PACKBENCH

# Software Packaging Workbench

## User Guide PackBench for RayPack 7.0

PackBench is part of RaySuite.

raynet

**Copyright © Raynet GmbH (Germany, Paderborn HRB 3524). All rights reserved.**
**Complete or partial reproduction, adaptation, or translation without prior written permission is prohibited.**

User Guide PackBench for RayPack

Raynet and RayFlow are trademarks or registered trademarks of Raynet GmbH protected by patents in European Union, USA and Australia, other patents pending. Other company names and product names are trademarks of their respective owners and are used to their credit.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Raynet GmbH. Raynet GmbH assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. All names and data used in examples are fictitious unless otherwise noted.

Any type of software or data file can be packaged for software management using packaging tools from Raynet or those publicly purchasable in the market. The resulting package is referred to as a Raynet package. Copyright for any third party software and/or data described in a Raynet package remains the property of the relevant software vendor and/or developer. Raynet GmbH does not accept any liability arising from the distribution and/or use of third party software and/or data described in Raynet packages. Please refer to your Raynet license agreement for complete warranty and liability information.

Raynet GmbH Germany
See our website for locations.

www.raynet.de

# Contents


Introduction ... 7
- Development Roadmap ... 9
- Additional Resources ... 9

System Requirements ... 10
- Hardware Requirements ... 10
- Supported OS ... 10
- Prerequisite Software ... 11

Product installation ... 12
- Installing standalone product ... 12
- Installing server component ... 21
- Installing client component ... 29
- Security and required firewall configuration ... 36
- Product activation ... 39
  - License wizard ... 39
    - Order number ... 44
    - License file ... 48
    - Floating License Server ... 49
    - I do not have a license or order number ... 50
    - I want to take my activation back ... 51

Basic Concepts ... 52
- Application Architecture ... 55
- Standard Objects ... 56
- Quick Start Guide ... 58

Home Screen ... 60
- Dashboard ... 60
- Configuration ... 61
- Library ... 61
- Recent ... 62

About ... 63
- Get Started ... 63
- License and Edition ... 65
  - License File Management ... 66
- Troubleshooting ... 67

Settings ... 69
- Interface ... 69
- Connection ... 70
- Bench ... 71

RayFlow ... 72
Workflow Runs ... 73
Managing Runs ... 73
Starting Runs ... 79
Working with Runs ... 86
Overview ... 87
Summary ... 88
Files ... 89
Custom Variables ... 90
RayFlow ... 91
Generating reports ... 92
Task View ... 92
Sidebar ... 94
Strict Mode and Editing Limitations ... 94
Deleting Runs ... 97
Working with RayFlow ... 98
Introduction ... 98
Signing-in to RayFlow ... 98
Privacy Concerns ... 102
Creating Runs from RayFlow Tasks ... 102
Managing RayFlow Task ... 107
Managing State and Comments ... 107
Uploading Files and Folders ... 109
Downloading Files ... 111
Updating Data Field Values ... 112
Importing Data Fields from RayFlow ... 114
Signing Out from RayFlow ... 116
Configuration ... 117
Configuring Projects ... 117
Creating Projects ... 119
Editing Projects ... 121
Display ... 121
Naming Conventions ... 124
Reports ... 125
RayFlow ... 128
Archiving Projects ... 130
Configuring Workflows ... 132
Creating Workflows ... 135
Editing Workflows ... 138

Display ... 138
Tasks ... 139
Custom Variables ... 149
Advanced ... 153
Cloning Workflows ... 154
Deleting Workflows ... 156
Custom Variables ... 158
Text ... 159
Boolean Value ... 161
Numeric Value ... 162
Date ... 163
Single Option ... 164
Multi Option ... 165
File ... 166
Folder ... 168
Configuring Tools ... 169
Creating Tools ... 172
Editing Tools ... 174
Display ... 174
Command ... 176
Configurations ... 179
Help ... 183
Deleting Tools ... 185
Exporting Tools ... 186
Importing Tools ... 186
Common Dialogs ... 188
Variable Editor ... 188
General ... 189
RaySuite ... 190
Environment ... 191
Project ... 192
Workflow ... 192
Run ... 193
Variables ... 193
Tool ... 194
Direct Value Editor ... 194
Command Line Tools ... 196
Silent Command Line Switches ... 196
NewRun ... 196

Advanced Topics ..... 199
Customizing Reporting Templates ..... 199
Custom Run Status ..... 205
Custom Tool Groups ..... 205
Changing RayFlow Server ..... 206
Advanced Logging Options ..... 207
Configuration Files ..... 209
Client ..... 209
Server ..... 211
Reference and Cheat Sheets ..... 213
PackBench Variables ..... 213
Variable Modifiers ..... 216
Regular Expressions ..... 217
User Interface Shortcuts ..... 220
Help and Support ..... 221
Additional Information ..... 222

# Introduction

PackBench is designed to help teams perform complex, yet uniform procedures over and over again: at highest operational standard, a maximum of status clarity and optimal result quality. Information transparency and procedure efficiency are the imperatives for any functionality within PackBench. Therefore, this tool makes sure to define all required information and tasks for a specific procedure, turning it into a well-defined workflow teams may easily comprehend and execute.

Adding tools and variables to those workflows enables flexible, context defined process performance (called run), applying individual values for general specifications. PackBench does not only support mere confirmation of manual task execution, but at the same time enables workflow designers to integrate tools for automated task execution. By handing over variables to those script or application based tools, PackBench becomes the ideal instrument for neat and secure automation of tasks which are usually time-consuming and error-prone when they are performed manually. Allowing integrated tools to modify specific run properties after their execution drives the process further, based upon standardized real-time data.

Each task of a PackBench workflow can be equipped with detailed instructions on how performance is expected. Workflows may be organized in projects, making sure that each team has a dedicated context to operate in. Defining individual task status values reflects context dependent communication and execution standards, allowing users to take over any run at any time. Read the chapters of this document to learn how PackBench features support your individual workflow performance needs!

# PackBench is part of RaySuite

## Best-Practice Workflow for Enterprise Application Lifecycle Management

RaySuite offers product solutions for the creation, operation and control of individually customized Enterprise Application Lifecycle Management projects in automatically administered environments.

