)*\d+</code> matches a list of numbers separated by commas, such as <code>1</code> or <code>1, 23, 456</code>).</td>
</tr>
<tr>
<td>{ }</td>
<td>Indicates a match group.</td>
</tr>
<tr>
<td>\</td>
<td>Escape character: interpret the next character literally (for example, <code>[0-9]</code> matches one or more digits, but <code>[0-9]\+</code> matches a digit followed by a plus character). Also used for abbreviations (such as <code>\a</code> for any alphanumeric character – see the following table for details).</td>
</tr>
<tr>
<td>$</td>
<td>At the end of a regular expression, this character matches the end of the input (for example, <code>[0-9]$</code> matches a digit at the end of the input).</td>
</tr>
<tr>
<td>!</td>
<td>Negation operator: the expression following ! does not match the input (for example, <code>a!b</code> matches <code>a</code> not followed by <code>b</code>).</td>
</tr>
<tr>
<td></td>
<td>The following abbreviations can be used in regular expressions.</td>
</tr>
<tr>
<td>\a</td>
<td>Any alphanumeric character: <code>([a-zA-Z0-9])</code></td>
</tr>
<tr>
<td>\b</td>
<td>White space (blank): <code>([ \t])</code></td>
</tr>
<tr>
<td>\c</td>
<td>Any alphabetic character: <code>([a-zA-Z])</code></td>
</tr>
<tr>
<td>\d</td>
<td>Any decimal digit: <code>([0-9])</code></td>
</tr>
<tr>
<td>\h</td>
<td>Any hexadecimal digit: <code>([0-9a-fA-F])</code></td>
</tr>
<tr>
<td>\n</td>
<td>Newline: <code>([^\r\n])</code></td>
</tr>
<tr>
<td>\q</td>
<td>A quoted string: `(&quot;[^&quot;&quot;]&quot;)</td>
</tr>
<tr>
<td>\w</td>
<td>A simple word: <code>([a-zA-Z])</code></td>
</tr>
<tr>
<td>\z</td>
<td>An integer: <code>([0-9])</code></td>
</tr>
</tbody>
</table>

Now you are introduced to the possible ways of specifying a filter condition and should be able to provide any condition to solve any problem.
**System Folder Definition Placeholders**

MSI Package Builder allows you to use system folder definition placeholders while configuring filters in the program preferences and when providing file system modifications, registry modifications, etc. These placeholders are automatically resolved to the corresponding file system paths during an installation. The system folder placeholder should be proceeded with \$\{ and succeeded with \}$, e.g. \$\{ProgramFilesFolder\}$.

Below is a list of the available placeholders for system folder definitions.

<table>
<thead>
<tr>
<th>Placeholder</th>
<th>Replacement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdminToolsFolder</td>
<td>The file system directory that is used to store administrative tools for the current user.</td>
<td></td>
</tr>
<tr>
<td>AppDataFolder</td>
<td>The Application Data folder for the current user.</td>
<td></td>
</tr>
<tr>
<td>CommonAdminToolsFolder</td>
<td>The file system directory that contains administrative tools for all users of the computer.</td>
<td></td>
</tr>
<tr>
<td>CommonAppDataFolder</td>
<td>The Application Data folder that is common for all users.</td>
<td></td>
</tr>
<tr>
<td>CommonDesktopFolder</td>
<td>The folder for the Desktop content that is common for all users.</td>
<td></td>
</tr>
<tr>
<td>CommonDocumentFolder</td>
<td>The file system directory that contains documents that are common to all users.</td>
<td></td>
</tr>
<tr>
<td>CommonFiles64Folder</td>
<td>The Common Files folder for 64-bit applications.</td>
<td></td>
</tr>
<tr>
<td>CommonFilesFolder</td>
<td>The Common Files folder for 32-bit applications.</td>
<td></td>
</tr>
<tr>
<td>CommonProgramMenuFolder</td>
<td>The folder for the All Programs menu content that is common for all users.</td>
<td></td>
</tr>
<tr>
<td>CommonStartMenuFolder</td>
<td>The folder for the Start menu content that is common for all users.</td>
<td></td>
</tr>
<tr>
<td>CommonStartupFolder</td>
<td>The folder for the programs executed on startup for all users.</td>
<td></td>
</tr>
<tr>
<td>CurrentUserProfileFolder</td>
<td>The current user's profile folder.</td>
<td></td>
</tr>
<tr>
<td>DesktopFolder</td>
<td>The current user's Desktop content folder.</td>
<td></td>
</tr>
<tr>
<td>FavoritesFolder</td>
<td>The Favorites folder for the current user.</td>
<td></td>
</tr>
<tr>
<td>FontsFolder</td>
<td>The folder used to install fonts to.</td>
<td></td>
</tr>
<tr>
<td>LocalAppDataFolder</td>
<td>The folder that serves as a data repository for local (nonroaming) applications for the current user.</td>
<td></td>
</tr>
<tr>
<td>MyPicturesFolder</td>
<td>The Pictures folder for the current user.</td>
<td></td>
</tr>
<tr>
<td>NetHoodFolder</td>
<td>The file system directory that contains the link objects that may exist in the My Network Places virtual folder.</td>
<td></td>
</tr>
<tr>
<td>PersonalFolder</td>
<td>The Documents folder for the current user.</td>
<td></td>
</tr>
<tr>
<td>PrintHoodFolder</td>
<td>The file system directory that contains the link objects that can exist in the Printers virtual folder.</td>
<td></td>
</tr>
<tr>
<td>ProgramFiles64Folder</td>
<td>The Program Files folder for 64-bit applications.</td>
<td></td>
</tr>
<tr>
<td>ProgramFilesFolder</td>
<td>The Program Files folder for 32-bit applications.</td>
<td></td>
</tr>
</tbody>
</table>
## Placeholder and Filters

### Placeholders and Replacement

<table>
<thead>
<tr>
<th>Placeholder</th>
<th>Replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProgramMenuFolder</td>
<td>The folder for the <strong>All Programs</strong> menu content for the current user.</td>
</tr>
<tr>
<td>RecentFolder</td>
<td>The file system directory that contains shortcuts to the user's most recently used documents.</td>
</tr>
<tr>
<td>SendToFolder</td>
<td>The folder for the <strong>Send To</strong> menu content for the current user.</td>
</tr>
<tr>
<td>StartMenuFolder</td>
<td>The folder for the <strong>Start</strong> menu content for the current user.</td>
</tr>
<tr>
<td>StartupFolder</td>
<td>The folder for the programs executed on startup for the current users.</td>
</tr>
<tr>
<td>System16Folder</td>
<td>The <strong>System folder</strong> for 16-bit libraries.</td>
</tr>
<tr>
<td>System64Folder</td>
<td>The <strong>System folder</strong> for 64-bit libraries.</td>
</tr>
<tr>
<td>SystemFolder</td>
<td>The <strong>System folder</strong> for 32-bit libraries.</td>
</tr>
<tr>
<td>TempFolder</td>
<td>The file system directory that is designated for temporary files.</td>
</tr>
<tr>
<td>TemplateFolder</td>
<td>The <strong>Templates</strong> folder for the current user.</td>
</tr>
<tr>
<td>UserProfilesFolder</td>
<td>The system user profiles folder.</td>
</tr>
<tr>
<td>WindowsFolder</td>
<td>The folder the operating system is installed to.</td>
</tr>
<tr>
<td>WindowsVolume</td>
<td>The volume the operating system is installed on.</td>
</tr>
</tbody>
</table>

Please note that specific system folders can be indefinite on specific operating systems, so you should make sure that the operating systems you have defined as target during a deployment package creation process support the folders you have defined.

### Property Definition Placeholders

MSI Package Builder allows you to use both standard and user-defined MSI property values when configuring custom actions, registry modifications, etc. To access property values during the deployment process, you should use the property definition placeholders. The property definition placeholder is a property name proceeded with `$` and succeeded with `$`, e.g. `$\{Manufacturer\}$`

You are already familiar with a part of those properties, which are the **system folder definition placeholders**. To review the full list of available standard properties, please refer to the Property Reference article on MSDN. The other properties are package specific and are always spelled with capital letters. For the properties spelled with all capital letters, it is possible to override their default values during an MSI package deployment, e.g.

```
msiexec.exe /i installer.msi AUTOUPDATE=1;REGISTEREDOWNER=Dreamlight;REINSTALLMODE=omus
```
Chapter 7: Log

The log is designed to store information on the events taking place while the program is running. Most of them are messages sent by the operations and operations results. Such events are displayed in the Log view and can significantly help you to analyze the operation execution results and troubleshoot problems taking place while the program is in use. The log may grow continually, thus slowing down the program loading and response time. To prevent this, it can be cleared manually at any moment.

In this chapter, we will explain how to analyze the log to ensure that the operations are completed successfully or to troubleshoot possible problems and describe the option of exporting the log to a simple format.

What’s Inside

Analyzing Log
Exporting Log

Analyzing Log

The main purpose of the Log view is to help you understand if the operation execution has succeeded and troubleshoot problems if any have occurred. Each entry in the log has a severity icon, a title and a description. From the title, you can understand which operation has been performed; the description provides you with the result message and a hint on solving the problem, if any; and the severity icon can be used to quickly understand if the operation has fully succeeded.