Designed as part of the RaySuite, Raynet's product suite for Enterprise Application Lifecycle Management, PackBench is a packaging process related workflow performance tool. However, due to its flexible design PackBench actually is a multi-purpose application for any kind of standardized, task-based procedure execution. It may be used either way - standalone or as integrated part of RaySuite.

When PackBench is used as part of the RaySuite, it is connected to RayFlow, the workflow management component of the RaySuite, which orchestrates Application Lifecycle Management activities. In this case there are some additional features, such as directly exporting run results to RayFlow, which will not be available in a standalone instance of PackBench. Please refer to the RaySuite website for further information regarding the benefits this product family offers for your daily business tasks.

# Development Roadmap

Upcoming releases will introduce additional new components and features, resulting in boosted productivity, saving time and resources, and overall improvement of the product experience. The development of PackBench is partner and customer oriented and driven; therefore, any ideas or suggestions for further development or to meet your specific business needs are not only welcomed, but encouraged. Simply let us know your idea and then watch it come into fruition before your very eyes. Our sales team will be happy to help you.

Also make sure to check our website http://raynet.de to never miss upcoming releases, announcements, special offers, and product trainings.

**Be aware:**
Please keep in mind, that PackBench is currently in an early phase and still under development. If you encounter any bugs or other problems, please open an incident using the Raynet Support Panel.

# Additional Resources

Further information regarding PackBench can be found in several resources.

- The PackBench for RayPack 7.0 Release Notes provide an overview about the changes and new features that are part of this version of PackBench.
- The PackBench for RayPack 7.0 Operations Supplement provides information about third-party software packages and libraries redistributed with PackBench.
- The product website provides information about the product, news, and support.
- Raynet and its partners offer a range of training courses that can also be customized to meet your requirements. For more information on these courses, speak with your Raynet consultant or contact the Raynet Sales department via sales@raynet.de.

# System Requirements

The given requirements name prerequisites for devices running PackBench.

## Hardware Requirements

### Minimal

- CPU: Intel Core i5
- Screen resolution: 1024 x 768 pixels
- RAM: 4GB
- Disk space: 10GB

### Recommended

- CPU: Intel Core i7
- Screen resolution: 1280 x 1024 pixels
- RAM: 16GB or higher
- Disk space: 100GB or more

**Note:**
The installation of the PackBench framework itself requires about 100MB of disk space. The amount of additional space required depends on the amount of packaging material and the location of your data store.

## Supported OS

The following represents the list of supported operating systems at the time of release.

- Windows Vista SP2
- Windows 7 SP1
- Windows 8
- Windows 8.1
- Windows 10
- Windows Server 2008 R2
- Windows Server 2008 SP1
- Windows Server 2012
- Windows Server 2012 R2

- Windows Server 2016
- Windows Server 2019

# Prerequisite Software

## General

- .NET 4.5 Client & Full for Windows Vista up to Windows 8 systems (both 32-bit and 64-bit). Windows 10 already contains the required framework.

## RayFlow integration

In order to use the RayFlow functionality directly from PackBench, a running RayFlow server has to be accessible.

## Packaging and other Raynet Tools

In order to use out-of-the-box integration tools, RayPack Studio 7.0 must be installed (including RayPack, RayQC, RayQC Advanced and RayEval). Depending on the configuration of PackBench, additional tools and/or components of RaySuite may be required. To get more information about the command line usage of these tools refer to the respective *User Guides* of these products.

### Requirements for Server and Standalone Installation

Both server and standalone installation require an instance of SQL Server, version 2012 or higher. Express editions are also supported. The SQL Server can be installed on the same machine as PackBench components, or simply accessed from another machine.

# Product installation

In the following the installation of PackBench is described. PackBench can be used in one of the following scenarios:

- **Standalone installation**
  PackBench components are installed on the same machine. The data is stored on a database server which can be either available locally, or hosted by a third machine. Each installation has to know the details of SQL database server connection.

- **Server-client**
  PackBench components are split. Core components and services are installed on a central server, while other machines (clients) only require a thin PackBench client which then connects to the server. The data is stored on a database server which can be either available on the same machine where server components are installed, or hosted by yet another machine. The server components include a Windows Service which runs in the background and listens on a specific port. Each client only needs to know the URL and port that the server is listening to. The server configuration requires setting up the SQL database server connection. The server-client scenario allows easy abstraction, as the clients do not know anything about how and where the data is stored.

Regardless of whether the standalone or the client mode is installed, by the virtue of the fact that the database server can be hosted by any machine, both scenarios can be used to configure a shared workbench. Also, both scenarios are equally functional when the RayFlow interoperability is concerned.

**Be aware:**
The standalone instance of PackBench can only operate as a shared server as long as it is running. After the instance has been closed all the server capabilities are gone. On the other hand, a server installation of PackBench will install a windows server which will run all the time..

In the following the installation for *server*, *client*, and the combined installation as a *standalone device* are described.

# Installing standalone product

## Preparing the installation

The PackBench installation resources come along as an MSI file with additional documents. Before the application is installed on a device, some preparations are needed:

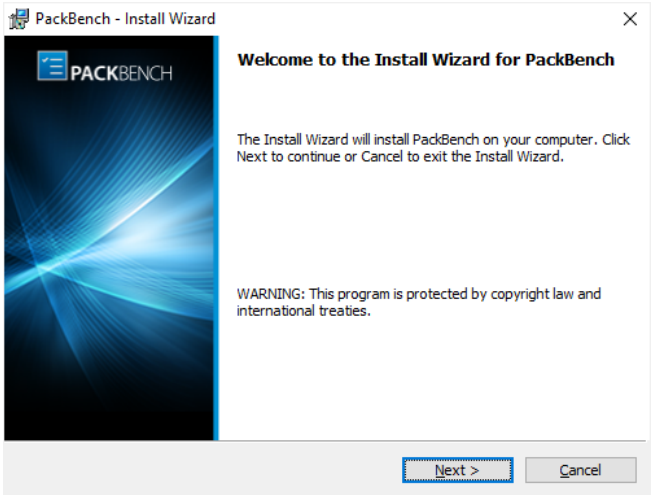
1. Please make sure to have your order number or license file at hand. Either one will be provided by our support team, or your Raynet sales representative.