For example, let us take a closer look at the following result set in the log Pic 1.

![Pic 1. Sample logged events](image)

The picture above shows the set of result we received after performing an operation. As we can see, most of the logged events are informational, but some stand for problems. We need to find out what caused the problem and what should be done to avoid it in future. In addition, it may be interesting to go through the warnings to see if anything wrong is going on.

After the results have been reviewed and all the problems have been solved, you can run the operation again and ensure that it completes successfully.
Exporting Log

With MSI Package Builder, you can easily export the log to the CSV file format for future analysis or processing by an automated tool. To export the log, click the Export button from the Log view toolbar or choose the Export item from the pop-up menu. The Export Log dialog will appear on the screen [Pic 1].

In the Export Log dialog, you are offered to choose a file you are going to save the log data to. The file path should be provided to the Export To field. After you are ready with configuring the export, press OK to proceed. The file containing the log data will be created in the path specified.

The CSV file with the exported data consists of four columns, which are the following:

<table>
<thead>
<tr>
<th>Column Index</th>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Title</td>
<td>The logged event title (including the path).</td>
</tr>
<tr>
<td>2</td>
<td>Severity</td>
<td>The logged event severity level.</td>
</tr>
<tr>
<td>3</td>
<td>Description</td>
<td>The logged event description.</td>
</tr>
<tr>
<td>4</td>
<td>Time</td>
<td>The time when the event occurred.</td>
</tr>
</tbody>
</table>

Sample exported log data in the CSV format

"Title","Severity","Description","Time"
"Create MSI Package","Information","The operation was completed successfully.","12/3/2013 11:14:50 AM"
"Create MSI Package","Cancel","The operation was canceled by the user.","12/3/2013 11:12:51 AM"
"Create MSI Package\Code Page Verification","Cancel","All code page errors should be resolved for a successful MSI package creation. To find out which items could not pass the code page verification process, review the Application Log.","12/3/2013 11:12:51 AM"
"Create MSI Package\Code Page Verification","Warning","The characters contained in the 'HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Adobe\PDF Owner\PreviousOwner\Acroexch.Document.7\shell\Open' Registry item are not available in the code page selected for the MSI package.","12/3/2013 11:12:43 AM"
"Create MSI Package\Code Page Verification","Warning","The characters contained in the 'HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Adobe\PDF Owner\PreviousOwner\CLSID\{B801CA65-A1FC-11D0-85AD-444553540000}\Verb\0' Registry item are not available in the code page selected for the MSI package.","12/3/2013 11:12:43 AM"
"Create MSI Package\Code Page Verification","Warning","The characters contained in the 'HKEY_LOCAL_MACHINE\SOFTWARE\Classes\AcroExch.Document.7\shell\Open' Registry item are not available in the code page selected for the MSI package.","12/3/2013 11:12:43 AM"

Now you are introduced to the log export and export data file format and can use the export feature for the log analysis without any misunderstanding.
Chapter 8: Program Preferences

MSI Package Builder comes with a wide range of settings available for changing by any user. Every preference page has a detailed description of its content and of the feature it is used to configure. To reach the program preferences, click the Preferences button available from the Application Menu. Besides, the clickable Ribbon groups’ glyphs open the preference pages that configure the functionality incorporated in the respective group.

What’s Inside

MSI Package Builder Part
Filters Part
Miscellaneous Part

MSI Package Builder Part

The MSI Package Builder part of the program preferences is used to configure the main program settings, such as the projects storage location, project signing options and user interface settings. To open the Preferences dialog, click the Preferences button available from the Application Menu. Feel free to configure the available settings to suit your needs best.

What's Inside

Projects Configuration Page
Package Signing Page
User Interface Page
Projects Configuration Page

MSI Package Builder is a project based program – this means that all data is grouped by the projects. Each project contains one or several packages. Such data structure allows grouping the coupled installations, storing them and managing them as the unit. The projects are saved to the projects storage. To define a projects storage location, open the program preferences using the Preferences button from the Application Menu and click the Projects Configuration link on the navigation bar on the left of the Preferences dialog within the MSI Package Builder group [Pic 1].

The path to the folder to organize the MSI Package Builder projects storage in should be provided to the Path field within the Projects Root Directory group.

On the Projects Configuration preference page, you can provide the default support information to be used for all deployment packages, if not overridden for a specific project.
Package Signing Page

MSI Package Builder allows you to sign generated deployment packages to help guarantee that users know if your packages have been modified and that they came from you, the publisher. To configure the signing options, open the program preferences using the Preferences button from the Application Menu and click the Package Signing link on the navigation bar on the left of the Preferences dialog within the MSI Package Builder group (Pic 1).

Within the Digital Signature group, you can define if generated deployment packages should be signed, select the certificate to be used for a digital signature (either from those available in the certificate storage, or from the file, or use auto-generated one) and choose the time server that should be used for creating a time stamp for the digital signature.

The package signing options defined in the program preferences are used as default for each project, but it is possible to override those options for each specific project on the Project Details view or while creating an MSI package from the project.

Refer to the Signing Packages section of this document for detailed information on the purpose and the process of packages signing.
User Interface Page

MSI Package Builder is a multifunctional tool, which can be used as an easy-to-use installations repackager and as an advanced packages editor. That is why different user interface configuration may be required for different persons. The configuration is performed on the User Interface preference page. To access this page, click the Preferences button from the Application Menu and select the appropriate link in the navigation bar, on the left in the Preferences dialog within the MSI Package Builder group.

![Pic 1. Configuring user interface](image)

For those who use it as a simple repackager, it might be enough to walk through the repackaging wizard and build a resulting deployment package, so it would be convenient for them to display the Repackage Installation wizard on start up. However, it is not convenient for those who are using the program also for managing the software packages or creating their own packages. It is also possible to define if the deployment package creation is always run in background.

The other aspect is a decision making. By default, after each deployment package creation the program asks if the folder containing the newly created package should be opened in Windows Explorer. If your answer is always the same, you can answer this question only once, on the User Interface page.
Filters Part

During the monitoring process MSI Package Builder records the modifications performed by all processes that operate in the system. Thus monitoring results can include the modifications performed by system processes. These modifications do not refer to the application or installation that you have monitored and that is why they should be excluded from monitoring results. This goal can be reached using the Monitoring Processes Filters configuration. While using process filters, the changes made by those processes are automatically excluded from the resulting deployment package. In the same manner, there are registry keys and file system paths that should be excluded from the resulting deployment package. The Monitoring Registry Filters and Monitoring File Filters configuration allow you to fulfill this.

When the installation creates a joint resource that is commonly used by several applications, such resource should be indicated as permanent. Permanent files, keys and registry entries are not deleted during application uninstall process to prevent functionality violation of applications that use these resources. You are able to set permanency option for each resource in separate during its edit but it seems to be quite a complex way. To simplify the operation with permanent resources MSI Package Builder comes with a set of predefined uninstall filters and allows you to add some.

All the filters in MSI Package Builder are set up in the program preferences. Press the Preferences button from the Application Menu and you'll find the filtration settings under the Filters group.

What's Inside

Monitoring File Filters Page
Monitoring Registry Filters Page
Monitoring Processes Filters Page
Uninstall File Filters Page
Uninstall Registry Filters Page
Monitoring File Filters Page

The Monitoring File Filters are used to allow the Live Monitoring technology to skip the changes performed to specific files and folders. This prevents unnecessary changes from being included into a deployment package based on monitoring results. To configure the Monitoring File Filters open the program preferences using the **Preferences** button from the **Application Menu** and click the **Monitoring File Filters** link on the navigation bar on the left of the **Preferences** dialog within the Filters group [Pic 1].

![Pic 1. Configuring Monitoring File Filters](image)

The filters are divided into two categories; those are **User Filters** and **System Filters**. The system filters are predefined ones and cannot be deleted, but can be disabled, if required, using the **Disable Selected** item from the pop-up menu. You can then re-enable any filter using the **Enable Selected** menu item. To enable/disable a filter condition and to check if it is enabled, you can use the check boxes located on the left of each row representing the filter condition. It is also possible to refuse from filtering file system items by disabling the **Use the following filter conditions** option.

**Toolbar Overview**

| **Add Filter** | The **Add Filter** button should be used to add a new user-defined Monitoring File Filter condition. |
| **Edit**       | The **Edit** button allows you to change the selected user-defined Monitoring File Filter condition. |
As for the user filters, those are the user-defined ones. To add a new filter condition, use the Add Filter item from the pop-up menu or press the Add Filter button on the toolbar.

When configuring a Monitoring File Filter condition [Pic 2], you should provide a file system path to be filtered. All changes to the specified path are ignored during a monitoring process. You can provide the path manually or select it through the file picker using the button built into the edit box. It is possible to use the system folder definition placeholders while specifying the filter condition, these placeholders are replaced automatically with corresponding file system paths during an installation monitoring process. See System Folder Definition Placeholders section of this document for the list of available placeholders.

Each system folder definition should be proceeded with $\{ and succeeded with $\}, e.g. $\{ProgramFilesFolder}\$.