2. The machine has to meet the system requirements described within the chapter *System Requirements*.
3. A Windows User with sufficient rights for installations has to be logged in.
4. Close all dispensable applications during the setup routine execution.
5. Ensure that the machine contains all necessary prerequisite software as mentioned in chapter *prerequisite software*.

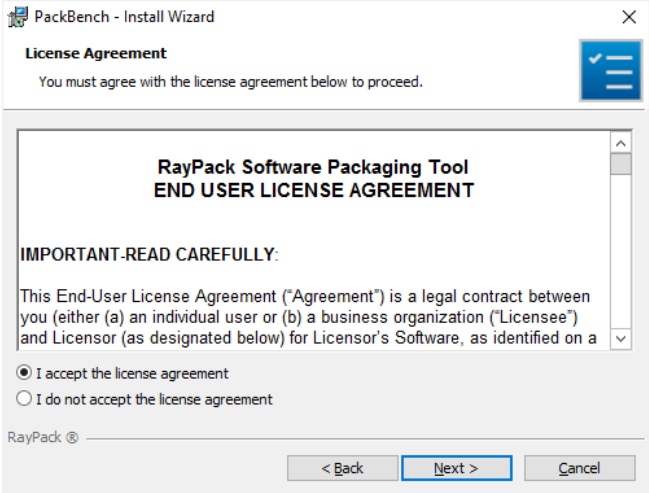
**Be aware:**
If no other configuration has been made, PackBench is using port 23624 by default. Please ensure, that this port is not blocked by a firewall. More details about required security configuration are to be found in chapter Security and required firewall configuration.

# Installing PackBench

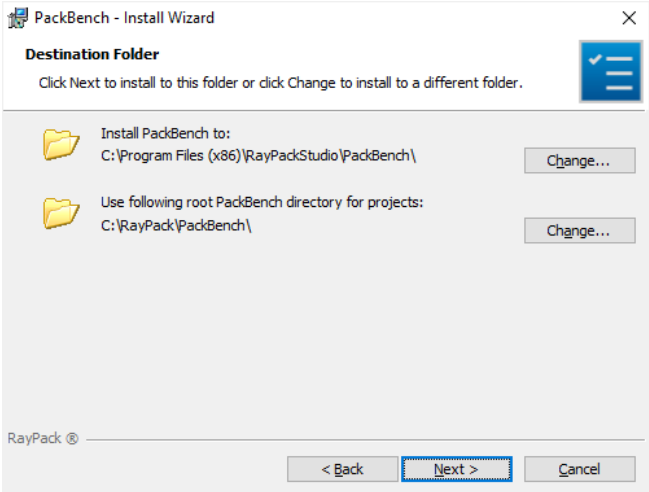
Launch the PackBench setup with a double-click on the MSI file and wait for the Welcome Screen to be prepared.



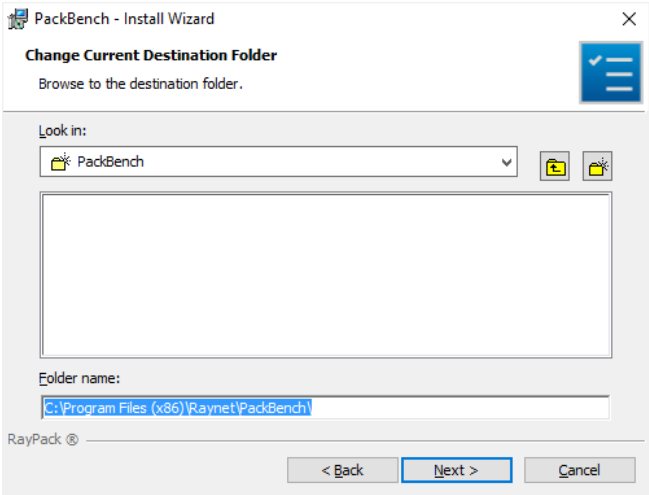

Click **Next >** to proceed.




In order to be allowed to install PackBench, the End User License Agreement has to be accepted. To do so, select the upper radio button and click **Next >** to proceed.

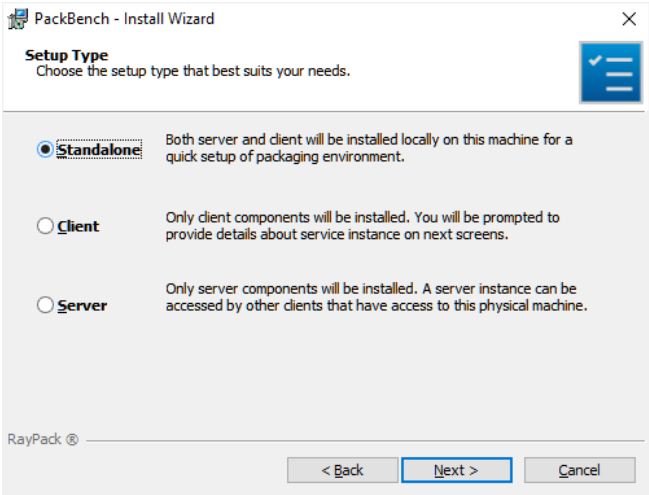



Please choose the destination folder and the root directory for projects by either keeping the suggested default or clicking on the **Change...** button to select another target directory.




If a custom destination folder has to be defined, the dialog displayed above has to be used. Please use the icons to navigate to the desired installation directory or to create a new one where necessary. As soon as the right target is defined, click **OK** to return to the Destination Folder screen. It will be updated to display the custom destination recently selected.

On the Destination Folder screen, click **Next >** to proceed.

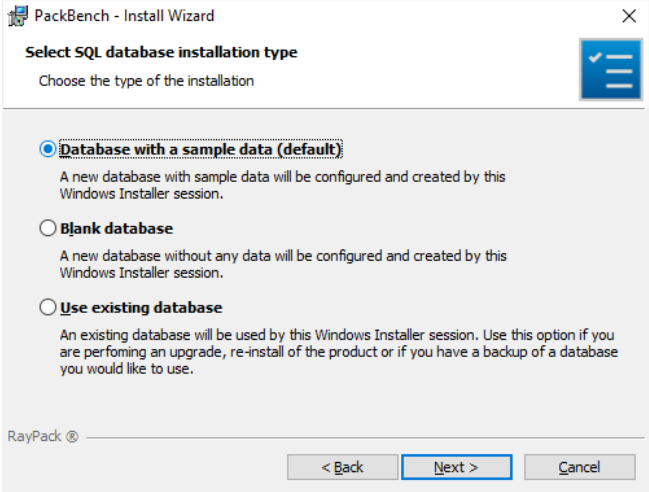



The Setup Type screen is displayed, presenting the three options available for PackBench installations:

- **Standalone** installs all required components on one machine. This is the easiest configuration, requiring only an access to an SQL server..
- **Client** installs the client component only. The client will need access to a server. This type of installation requires, that there is a server running the PackBench server components.
- **Server** installs the server component only. This instance can be accessed by all clients that have access to this physical machine. This type of installation requires configuration details of the SQL database being in use.

Press **Next >** to continue. The content of the upcoming step depends on the selected type.

### Select SQL database installation type screen

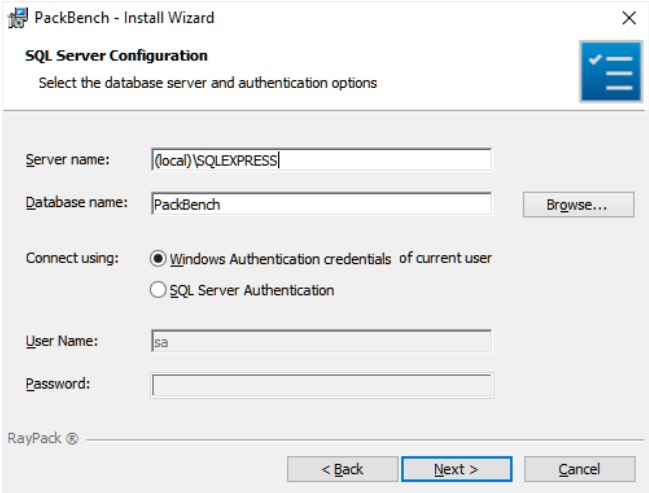



The type of the database configuration is chosen in this screen. There are three options available:

- **Database with a sample data (default)**: If this option is chosen, a new database will be created. The database will already contain some sample data. As already indicated, this is the default option.
- **Blank database**: If this option is chosen, an empty database will be created.
- **Use existing database**: If this option is chosen, an option to use the existing database (with update to the required version) will be provided.

Select the desired option and click on the **Next >** button to proceed.

### SQL Server Configuration Screen



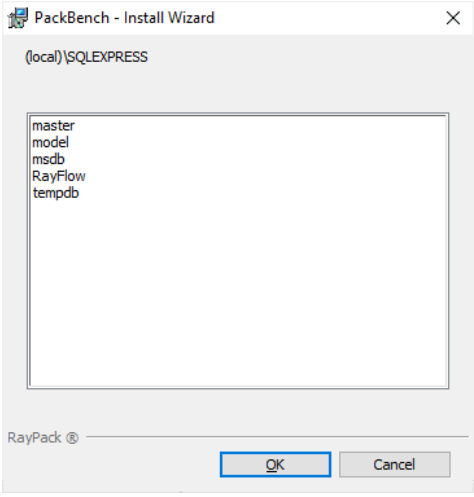
In the SQL Server Configuration screen the SQL server connection credentials will be set up.

- **Database server that you are installing to**: The SQL server needs to be defined. Please specify the server together with the instance (if an instance is required). The server can be specified either by its name or by its IP address. The SQL server must exist and must have a user which can create a database and has read and write access to the newly created database. Click on the **Browse...** button to select the instance by browsing to it.

  In order to provide an instance name, separate it from the server name using a backslash (\) character.

- **Connect using**: The authentication method which is used for PackBench is defined here. There are two options available:
  - **Windows authentication credentials of current user**: If this is chosen, the Windows authentication of the current user will be used for login.
  - **Server authentication using the Login ID and password below**: If this is chosen, enter an ID into the Login ID field and a Password into the Password field.

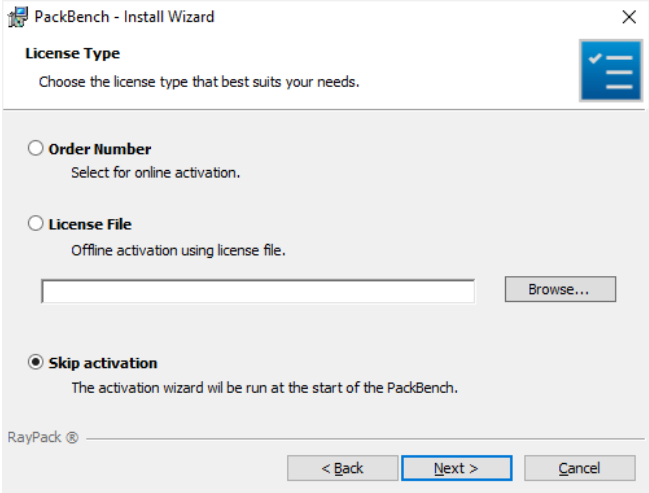
- **Name of database catalog**: Define the name of the database which will be used by PackBench. When clicking on the **Browse...** button a window will open which shows databases which are already configured at the server.




  Select one of these databases and click on the **OK** button to use an already existing database.

Click **Next >** to proceed.

### The License Type screen

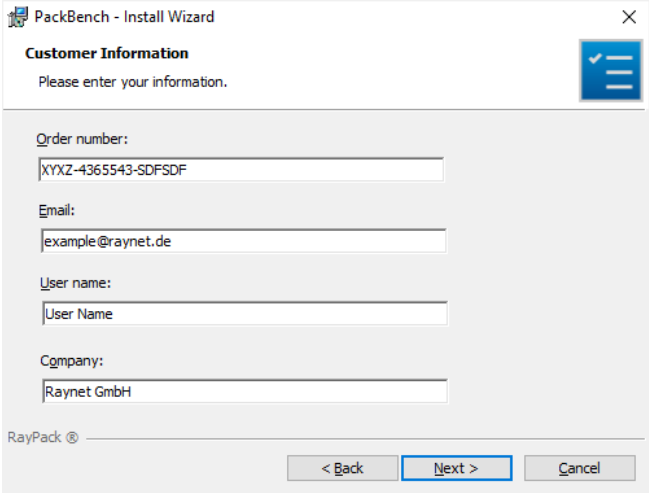



The License Type screen provides options to either activate the PackBench instance via order number and online activation service, by using an already prepared license file (`*.rsl`), or to skip activation for now. If the activation is skipped, it will need to be performed later when PackBench is launched.