The and buttons built into the edit box can help you with replacing specific paths to their system folder definitions and vice versa. If you enable the Use a regular expression while specifying the filter condition option, you can define a filter that will exclude all paths matching the expression from monitoring results. See the How should I correctly specify the filter condition? section of this document for a detailed description of possible conditions.

To edit a user-defined Monitoring File Filter select it in the filters table and choose the Edit menu item from the pop-up menu or press the Edit button on the toolbar. The filter condition edit process is similar to the above-stated creation process. To delete the user-defined conditions that are no longer needed, you can select those conditions in the filters table and choose the Delete menu item from the pop-up menu or press the Delete button on the toolbar. It is also possible to delete all user-defined conditions using the Clear User Filters menu item and the corresponding button on the toolbar.
Monitoring Registry Filters Page

The Monitoring Registry Filters are used to allow the Live Monitoring technology to skip the changes performed to specific registry keys. This prevents unnecessary changes from being included into a deployment package based on monitoring results. To configure the Monitoring Registry Filters open the program preferences using the Preferences button from the Application Menu and click the Monitoring Registry Filters link on the navigation bar on the left of the Preferences dialog within the Filters group [Pic 1].

![Pic 1. Configuring Monitoring Registry Filters]

The filters are divided into two categories; those are User Filters and System Filters. The system filters are predefined ones and cannot be deleted, but can be disabled, if required, using the Disable Selected item from the pop-up menu. You can then re-enable any filter using the Enable Selected menu item. To enable/disable a filter condition and to check if it is enabled, you can use the check boxes located on the left of each row representing the filter condition. It is also possible to refuse from filtering registry keys by disabling the Use the following filter conditions option.

Toolbar Overview

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Filter</td>
<td>The Add Filter button should be used to add a new user-defined Monitoring Registry Filter condition.</td>
</tr>
<tr>
<td>Edit</td>
<td>The Edit button allows you to change the selected user-defined Monitoring Registry Filter condition.</td>
</tr>
</tbody>
</table>
Delete
The Delete button allows you to delete the selected user-defined Monitoring Registry Filter conditions.

Clear User Filters
The Clear User Filters button should be used to delete all user-defined Monitoring Registry Filter conditions.

As for the user filters, those are the user-defined ones. To add a new filter condition, use the Add Filter item from the pop-up menu or press the Add Filter button on the toolbar.

When configuring a Monitoring Registry Filter condition, you should define a registry key to be filtered. All changes to the specified key and its sub-keys are ignored during a monitoring process. To define a registry key you firstly choose the root key from the drop down list and then type in the path to the key in the text edit. If you enable the Use a regular expression while specifying the filter condition option, you can define a filter that will exclude changes to all keys matching the expression from monitoring results. See the How should I correctly specify the filter condition? section of this document for a detailed description of possible conditions.

To edit a user-defined Monitoring Registry Filter select it in the filters table and choose the Edit menu item from the pop-up menu or press the Edit button on the toolbar. The filter condition editing process is similar to the above-stated creation process. To delete the user-defined conditions that are no longer needed, you can select those conditions in the filters table and choose the Delete menu item from the pop-up menu or press the Delete button on the toolbar. It is also possible to delete all user-defined conditions using the Clear User Filters menu item and the corresponding button on the toolbar.
Monitoring Processes Filters Page

The Monitoring Process Filters are used to allow the Live Monitoring technology to skip the changes performed by certain processes. This prevents unnecessary changes from being included into a deployment package based on monitoring results. To configure the Monitoring Process Filters open the program preferences using the Preferences button from the Application Menu and click the Monitoring Process Filters link on the navigation bar on the left of the Preferences dialog within the Filters group.

Pic 1. Configuring Monitoring Process Filters

The filters are divided into two categories; those are User Filters and System Filters. The system filters are predefined ones and cannot be deleted, but can be disabled, if required, using the Disable Selected item from the pop-up menu. You can then re-enable any filter using the Enable Selected menu item. To enable/disable a filter condition and to check if it is enabled, you can use the check boxes located on the left of each row representing the filter condition. It is also possible to refuse from filtering processes by disabling the Use the following filter conditions option.

Toolbar Overview

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Filter</td>
<td>The Add Filter button should be used to add a new user-defined Monitoring Process Filter condition.</td>
</tr>
<tr>
<td>Edit</td>
<td>The Edit button allows you to change the selected user-defined Monitoring Process Filter condition.</td>
</tr>
<tr>
<td>Delete</td>
<td>The <strong>Delete</strong> button allows you to delete the selected user-defined <strong>Monitoring Process Filter</strong> conditions.</td>
</tr>
<tr>
<td>Clear User Filters</td>
<td>The <strong>Clear User Filters</strong> button should be used to delete all user-defined <strong>Monitoring Process Filter</strong> conditions.</td>
</tr>
</tbody>
</table>

As for the user filters, those are the user-defined ones. To add a new filter condition, use the **Add Filter** item from the pop-up menu or press the **Add Filter** button on the toolbar.

![Add Monitoring Process Filter](image)

**Pic 2. Configuring a Monitoring Process Filter condition**

When configuring a Monitoring Process Filter condition (Pic 2), you should define a process name to be filtered. All changes performed by the specified process are ignored during a monitoring. If you enable the **Use a regular expression while specifying the filter condition** option, you can define a filter that will exclude changes by all processes which names are matching the expression from monitoring results. See the **How should I correctly specify the filter condition?** section of this document for a detailed description of possible conditions.

To edit a user-defined Monitoring Process Filter select it in the filters table and choose the **Edit** menu item from the pop-up menu or press the **Edit** button on the toolbar. The filter condition editing process is similar to the above-stated creation process. To delete the user-defined conditions that are no longer needed, you can select those conditions in the filters table and choose the **Delete** menu item from the pop-up menu or press the **Delete** button on the toolbar. It is also possible to delete all user-defined conditions using the **Clear User Filters** menu item and the corresponding button on the toolbar.

Along with the generic process monitoring filters, it is proposed to define if the changes performed by specific system modules should be filtered or not. Currently, you can choose if the activity of a default browser should be taken into account during the monitoring process using the **Filter default browser** option.
Uninstall File Filters Page

The Uninstall File Filters allow you to prevent removing system-critical files while uninstalling a generated deployment package. All changes to files and folders in the filtered folder that are captured by the Live Monitoring technology are treated as permanent. To configure the Uninstall File Filters open the program preferences using the Preferences button from the Application Menu and click the Uninstall File Filters link on the navigation bar on the left of the Preferences dialog within the Filters group Pic 1.

![Pic 1. Configuring Uninstall File Filters](image)

The filters are divided into two categories; those are User Filters and System Filters. The system filters are predefined ones and cannot be deleted, but can be disabled, if required, using the Disable Selected item from the pop-up menu. You can then re-enable any filter using the Enable Selected menu item. To enable/disable a filter condition and to check if it is enabled, you can use the check boxes located on the left of each row representing the filter condition. It is also possible to refuse from filtering files and folders from uninstall by disabling the Use the following filter conditions option.

Toolbar Overview

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Filter</td>
<td>The Add Filter button should be used to add a new user-defined Uninstall File Filter condition.</td>
</tr>
<tr>
<td>Edit</td>
<td>The Edit button allows you to change the selected user-defined Uninstall File Filter condition.</td>
</tr>
</tbody>
</table>
Delete

The **Delete** button allows you to delete the selected user-defined **Uninstall File Filter** conditions.

Clear User Filters

The **Clear User Filters** button should be used to delete all user-defined **Uninstall File Filter** conditions.

As for the user filters, those are the user-defined ones. To add a new filter condition, use the **Add Filter** item from the pop-up menu or press the **Add Filter** button on the toolbar.

When configuring an **Uninstall File Filter** condition **Pic 2**, you should provide a file system path to be treated as persistent. No files and folders in the specified path, including the one represented with this path, will be deleted while uninstalling a generated deployment package. You can provide the path manually or select it through the file picker using the **...** button built into the edit box. It is possible to use the system folder definition placeholders while specifying the filter condition, these placeholders are automatically replaced with the corresponding file system paths during a deployment package uninstall process. See the **System Folder Definition Placeholders** section of this document for the list of available placeholders.

Each system folder definition should be proceeded with `{$` and succeeded with `}$`, e. g. `{$(SystemFolder)}`.

The **...** and **...** buttons built into the edit box can help you with replacing specific paths to their system folder definitions and vice versa. If you enable the **Use a regular expression while specifying the filter condition** option, you can define a filter that will treat as permanent all paths matching the expression. See the **How should I correctly specify the filter condition?** section of this document for a detailed description of possible conditions.

To edit a user-defined Uninstall File Filter select it in the filters table and choose the **Edit** menu item from the pop-up menu or press the **Edit** button on the toolbar. The filter condition editing process is similar to the above-stated creation process. To delete the user-defined conditions that are no longer needed, you can select those conditions in the filters table and choose the **Delete** menu item from the pop-up menu or press the **Delete** button on the toolbar. It is also possible to delete all user-defined conditions using the **Clear User Filters** menu item and the corresponding button on the toolbar.
Uninstall Registry Filters Page

The Uninstall Registry Filters allow you to prevent removing system-critical registry keys and values while uninstalling a generated deployment package. All changes to values and sub-keys in the filtered key that are captured by the Live Monitoring technology are treated as permanent. To configure the Uninstall Registry Filters open the program preferences using the Preferences button from the Application Menu and click the Uninstall Registry Filters link on the navigation bar on the left of the Preferences dialog within the Filters group (Pic. 1).

Pic 1. Configuring Uninstall Registry Filters

The filters are divided into two categories; those are User Filters and System Filters. The system filters are predefined ones and cannot be deleted, but can be disabled, if required, using the Disable Selected item from the pop-up menu. You can then re-enable any filter using the Enable Selected menu item. To enable/disable a filter condition and to check if it is enabled, you can use the check boxes located on the left of each row representing the filter condition. It is also possible to refuse from filtering registry keys from uninstall by disabling the Use the following filter conditions option.

Toolbar Overview

<table>
<thead>
<tr>
<th>Add Filter</th>
<th>The Add Filter button should be used to add a new user-defined Uninstall Registry Filter condition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>The Edit button allows you to change the selected user-defined Uninstall Registry Filter condition.</td>
</tr>
</tbody>
</table>
Delete

The **Delete** button allows you to delete the selected user-defined **Uninstall Registry Filter** conditions.

Clear User Filters

The **Clear User Filters** button should be used to delete all user-defined **Uninstall Registry Filter** conditions.

As for the user filters, those are the user-defined ones. To add a new filter condition, use the **Add Filter** item from the pop-up menu or press the **Add Filter** button on the toolbar.

![Pic 2. Configuring an Uninstall Registry Filter condition](image)

When configuring an Uninstall Registry Filter condition **Pic 2**, you should define a registry key to be treated as persistent. No keys and values from the specified key, including the key itself, will be deleted while uninstalling a generated deployment package. To define a registry key you firstly choose the root key from the drop down list and then type in the path to the key in the text edit. If you enable the **Use a regular expression while specifying the filter condition** option, you can define a filter that will treat as permanent all keys matching the expression. See the **How should I correctly specify the filter condition?** section of this document for a detailed description of possible conditions.

To edit a user-defined Uninstall Registry Filter select it in the filters table and choose the **Edit** menu item from the pop-up menu or press the **Edit** button on the toolbar. The filter condition editing process is similar to the above-stated creation process. To delete the user-defined conditions that are no longer needed, you can select those conditions in the filters table and choose the **Delete** menu item from the pop-up menu or press the **Delete** button on the toolbar. It is also possible to delete all user-defined conditions using the **Clear User Filters** menu item and the corresponding button on the toolbar.

### Miscellaneous Part

The **Miscellaneous** part of the program preferences should be used to configure the common MSI Package Builder options, such as automatic update settings, layouts configuration, the proxy settings to be used to connect to the Internet, etc. To open the **Preferences** dialog, click the **Preferences** button available from the **Application Menu**. Configure the available settings to best suit your needs.

### What's Inside
General Settings Page

MSI Package Builder can automatically check for updates for you to always have the latest version of the program. You can configure this feature from the General Settings preference page. To open this page, click the Preferences button from the Application Menu and select the General Settings link in the navigation bar on the left in the Preferences dialog within the Miscellaneous group.  

Pic 1. Configuring general settings

MSI Package Builder can check for updates automatically every day or once a week. To enable an automatic checking for updates, check the Automatically check for updates option and choose the checking frequency between Daily and Weekly. You can also define if the program should check for major updates by changing the Automatically check for major updates option value.

If you use a proxy server to connect to the Internet and the required proxy settings are not provided, an automatic check for updates will not take place.

The application may display warnings and hints, and you can choose if you would like each one to be shown again. On this page, you can reset all the remembered decisions to the defaults so that all the warnings and hints are shown. Use the Reset All Remembered Decisions button to this purpose.
Layouts Page

Layout management technology allows you to configure MSI Package Builder user interface, save it any time you want and then select between saved layouts to change the appearance without reconfiguring windows positions, sizes, etc.

The **Layouts** preference page [Pic 1] is designed to help you with windows layout management. To access this page click **Preferences** button from the **Application Menu** and select the appropriate link in the navigation bar on the left of the **Preferences** dialog.

![Preferences dialog]

**Pic 1. Managing Layouts**

On the **Layouts** preference page [Pic 1] you can create new layout, modify existing one or choose from already saved layouts. Currently active layout is colored gray and cannot be either renamed or deleted.
Proxy Settings Page

MSI Package Builder requires an Internet connection to support the Live Update and Feedback features. Therefore, if a proxy server has to be used to connect to the Internet, it should be configured on the Proxy Settings preference page [Pic 1]. To access this page, click the Preferences button from the Application Menu and select the appropriate link in the navigation bar on the left in the Preferences dialog within the Miscellaneous group.

On this page [Pic 1], you may choose among three variants of the proxy configuration to be used by the program. If Auto-detect proxy settings for this network is chosen, the program uses the settings predefined in the Internet Explorer. If MSI Package Builder does not have to use a proxy server to connect to the Internet, the Direct connection to the Internet option should be chosen. The Manual proxy configuration option allows you to provide the proxy server address and port manually.

Both for the automatic detection and manual configuration, it is possible to specify if the proxy server requires authentication and what credentials should be used to connect to the proxy server.
Chapter 9: Evaluation of the Program

EMCO MSI Package Builder Enterprise is a shareware product but you can use it as long as the evaluation period does not expire to get a closer introduction to its main features. To activate the program you should enter the License Code EMCO Software provides you with after you have purchased the license for using EMCO MSI Package Builder Enterprise. This chapter will cover the particularities of the evaluation mode, tell you how and where you can get the license code and how you can request the extended evaluation. Read this chapter carefully to face no difficulties during the EMCO MSI Package Builder Enterprise evaluation.
Evaluation Wizard

As long as the EMCO MSI Package Builder Enterprise is not activated on each program startup the **Evaluation Wizard** [Pic 1] is displayed on the screen, showing you the information about the evaluation process and providing with quick links for the program activation, purchase and extended evaluation request.

![EMCO MSI Package Builder Enterprise Evaluation Wizard](image)

Pic 1. The EMCO MSI Package Builder Enterprise Evaluation Wizard welcome page
The welcome page of the Evaluation Wizard allows you to choose between four options to continue. Those are Evaluate, Enter License Code, Request extended evaluation and Buy now. Optionally you can press Exit button to close the program. In this section we will help you to choose the option that will best fit your needs.

The Evaluate option shows you the time left until the evaluation period expires. You should choose this option to continue the evaluation process – the wizard will be closed and you can start working with MSI Package Builder. You can use the program as long as the evaluation period does not expire to get a closer introduction to its main features.

It you have already purchased the license for using the EMCO MSI Package Builder Enterprise you should choose the Enter License Code option to activate the program. This options also should be chosen if the extended evaluation request has been approved by EMCO Software and you have been provided with the Extended License. If you are having problems with finding the License Code refer to the Where can I get my License Code? section of this document. After choosing the Enter License Code option the program activation page is displayed on the screen Pic 2.

Pic 2. Activating EMCO MSI Package Builder Enterprise
To activate EMCO MSI Package Builder Enterprise copy and paste the License Code to the input field on this page and press **Activate** - the program will be restarted to activate.

If the evaluation period has expired and you are not sure you have fully introduced yourself to EMCO MSI Package Builder Enterprise main features you can once request the extended evaluation. As soon as the request is processed by EMCO Software you are provided with the Extended License to prolong the evaluation period. To request the Extended License you should choose the **Request extended evaluation** option. After choosing this option the request form will appear on the screen.

![Pic 3](image)

**Pic 3. Requesting an extended evaluation**

To request the Extended License fill the fields on the request form regarding the recommendations and press the **Send Request** button.

The **Evaluation Wizard** also provides you with the short cut action that allows you to visit EMCO Software web store. To use this feature choose the **Buy Now** option. The on-line ordering process is fast, easy, and fully secure.
Where can I get my License Code?

After you have purchased the license for using EMCO MSI Package Builder Enterprise our experts will generate the License Code and send it to you via e-mail to the address you have specified during the purchasing process. You are supposed to receive two e-mail messages – one with the License Code written in the message body and one with the attached text file (license.txt), containing the license. It is your choice to use any message because both License Codes are identical.

To activate the program the License Code received via e-mail should be copied and pasted to the program activation form. This form can be reached using the Enter License Code button from the Information group on the Ribbon bar or by choosing the appropriate option in the Evaluation Wizard.

To activate MSI Package Builder, copy and paste the License Code to the input field and press Activate - the program will be restarted to activate.
How should I formulate the Extended License request?

The Extended License is used to prolong the evaluation period to get a closer look at EMCO MSI Package Builder Enterprise. This feature can be reached by choosing the appropriate option in the Evaluation Wizard. Here we'll give you the recommendations on filling the Request extended evaluation form Pic 1.

Pic 1. The Request Extended Evaluation form

In the Request extended evaluation form you should obligatory specify your name in the Name field, company name in the Company field and your e-mail address in the E-mail field.

Though the Comment field is optional it is strongly recommended to use this field for providing EMCO Software with the reason of requesting the Extended License. Please notice that EMCO Software reserves the right to decline the request without providing a requester with any explanations.