To use an already existing license file, which most likely has been provided by the Raynet support team, the **Browse...** button has to be clicked. Use the controls of the system browser dialog to navigate to the `*.rsl` file and select it with a click on the **Open** button.

Click **Next >** to proceed.

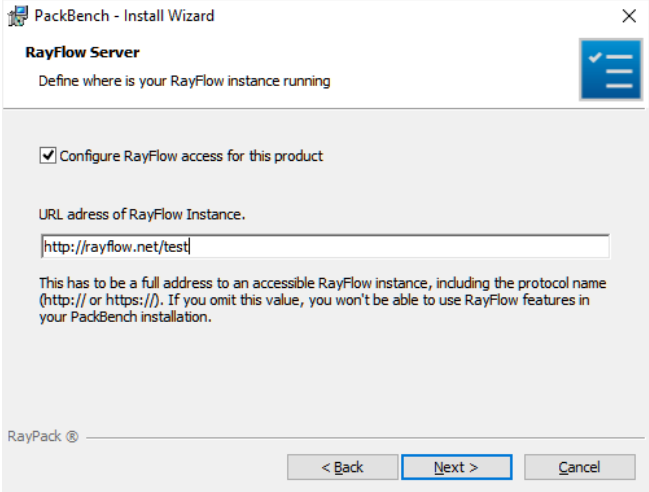
If activation by order number has been selected during the previous step, the Customer Information screen is displayed:




Please enter your individual PackBench Order number and provide required user information, such as E-Mail, user name, and company name. The information will be used to verify the order number during the upcoming installation execution procedure.

Click **Next >** to proceed.

### RayFlow screen

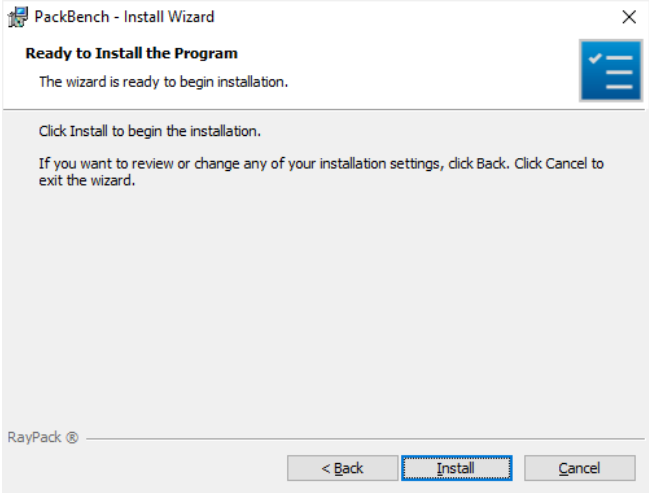



If PackBench will be used together with RayFlow, tick checkbox **Configure RayFlow access for this product** and enter the full RayFlow URL here. More information about RayFlow can be found in chapter *Working with RayFlow*.

**Note:**
If no RayFlow URL is entered, all RayFlow-related functionalities will be unavailable. In order to configure RayFlow instance after the installation is over, perform steps described in advanced topic *Changing RayFlow server*.

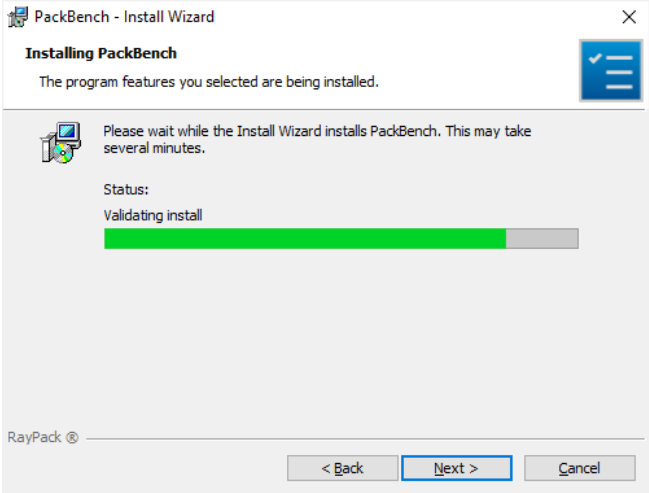
Click **Next >** to proceed.

### Ready to Install the Program screen

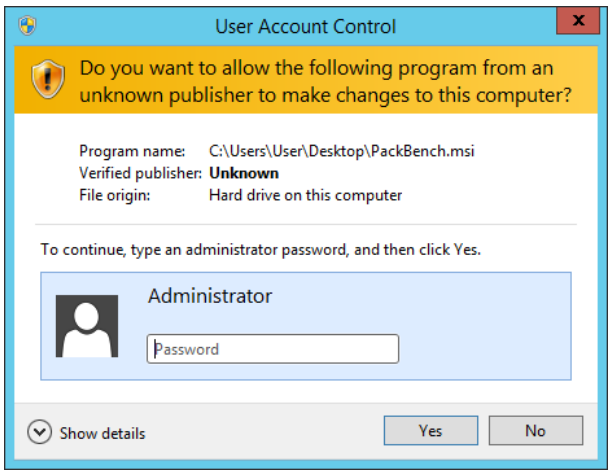


All required settings and properties are defined and PackBench is ready to be installed. Click **Install** to start the process.

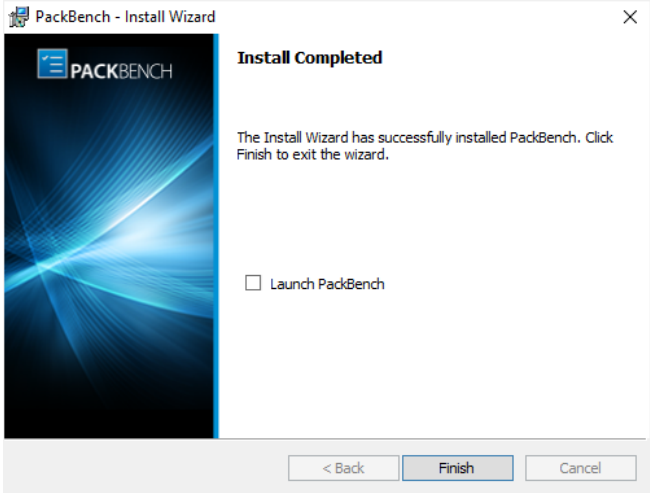
A Progress Indication dialog is displayed as long as the installation steps are executed.