If the extended license request is approved by EMCO Software experts you'll receive the License Code to the e-mail address specified.
Chapter 10: Program Updates

EMCO Software cares for versatile needs of the users of EMCO programs and fully understands their wish to have the most up-to-date software installed on their PCs. That is why we provide you with an easy update feature. You do not need to browse the Internet again and again to find out if any updates are available – MSI Package Builder will do this work for you. Checking for updates can be performed both manually and automatically. This chapter describes the Live Update process for the current major version of the program and the Major Update feature which allows you to get a brand new version of MSI Package Builder quickly and easily.
Live Update

MSI Package Builder can be easily updated with just a few clicks. The update process is performed via an Internet connection using preconfigured proxy settings.

Check for Updates

The Check for Updates button from the Update Ribbon group should be used to check for new versions of MSI Package Builder.

MSI Package Builder can check for updates automatically. You can configure the program behavior regarding the automatic check for updates on the General Settings preference page.

To check for updates, click the Check for Updates button from the Application Menu or from the Update group of the Program Ribbon page. MSI Package Builder will check if any updates are available and if so, the Live Update Wizard will appear on the screen.

The Live Update Wizard will introduce you to the changes made in the newer version and guide you through the whole updating process while showing the detailed download progress. When the download is finished, the program will be restarted to perform the actual update.
Major Update

Along with the Live Update feature, MSI Package Builder comes with a built-in function of automatic checking for Major Updates. The Major Update is an update to a brand-new version of MSI Package Builder that includes a number of significant changes.

You can install this version alongside the version you are using now. It will import the settings and data from your current version, so that you won't need to configure the new version in the same way you configured the one you are using at the moment. Such parallel installation allows you to inspect the new version and compare it with the previous one in your environment with your data and settings.

The Major Update is installed alongside the version you currently use. The existing version is not automatically uninstalled from your PC, and you can continue using the program version you are accustomed to while having a look at the brand new one.

If the program detects availability of a Major Update, the Major Update Wizard will appear on the screen.

The Major Update Wizard will introduce you to the features available in the brand new version of MSI Package Builder and guide you through the update process. The message displayed at the bottom of the welcome page will let you know if the current License allows you to install and use the Major Update for free. When the download is finished, the new version installation will be run automatically.
Chapter 11: Main Program Actions

The main program actions are all gathered on the Ribbon bar and are grouped by the functions performed into pages. There are static pages, that are always displayed, and categories, containing contextual pages, that are displayed only when a specific context is active. As for the static pages, they are the following: Home, Project and Program. These static pages are filled up with the File System, Registry, Environment Variables, Services, Assemblies, Drivers, Printers, Custom Actions and Wrapped Packages pages.

This chapter is to describe the actions available on all the pages and can be used as a glossary while working with MSI Package Builder.

Home Ribbon Page

The Home Ribbon page contains the actions for repackaging, projects management and deployment package creation. This section describes each group from this page and all available actions.

Builder Ribbon Group

The Builder Ribbon group is used to provide access to wizards specially designed to simplify the installation repackaging and projects management workflow.

<table>
<thead>
<tr>
<th>Repackage Installation</th>
<th>The Repackage Installation button from the Builder group on the Home Ribbon page should be used to open the Repackage Installation wizard, that is designed to help you with the repackaging process.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Setup</td>
<td>The Project Setup button from the Builder group on the Home Ribbon page opens the project setup wizard, that can be used as an entry point both for managing MSI Package Builder projects and creating MSI packages on the basis of those projects.</td>
</tr>
<tr>
<td>Create MSI Package</td>
<td>The Create MSI Package button from the Builder group on the Home Ribbon page should be used to generate an MSI package based on the selected project.</td>
</tr>
</tbody>
</table>

Project Management Ribbon Group

The Project Management Ribbon group contains the actions on managing the projects storage and a set of projects available in the Projects view.

<table>
<thead>
<tr>
<th>New</th>
<th>The New button from the Project Management group on the Home Ribbon page should be used to create a new project with a single empty package in it.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>The Open button from the Project Management group on the Home Ribbon page allows you to open projects available in the projects storage.</td>
</tr>
<tr>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Close</td>
<td>The Close button from the Project Management group on the Home Ribbon page should be used to close the selected projects, thus remove them from the Projects view.</td>
</tr>
<tr>
<td>Close All</td>
<td>The Close All button from the Project Management group on the Home ribbon page allows you to close all currently opened projects, thus remove them from the Projects view.</td>
</tr>
<tr>
<td>Rename</td>
<td>The Rename button from the Project Management group on the Home Ribbon page should be used to provide a new name for the selected project.</td>
</tr>
<tr>
<td>Delete</td>
<td>The Delete button from the Project Management group on the Home Ribbon page allows you to delete the selected projects from the projects storage.</td>
</tr>
<tr>
<td>Import Package</td>
<td>The Import Package button from the Project Management group on the Home Ribbon page should be used to create a new MSI Package Builder project based on a changes performed by an existing deployment package.</td>
</tr>
</tbody>
</table>

**Capture Ribbon Group**

The Capture Ribbon group should be used to control the manually initiated advanced monitoring of changes to create a project for further generation of deployment packages based on those changes.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Monitoring</td>
<td>The Start Monitoring button from the Capture group on the Home Ribbon page should be used to start a new session for tracking changes performed to the system.</td>
</tr>
<tr>
<td>Stop Monitoring</td>
<td>The Stop Monitoring button from the Capture group on the Home Ribbon page should be used to stop the currently running monitoring session.</td>
</tr>
</tbody>
</table>

**Project Ribbon Page**

The Project Ribbon page contains the actions for creating new project structure units and preparing a project. This section describes each group from this page and all available actions.

**New Ribbon Group**

The New Ribbon group is used to create new items within a project.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Folder</td>
<td>The System Folder button from the New group on the Project Ribbon page should be used to add a new system folder object to the currently configured project.</td>
</tr>
</tbody>
</table>
Drive
The Drive button from the New group on the Project Ribbon page should be used to add a new logical drive to the currently configured project.

Root Key
The Root Key button from the New group on the Project Ribbon page should be used to add a new root key to the currently configured project.

Environment Variable
The Environment Variable button from the New group on the Project Ribbon page should be used to add a new environment variable modification to the currently configured project.

Service
The Service button from the New group on the Project Ribbon page should be used to add a new service modification to the currently configured project.

Win32 Assembly
The Win32 Assembly button from the New group on the Project Ribbon page should be used to add a new Win32 assembly to the currently configured project.

.NET Assembly
The .NET Assembly button from the New group on the Project Ribbon page should be used to add a new .NET assembly to the currently configured project.

Basic Driver
The Basic Driver button from the New group on the Project Ribbon page should be used to add a new basic driver deployment action to the currently configured project.

Driver Package
The Driver Package button from the New group on the Project Ribbon page should be used to add a new driver package to the currently configured project to be either installed or pre-installed by a resulting deployment package.

DIFx Driver Package
The DIFx Driver Package button from the New group on the Project Ribbon page should be used to add a new Driver Installation Framework package to the currently configured project to be either installed or pre-installed by a resulting deployment package.

Printer Entries
The Printer Entries drop-down button from the New group on the Project Ribbon page can be used to create modifications to the printing system to be performed by a resulting deployment package.

Custom Actions
The Custom Actions drop-down button from the New group on the Project Ribbon page should be used to create a new action of a specific type and add it to the selected project.

Wrapped Package
The Wrapped Package button from the New group on the Project Ribbon page should be used to create a new wrapped package and add it to the selected project.
Links Ribbon Group

The **Links** Ribbon group contains the actions for preparing projects for a deployment package creation.

- **Prepare**
  - The **Prepare** button from the **Links** group on the **Project** Ribbon page should be used to prepare the selected projects.

- **Missing Links**
  - The **Missing Links** drop-down button from the **Links** group on the **Project** Ribbon page allows you to choose the method for resolving the problems with missing files in the selected projects. You can either remove those files, or try to check if they became available.

Import Ribbon Group

The **Import** Ribbon group should be used to import file system or registry changes into projects.

- **File System**
  - The **File System** drop-down button from the **Import** group on the **Project** Ribbon page should be used to add files and folders from the file system to the selected project.

- **Registry**
  - The **Registry** button from the **Import** group on the **Project** Ribbon page allows you to add the changes defined in the specific registration entries (.reg) file to the selected project.

System Folders Ribbon Group

The **System Folders** Ribbon group contains the actions for rolling and unrolling the system folders for a project.

- **Roll All**
  - The **Roll All** button from the **System Folders** group on the **Project** Ribbon page should be used to replace all system folders with their definitions in the selected projects.

- **Unroll All**
  - The **Unroll All** button from the **System Folders** group on the **Project** Ribbon page allows you to expand the system folder definitions to their absolute local paths in the selected projects.
**View Ribbon Page**

The View Ribbon page is used to control the program representation, such as the currently applied skin, visible views and their layout.

**Layout Ribbon Group**

The Layout Ribbon group should be used for a workspace layout management.

- **Layout Chooser**
  
  The Layout Chooser is represented by a drop-down list filled with available window layouts. You can choose one of the existing layouts to apply it or input a name for a new layout to be saved.