Whilst the installation is executed, Windows may present a User Account Control message like the one shown below:




Click **Yes** to perform the installation.




As soon as all required measures are done, the Install Completed dialog is presented. Activate the checkboxes to immediately launch PackBench and to read the Release Notes with essential information about the requirements of the installed product version, as well as listings of new features and issues which have been solved or documented.

Click **Finish** to exit the setup.

If the **Launch PackBench** option has been activated, the application is loaded immediately. During the software launch, another Windows User Account Control message may be displayed.

Please click **Yes** to proceed if this message occurs.

As soon as the *Home Screen* of PackBench is displayed, your packaging tool is ready for action.

# Installing server component

## Preparing the installation

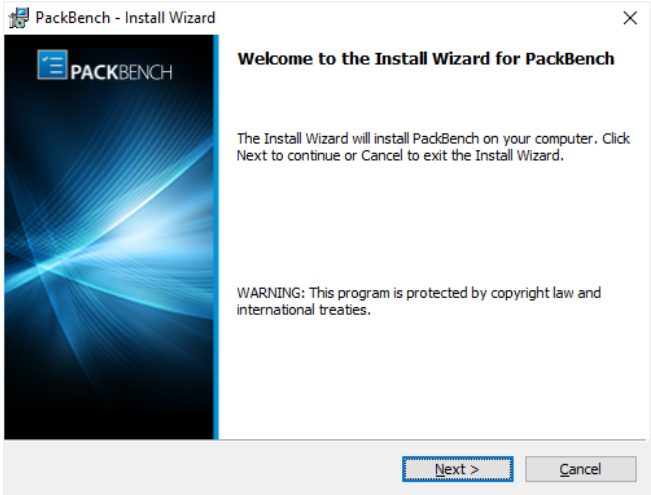
The PackBench installation resources come along as an MSI file with additional documents, such as the current Release Notes. Before the application is installed on a server, some preparations are needed:

1. The server machine has to meet the system requirements described within the chapter *System Requirements*.
2. A Windows User with sufficient rights for installations has to be logged in.
3. Close all dispensable applications during the setup routine execution.
4. Ensure that the machine contains all necessary prerequisite software as mentioned in chapter *prerequisite software*.

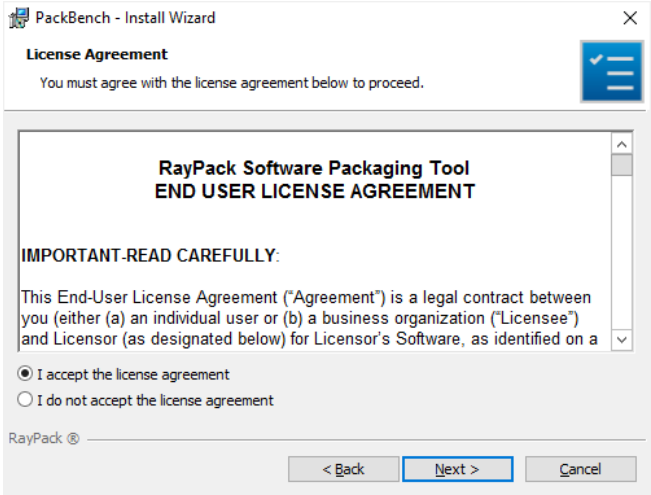
**Be aware:**
If no other configuration has been made, PackBench is using port 23624 by default. Please ensure, that this port is not blocked by a firewall. More details about required security configuration are to be found in chapter Security and required firewall configuration.

# Installing PackBench

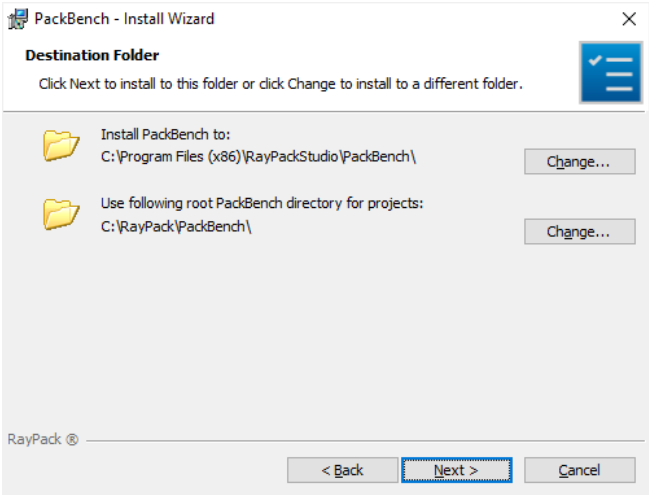
Launch the PackBench setup with a double-click on the MSI file and wait for the Welcome Screen to be prepared.



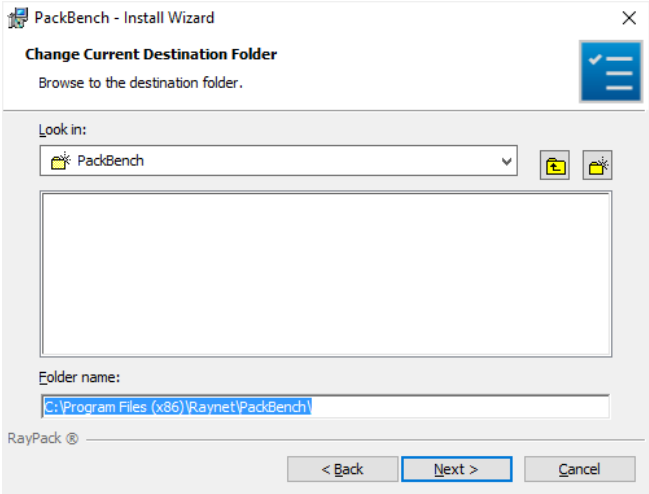

Click **Next >** to proceed.




In order to be allowed to install PackBench, the End User License Agreement has to be accepted. To do so, select the upper radio button and click **Next >** to proceed.