- **Reset Workspace**
  
  The Reset Workspace action should be used to restore the default windows’, views’ and dock panels’ layout.

**Show Ribbon Group**

The Show Ribbon group should be used for managing the currently visible MSI Package Builder views.

<table>
<thead>
<tr>
<th></th>
<th>Projects</th>
<th>Log</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shows or hides the Projects view. The Projects view is designed to display a set of currently opened MSI Package Builder projects.</td>
<td>Shows or hides the Log view. The Log view shows information on the events taking place during the program execution.</td>
<td>Shows or hides the Operations view. The Operations view shows detailed progress of each operation currently performed and allows canceling some or all running operations.</td>
</tr>
</tbody>
</table>

**Skins Ribbon Group**

The Skins Ribbon group provides you with an option of quick changing of the program skins.

- **Skin Chooser**
  
  The Skin Chooser is a drop-down button that should be used to select the skin from the available skins to be applied to the program. The currently active skin is the checked one.
Program Ribbon Page

The Program Ribbon page contains service actions that are not connected to the business area of the program.

Clipboard Ribbon Group

The Clipboard Ribbon group contains the actions to copy/move objects to the clipboard and paste data from the clipboard.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paste</td>
<td>The Paste button should be used to add the objects that are currently available from the clipboard to a selected location in the currently focused view.</td>
</tr>
<tr>
<td>Cut</td>
<td>The Cut button should be used to put the selected objects from the currently focused view to the clipboard and remove them from the source after pasting them to the destination.</td>
</tr>
<tr>
<td>Copy</td>
<td>The Copy button allows you to put the selected objects from the currently focused view to the clipboard to be copied to the destination when pasting.</td>
</tr>
</tbody>
</table>

Update Ribbon Group

The Update Ribbon group is responsible for the product updates function. The glyph of this group opens the General Settings preference page enabling you to configure the auto-update options.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for Updates</td>
<td>The Check for Updates button should be used to check for new versions of MSI Package Builder.</td>
</tr>
</tbody>
</table>

Feedback Ribbon Group

You can use the Feedback Ribbon group to send EMCO Software a request for a new feature you are missing in MSI Package Builder or to report on problems you faced while working with the program.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suggest a Feature</td>
<td>The Suggest a Feature button should be used to suggest a functionality you would like to see in the next versions of MSI Package Builder.</td>
</tr>
<tr>
<td>Report a Problem</td>
<td>The Report a Problem button should be used for reporting the problem you faced while working with MSI Package Builder.</td>
</tr>
</tbody>
</table>
Information Ribbon Group

The Information Ribbon group has a range of useful actions to manage licensing issues, get help or information you may require.

- **Enter License Code**
  The Enter License Code action should be used to enter the license code you received from EMCO to activate MSI Package Builder.

- **Buy Now**
  The Buy Now button is only visible if MSI Package Builder is not activated. It provides you with a quick access to the EMCO web store.

- **Edition Upgrade**
  The Edition Upgrade provides you with an ability of benefiting from update to a more comprehensive edition of MSI Package Builder with a help of the Edition Upgrade Wizard that will help you choose an appropriate edition and purchase a license for using it.

- **Contact Support**
  The Contact Support button should be used to get efficient technical assistance from EMCO Software support team.

- **EMCO on the Web**
  The EMCO on the Web button provides you with a short-cut to the EMCO website.

- **About**
  It opens the About dialog to review the MSI Package Builder information, licensing information and the End-User License Agreement.

File System Contextual Ribbon Page

The contextual File System Ribbon page is displayed when the File System view is active and contains the actions for file system modifications management. This section describes each group from this page and all available actions.

New Ribbon Group

The New Ribbon group is used to create file system modifications within a project.

- **System Folder**
  The System Folder button from the New group on the contextual File System Ribbon page should be used to add a new system folder object to the currently configured project.

- **Drive**
  The Drive button from the New group on the contextual File System Ribbon page should be used to add a new logical drive to the currently configured project.
## Main Program Actions

### Management Ribbon Group

The **Management** Ribbon group allows you to edit, delete and transfer file system modifications.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edit</strong></td>
<td>The <strong>Edit</strong> button from the <strong>Management</strong> group on the contextual <strong>File System</strong> Ribbon page should be used to change the selected file system modification.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>The <strong>Delete</strong> button from the <strong>Management</strong> group on the contextual <strong>File System</strong> Ribbon page allows you to delete the selected file system modification from the project.</td>
</tr>
<tr>
<td><strong>Copy To</strong></td>
<td>The <strong>Copy To</strong> button from the <strong>Management</strong> group on the contextual <strong>File System</strong> Ribbon page should be used to copy the selected file system modifications to another location.</td>
</tr>
<tr>
<td><strong>Move To</strong></td>
<td>The <strong>Move To</strong> button from the <strong>Management</strong> group on the contextual <strong>File System</strong> Ribbon page allows you to move the selected file system modifications to another location.</td>
</tr>
</tbody>
</table>

### Import Ribbon Group

The **Import** Ribbon group should be used to import file system changes into a project.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Folder</strong></td>
<td>The <strong>Folder</strong> button from the <strong>Import</strong> group on the contextual <strong>File System</strong> Ribbon page allows you to add a folder from the file system with or without its content to the selected project.</td>
</tr>
<tr>
<td><strong>File</strong></td>
<td>The <strong>File</strong> button from the <strong>Import</strong> group on the contextual <strong>File System</strong> Ribbon page allows you to add a file from the file system to the selected project.</td>
</tr>
</tbody>
</table>
System Folders Ribbon Group

The System Folders Ribbon group contains the actions for rolling and unrolling specific system folders within the file system modifications.

| Roll   | The Roll button from the System Folders group on the contextual File System Ribbon page should be used to replace the selected system folders with their definitions in the selected project. |
| Unroll | The Unroll button from the System Folders group on the contextual File System Ribbon page allows you to expand the selected system folders to their absolute local paths in the selected project. |

Presentation Ribbon Group

The Presentation Ribbon group should be used to switch between tree and flat presentation of file system modifications.

| Tree   | The Tree button from the Presentation group of the contextual File System tab should be used to display the file system modifications in form of a tree, similar to Windows Explorer. |
| Flat   | The Flat button from the Presentation group of the contextual File System tab allows you to display the file system modifications as a raw list, where each element is represented with its full path. |

Registry Contextual Ribbon Page

The contextual Registry Ribbon page is displayed when the Registry view is active and contains the actions for registry modifications management. This section describes each group from this page and all available actions.

New Ribbon Group

The New Ribbon group is used to create registry modifications within a project.

| Root Key | The Root Key button from the New group on the contextual Registry Ribbon page should be used to add a new root key to the currently configured project. |
| Key | The Key button from the New group on the contextual Registry Ribbon page should be used to create a new registry key modification within the selected registry key. |
Value

The **Value** drop-down button from the **New** group on the contextual **Registry** Ribbon page should be used to create a new registry value modification within the selected registry key.

Management Ribbon Group

The **Management** Ribbon group allows you to edit, delete and transfer registry modifications.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edit</strong></td>
<td>The <strong>Edit</strong> button from the <strong>Management</strong> group on the contextual <strong>Registry</strong> Ribbon page should be used to change the selected registry modification.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>The <strong>Delete</strong> button from the <strong>Management</strong> group on the contextual <strong>Registry</strong> Ribbon page allows you to delete the selected registry modification from the selected project.</td>
</tr>
<tr>
<td><strong>Copy To</strong></td>
<td>The <strong>Copy To</strong> button from the <strong>Management</strong> group on the contextual <strong>Registry</strong> Ribbon page should be used to copy the selected registry modifications to another location.</td>
</tr>
<tr>
<td><strong>Move To</strong></td>
<td>The <strong>Move To</strong> button from the <strong>Management</strong> group on the contextual <strong>Registry</strong> Ribbon page allows you to move the selected registry modifications to another location.</td>
</tr>
</tbody>
</table>

Import Ribbon Group

The **Import** Ribbon group should be used to import registry changes into project.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Registry</strong></td>
<td>The <strong>Registry</strong> button from the <strong>Import</strong> group on the contextual <strong>Registry</strong> Ribbon page allows you to add the changes defined in the specific registration entries (.reg) file to the selected project.</td>
</tr>
</tbody>
</table>

Presentation Ribbon Group

The **Presentation** Ribbon group should be used to switch between tree and flat presentation of registry modifications.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tree</strong></td>
<td>The <strong>Tree</strong> button from the <strong>Presentation</strong> group of the contextual <strong>Registry</strong> tab should be used to display the registry modifications in form of a tree, similar to <strong>Registry Editor</strong>.</td>
</tr>
<tr>
<td><strong>Flat</strong></td>
<td>The <strong>Flat</strong> button from the <strong>Presentation</strong> group of the contextual <strong>Registry</strong> tab allows you to display the registry modifications as a raw list, where each element is represented with its full path.</td>
</tr>
</tbody>
</table>
Environment Variables Contextual Ribbon Page

The contextual Environment Variables Ribbon page is displayed when the Environment Variables view is active and contains the actions for environment variables modifications management. This section describes each group from this page and all available actions.

New Ribbon Group

The New Ribbon group is used to create environment variables modifications within a project.

Environment Variable

The Environment Variable button from the New group on the contextual Environment Variables Ribbon page should be used to add a new environment variable modification to the currently configured project.

Management Ribbon Group

The Management Ribbon group allows you to edit, delete and transfer environment variables modifications.

Edit

The Edit button from the Management group on the contextual Environment Variables Ribbon page should be used to change the selected environment variables modification.

Delete

The Delete button from the Management group on the contextual Environment Variables Ribbon page allows you to delete the selected environment variables modifications from the selected project.

Copy To

The Copy To button from the Management group on the contextual Environment Variables Ribbon page should be used to copy the selected environment variables modifications to another location.

Move To

The Move To button from the Management group on the contextual Environment Variables Ribbon page allows you to move the selected environment variables modifications to another location.
Services Contextual Ribbon Page

The contextual Services Ribbon page is displayed when the Services view is active and contains the actions for services modifications management. This section describes each group from this page and all available actions.

New Ribbon Group

The New Ribbon group is used to create services modifications within a project.

Service

The Service button from the New group on the contextual Services Ribbon page should be used to add a new service modification to the currently configured project.

Management Ribbon Group

The Management Ribbon group allows you to edit, delete and transfer services modifications.

Edit

The Edit button from the Management group on the contextual Services Ribbon page should be used to change the selected services modification.

Delete

The Delete button from the Management group on the contextual Services Ribbon page allows you to delete the selected services modifications from the selected project.

Copy To

The Copy To button from the Management group on the contextual Services Ribbon page should be used to copy the selected services modifications to another location.

Move To

The Move To button from the Management group on the contextual Services Ribbon page allows you to move the selected services modifications to another location.

Assemblies Contextual Ribbon Page

The contextual Assemblies Ribbon page is displayed when the Assemblies view is active and contains actions for side-by-side assemblies management. This section describes each group from this page and all available actions.

New Ribbon Group

The New Ribbon group is used to create side-by-side assemblies to be included into a project.

Win32 Assembly

The Win32 Assembly button from the New group on the contextual Assemblies Ribbon page should be used to add a new Win32 assembly to the currently configured project.
Management Ribbon Group

The Management Ribbon group allows you to edit, delete and transfer side-by-side assemblies.

- **Edit**
  The Edit button from the Management group on the contextual Assemblies Ribbon page should be used to change the selected assembly.

- **Delete**
  The Delete button from the Management group on the contextual Assemblies Ribbon page allows you to delete the selected assemblies from the selected project.

- **Copy To**
  The Copy To button from the Management group on the contextual Assemblies Ribbon page should be used to copy the selected assemblies to another project.

- **Move To**
  The Move To button from the Management group on the contextual Assemblies Ribbon page allows you to move the selected assemblies to another project.

Drivers Contextual Ribbon Page

The contextual Drivers Ribbon page is displayed when the Drivers view is active and contains actions for driver modifications management. This section describes each group from this page and all available actions.

New Ribbon Group

The New Ribbon group is used to create a new basic driver or a driver package deployment action within a project.

- **Basic Driver**
  The Basic Driver button from the New group on the contextual Drivers Ribbon page should be used to add a new basic driver deployment action to the currently configured project.

- **Driver Package**
  The Driver Package button from the New group on the contextual Drivers Ribbon page should be used to add a new driver package to the currently configured project to be either installed or pre-installed by a resulting deployment package.
Management Ribbon Group

The Management Ribbon group allows you to edit, delete and transfer driver modifications.

**Edit**

The Edit button from the Management group on the contextual Drivers Ribbon page should be used to change the selected driver deployment action.

**Delete**

The Delete button from the Management group on the contextual Drivers Ribbon page allows you to delete the selected driver deployment actions from the selected project.

**Copy To**

The Copy To button from the Management group on the contextual Drivers Ribbon page should be used to copy the selected driver deployment actions to another location.

**Move To**

The Move To button from the Management group on the contextual Drivers Ribbon page allows you to move the selected driver deployment actions to another location.

Order Ribbon Group

The Order Ribbon group is used to organize the driver deployment order.

**Move Up**

The Move Up button from the Order group on the contextual Drivers ribbon page should be used to move the selected drivers up the deployment order.

**Move Down**

The Move Down button from the Order group on the contextual Drivers ribbon page should be used to move the selected drivers down the deployment order.

DIFx Driver Package

The DIFx Driver Package button from the New group on the Drivers contextual Ribbon page should be used to add a new Driver Installation Framework package to the currently configured project to be either installed or pre-installed by a resulting deployment package.
Printers Contextual Ribbon Page

The **Printers** contextual Ribbon page is displayed when the **Printers** view is active. It contains the actions for managing the printing system configuration. This section describes each group from this page and all the available actions.

**New Ribbon Group**

The **New** Ribbon group is used to create a new entry for configuring the printing system.

<table>
<thead>
<tr>
<th>Button</th>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Printer" /></td>
<td>Printer</td>
<td>The <strong>Printer</strong> button from the <strong>New</strong> group on the <strong>Printers</strong> contextual Ribbon page should be used to add a new deployment action for creating, modifying or deleting a printer during the package deployment.</td>
</tr>
<tr>
<td><img src="image2" alt="Printer Driver" /></td>
<td>Printer Driver</td>
<td>The <strong>Printer Driver</strong> button from the <strong>New</strong> group on the <strong>Printers</strong> contextual Ribbon page should be used to add a new deployment action for installing or removing a printer driver during the package deployment.</td>
</tr>
<tr>
<td><img src="image3" alt="Print Processor" /></td>
<td>Print Processor</td>
<td>The <strong>Print Processor</strong> button from the <strong>New</strong> group on the <strong>Printers</strong> contextual Ribbon page should be used to add a new deployment action for adding or removing a print processor during the package deployment.</td>
</tr>
<tr>
<td><img src="image4" alt="Print Monitor" /></td>
<td>Print Monitor</td>
<td>The <strong>Print Monitor</strong> button from the <strong>New</strong> group on the <strong>Printers</strong> contextual Ribbon page should be used to add a new deployment action for adding or removing a print monitor (either a language monitor or a port monitor) during the package deployment.</td>
</tr>
<tr>
<td><img src="image5" alt="Printer Port" /></td>
<td>Printer Port</td>
<td>The <strong>Printer Port</strong> button from the <strong>New</strong> group on the <strong>Printers</strong> contextual Ribbon page should be used to add a new deployment action for creating, modifying or deleting a printer port during the package deployment.</td>
</tr>
</tbody>
</table>

**Management Ribbon Group**

The **Management** Ribbon group allows you to edit, delete and transfer printing system configuration entries.

<table>
<thead>
<tr>
<th>Button</th>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image6" alt="Edit" /></td>
<td>Edit</td>
<td>The <strong>Edit</strong> button from the <strong>Management</strong> group on the <strong>Printers</strong> contextual Ribbon page should be used to change the selected printing system configuration action.</td>
</tr>
<tr>
<td><img src="image7" alt="Delete" /></td>
<td>Delete</td>
<td>The <strong>Delete</strong> button from the <strong>Management</strong> group on the <strong>Printers</strong> contextual Ribbon page allows you to delete the selected printing system configuration actions from the selected project.</td>
</tr>
</tbody>
</table>
Copy To
The Copy To button from the Management group on the Printers contextual Ribbon page should be used to copy the selected printing system configuration actions to another location.

Move To
The Move To button from the Management group on the Printers contextual Ribbon page allows you to move the selected printing system configuration actions to another location.

Custom Actions Contextual Ribbon Page
The contextual Custom Actions Ribbon page is displayed when the Custom Actions view is active and contains the items that used for custom actions management. This section describes each group from this page and all available actions.

New Ribbon Group
The New Ribbon group is used to create a new action for a specific project.

- **Pre & Post Action**
The Pre & Post Action button from the New group on the contextual Custom Actions Ribbon page should be used to create a new generic action and add it to the selected project.

- **SAM Licenses Installation**
The SAM Licenses Installation button from the New group on the Custom Actions contextual Ribbon page should be used to create a new action for installing Software Assets Management (SAM) licenses to the Software Licensing Service (SLS) and to add it to the selected project.

- **Pin Application Action**
The Pin Application Action button from the New group on the Custom Actions contextual Ribbon page allows you to create a new action to pin any application to or unpin it from the Start Menu and/or the Task Bar.

Management Ribbon Group
The Management Ribbon group allows you to edit, delete and transfer custom actions.

- **Edit**
The Edit button from the Management group on the contextual Custom Actions Ribbon page should be used to change the selected action configuration.

- **Delete**
The Delete button from the Management group on the contextual Custom Actions Ribbon page allows you to delete the selected actions from the selected project.
Order Ribbon Group

The **Order** Ribbon group is used to organize the custom actions execution order.

- **Move Up**
  The **Move Up** button from the **Order** group on the contextual **Custom Actions** Ribbon page should be used to move the selected user actions up the execution order.

- **Move Down**
  The **Move Down** button from the **Order** group on the contextual **Custom Actions** Ribbon page should be used to move the selected user actions down the execution order.