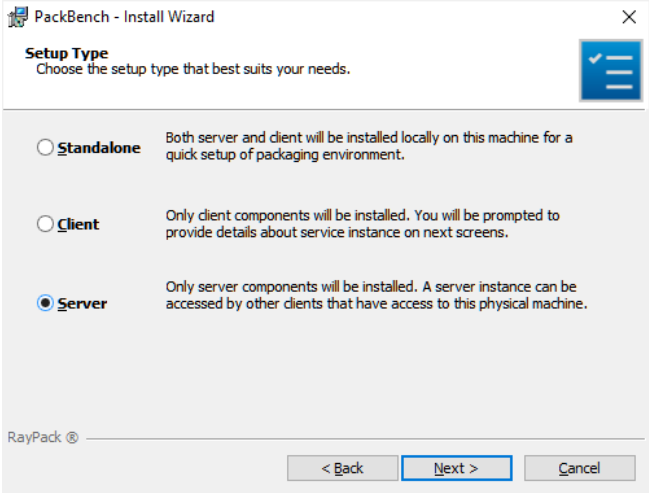



Please choose the destination folder and the root directory for projects by either keeping the suggested default or clicking on the **Change...** button to select another target directory.




If a custom destination folder has to be defined, the dialog displayed above has to be used. Please use the icons to navigate to the desired installation directory or to create a new one where necessary. As soon as the right target is defined, click **OK** to return to the Destination Folder screen. It will be updated to display the custom destination recently selected.

On the Destination Folder screen, click **Next >** to proceed.

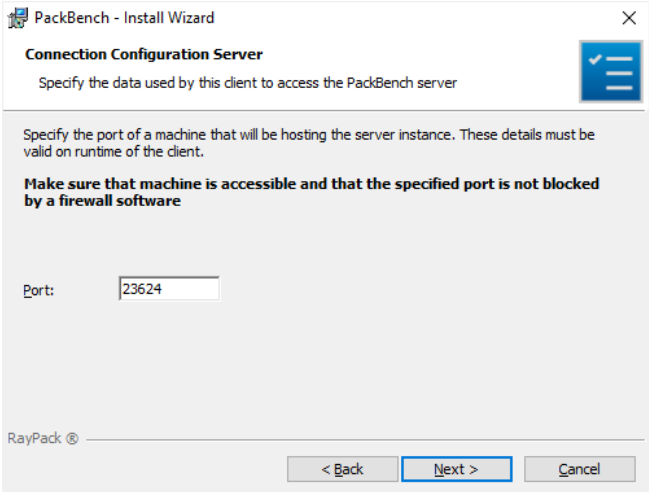



The Setup Type screen is displayed, presenting the three options available for PackBench installations:

- **Standalone** installs required components on a same machine. This is the simplest configuration, requiring only an access to an SQL server..
- **Client** installs the client component only. The client will need access to a server. This type of installation requires that there is a server running PackBench server components.
- **Server** installs the server component only. This instance can be accessed by all clients that have access to this physical machine. This type of installation requires configuration details of SQL database being in use.

Press **Next >** to continue. The content of the upcoming step depends on the selected type.

### Connection Configuration Server screen

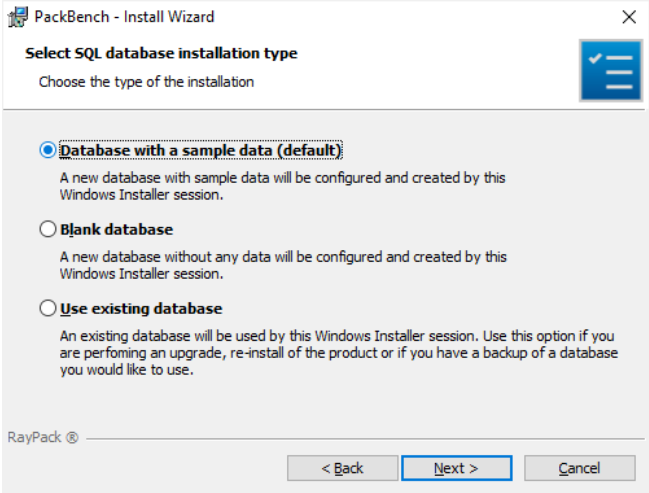



The port of the server is specified on this page. Before continuing with the next step ensure that the port is accessible.

**Warning:**
Ensure that the machine is accessible and that the given port is not blocked by a firewall software!

Click on the **Next >** button to proceed.

### Select SQL database installation type screen

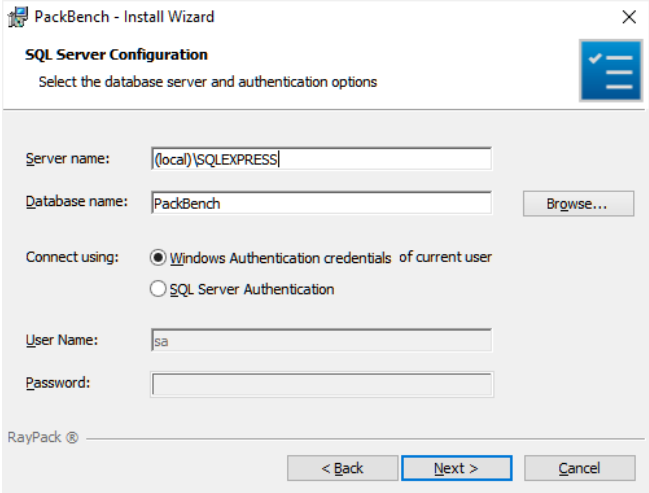



The type of the database configuration is chosen in this screen. There are three options available:

- **Database with a sample data (default)**: If this option is chosen, a new database will be created. The database will already contain some sample data. As already indicated, this is the default option.
- **Blank database**: If this option is chosen, an empty database will be created.
- **Do not configure**: If this option is chosen, no database will be configured. Continue with the *Ready to Install Program* screen.

Select the desired option and click on the **Next >** button to proceed.