Wrapped Packages Contextual Ribbon Page

The contextual **Wrapped Packages** Ribbon page is displayed when the **Wrapped Packages** view is active and contains the actions for wrapped packages management. This section describes each group from this page and all available actions.

New Ribbon Group

The **New** Ribbon group is used to create wrapped packages to be included into a project.

- **Wrapped Package**
  The **Wrapped Package** button from the **New** group on the contextual **Wrapped Packages** Ribbon page should be used to create a new wrapped package and add it to the selected project.

Management Ribbon Group

The **Management** Ribbon group allows you to edit, delete and transfer wrapped packages.

- **Edit**
  The **Edit** button from the **Management** group on the contextual **Wrapped Packages** Ribbon page should be used to change the selected wrapped package configuration.

- **Delete**
  The **Delete** button from the **Management** group on the contextual **Wrapped Packages** Ribbon page allows you to delete the selected wrapped packages from the selected project.
Copy To

The **Copy To** button from the **Management** group on the contextual **Wrapped Packages** Ribbon page should be used to copy the selected wrapped packages to another project.

Move To

The **Move To** button from the **Management** group on the contextual **Wrapped Packages** Ribbon page allows you to move the selected wrapped packages to another project.

Order Ribbon Group

The **Order** Ribbon group is used to organize the wrapped packages deployment order.

**Move Up**

The **Move Up** button from the **Order** group on the contextual **Wrapped Packages** ribbon page should be used to move the selected packages up the deployment order.

**Move Down**

The **Move Down** button from the **Order** group on the contextual **Wrapped Packages** ribbon page should be used to move the selected packages down the deployment order.
Chapter 12: Requirements

Please carefully read and follow all requirements, listed here, or you may not be able to successfully use the product. You can contact our support if you experience a problem during the product use.

System Requirements

Computer running MSI Package Builder must meet the following requirements:

Minimum Hardware Requirements

- Intel Core 2 Duo Processor or equivalent
- 2 GB of RAM
- 250 MB of free disk space

Recommended Hardware Requirements

- 6th Gen Intel Core Processor or equivalent
- 4 GB of RAM
- 500 MB of free disk space

Supported Platforms

Windows XP (with SP3 or later), Windows XP x64 (with SP2 or later), Windows 2003 (with SP2 or later), Windows 2003 x64 (with SP2 or later), Windows 2003 R2, Windows 2003 R2 x64, Windows Vista, Windows Vista x64, Windows 2008, Windows 2008 x64, Windows 2008 R2, Windows 7, Windows 7 x64, Windows 8, Windows 8 x64, Windows 2012, Windows 8.1, Windows 8.1 x64, Windows 2012 R2, Windows 10, Windows 10 x64, Windows 2016

Requirements

- Administrative rights on the local computer
- Microsoft .NET Framework 3.5 SP1 or above
Chapter 13: Edition Upgrade

EMCO MSI Package Builder Enterprise comes with a wide range of features but it is not the most comprehensive edition of the program. If you want to get to know about the features available in the advanced editions of MSI Package Builder or purchase an upgrade to a more comprehensive edition the program can help you with this task.

**Edition Upgrade**

The **Edition Upgrade** provides you with an ability of benefiting from update to a more comprehensive edition of MSI Package Builder with a help of the **Edition Upgrade Wizard** that will help you choose an appropriate edition and purchase a license for using it.

The **Edition Upgrade Wizard** [Pic 1] was designed to make the upgrade process easier. This wizard can be reached by clicking an appropriate hyperlink in the About dialog or by using the **Edition Upgrade** button from the Program Ribbon page.

With a help of the **Edition Upgrade Wizard** you can purchase an upgrade to more comprehensive edition of MSI Package Builder with a single click on the **Get more features with an advanced edition of the program** option or introduce yourself to the features available in the other edition of the program using the **Choose the program edition that is best for you** option. This option will open a feature list web page that shows you the detailed comparison matrix of the features available in different MSI Package Builder editions so that you can review all the features of each edition before choosing the one that best fits your needs.
Chapter 14: How can I leave my Feedback?

EMCO Software always takes care of its customers, and your opinion means a lot to us. For this reason, our programs have built-in features for your feedback. You can suggest a feature you want to see in new program versions or report a technical problem you have faced using the program. Specifying your contact information on the feedback forms ensures that you will be informed of any changes with regard to the reported issue, our plans for implementing the suggested feature or fixing the reported bug. Those actions can be found in the Feedback Ribbon group of the Program page.

**Suggest a Feature**

The Suggest a Feature button from the Feedback Ribbon group should be used to suggest a functionality you would like to see in the next versions of MSI Package Builder.

MSI Package Builder comes with a wide range of features, but if you feel some functionality is missing, you can always suggest a new feature to us that you want to see in the program. To suggest a new feature, you should press the Suggest a Feature button from the Feedback Ribbon group. After pressing this button, you will see the Feature Suggestion dialog on the screen where you are offered to enter your contact information and describe your suggestion.

Press **Send** when you are done with filling out the form to send your suggestion.

**Report a Problem**

The Report a Problem button should be used to report a problem you have faced while working with MSI Package Builder.
MSI Package Builder is easy to use and very stable. Nevertheless, if you have faced any difficulty or problem while working with it, you can send us a problem report. To send such a report, you should press the **Report a Problem** button from the **Feedback** Ribbon group. When this button is pressed, the **Problem Report** dialog [Pic 2] will appear on the screen where you are offered to enter your contact information and describe your problem.

In the **Environment** field, you can provide us with a description of the specific environment used while working with the program. Press **Send** when you are done with filling out the form to send your report.

Do not hesitate to contact EMCO Software - we are always glad to receive your feedback and are doing our best to satisfy our customers' preferences.
Chapter 15: About EMCO Software

EMCO Software is a leading innovator on the remote administration and management solutions market. Headquartered in Reykjavik, Iceland, we are an international company with offices and operations around the globe. Since the company's founding in 2001, we have been dedicated to providing network administrators with feature-rich, easy-to-use and affordable software and improving productivity of IT departments. Focused on the real-world needs of network professionals, our products are simple to try, simple to use, and simple to maintain while providing the power, scalability and flexibility needed by companies and organizations of all sizes.

Our solutions portfolio includes award-winning products targeted at remote network inventory, remote software deployment, remote desktop access and administration, network protection from viruses and malware, network security and health analyzing, remote power management, and remote automation.

Our Innovations

For a long time, products for remote desktops management, administration and inventoring were positioned on the market as enterprise-level solutions with a prohibitive total cost of ownership. A high price and a significant complexity kept many companies and organizations away from those solutions despite their high demand for tools that can improve productivity of their IT departments. We came onto the market looking to change this situation focusing on real-world needs of network professionals by providing simple but effective products and making them affordable for companies and organizations of all sizes.

Our first releases in 2001 proved that our products - which can be up and running within minutes, with no learning and complex configuration required, to immediately provide remote desktop control over the network - were exactly the solution that network engineers need most. An intuitive interface, valuable highly demanded features and an affordable price were the key factors to our quick market acquiring. After the first big success, our solutions portfolio quickly grew to 35 products that cover various fields of network administration and help automate hundreds of administration tasks. Along with this, there is still a lot of space for improvement, and we plan to further extend our product range to provide best-of-breed solutions.

EMCO Worldwide

More than 20,000 customers in 85 countries around the world rely on our products every day to take the complex routine out of their network management tasks. Our solutions help them to remotely manage as many as 2,3 million desktops and 103,000 servers and save millions of hours of work and multi-million dollar costs in total every year. Our worldwide customer base includes dozens of Fortune 100 and hundreds of Fortune 500 companies, as well as numerous large and small businesses representing all industries, educational and healthcare organizations, charity and non-profit organizations, and government institutions.

As a global company, we have partners all around the world who contribute to our products development, international sales and technical support. We are proud to cooperate with leading distributors and more than 50 resellers from 20 countries who represent our products on local markets.

Company Facts

- Year founded: 2001
- Headquarters: Reykjavik, Iceland
• Status: International corporation
• Expertise: Software network solutions
• Number of products: 35+
• Technology awards: 100+
• Number of customers: 20,000+
• Customers in: 85 countries worldwide
• Fortune 100 customers: 37%
• Fortune 500 customers: 49%
• Operations in: 20 countries worldwide
• Number of sales partners: 50+
Chapter 16: Contact Information

We would be glad to help you with any questions and problems you might have. Here you can find all the information you will need to contact us. To get quick answers regarding support issues or any other services, use corresponding contact information.

Company Requisites

- Name: EMCO ehf.
- Address: 12, Bildshofda St.
- City: Reykjavik
- Country: Iceland
- Postal code: 110
- E-mail: emco@emcosoftware.com
- Site: emcosoftware.com

Sales Questions

For any pre-sales or sales questions, contact us at sales@emcosoftware.com or call us:

- **+1 646 233-1163** Business days 9.00-18.00 GMT (8.00-13.00 EST; 8.00-12.00 CST; 8.00-10.00 PDT).
- **+44 20 3287-7651** Business days 9.00-18.00 GMT.

Other hours: please leave your phone number and message - we will recall you within one business day.

Technical Support

In case of any problems with our products or any technical question related to our products, please visit our Support area to receive FREE technical support.