### SQL Server Configuration Screen

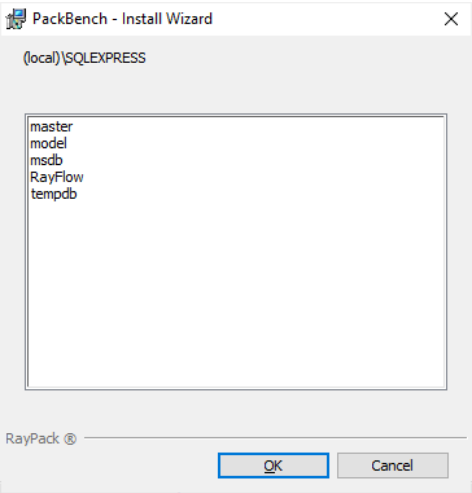



In the SQL Server Configuration screen the SQL server connection credentials will be set up.

- **Database server that you are installing to**: The SQL server needs to be defined. Please specify the server together with the instance (if an instance is required). The server can be specified either by its name or by its IP address. The SQL server must exist and must have a user which can create a database and has read and write access to the newly created database. Click on the **Browse...** button to select the instance by browsing to it.

- **Connect using**: The authentication method which is used for PackBench is defined here. There are two options available:
  - **Windows authentication credentials of current user**: If this is chosen, the Windows authentication of the current user will be used for login.
  - **Server authentication using the Login ID and password below**: If this is chosen, enter an ID into the Login ID field and a Password into the Password field.

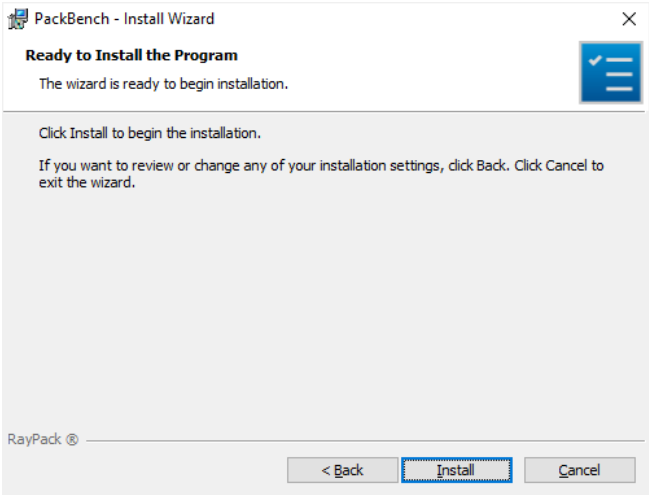
- **Name of database catalog**: Define the name of the database which will be used by PackBench. When clicking on the **Browse...** button a window will open which shows databases which are already configured at the server.



Select one of these databases and click on the **OK** button to use an already existing database.

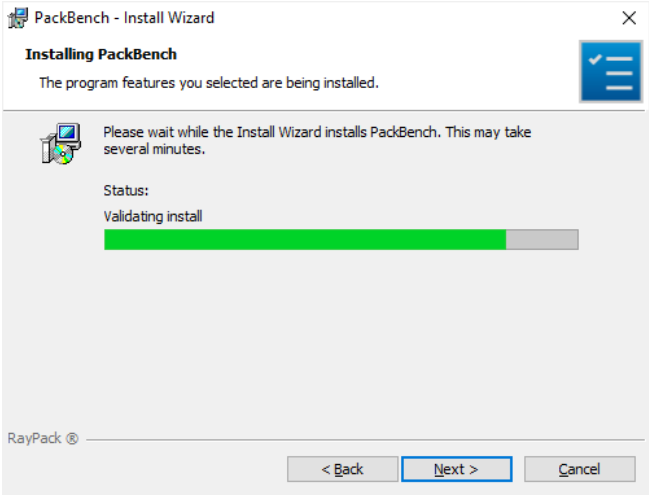
Click **Next >** to proceed.

### Ready to Install the Program screen

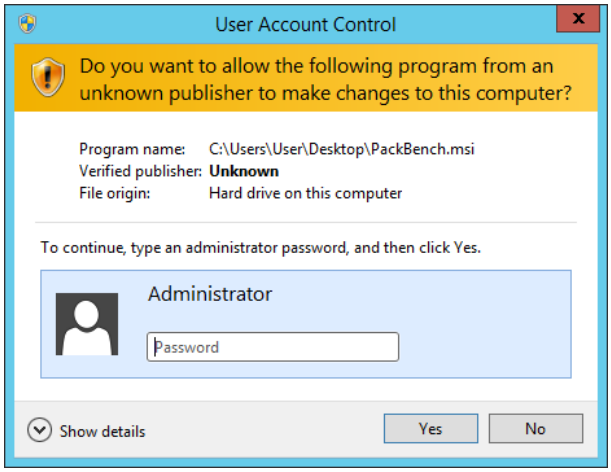



All required settings and properties are defined - PackBench is ready to be installed. Click **Install** to start the process.

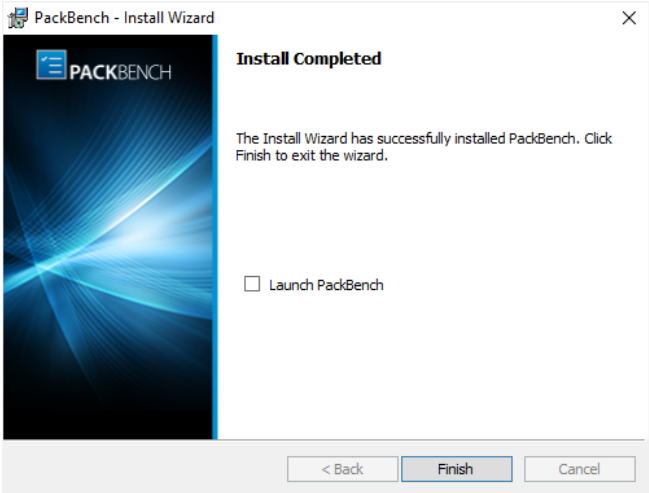
A Progress Indication dialog is displayed as long as the installation steps are executed.




Whilst the installation is executed, Windows may present a User Account Control message like the one shown below:




Click **Yes** to perform the installation.




As soon as all required measures are done, the Install Completed dialog is presented. Activate the checkboxes to immediately launch PackBench and to read the Release Notes with essential information about the requirements of the installed product version, as well as listings of new features and issues which have been solved or documented.

Click **Finish** to exit the setup.

If the **Launch PackBench** option has been activated, the application is loaded immediately. During the software launch, another Windows User Account Control message may be displayed.

Please click **Yes** to proceed if this message occurs.

# Installing client component

## Preparing the installation

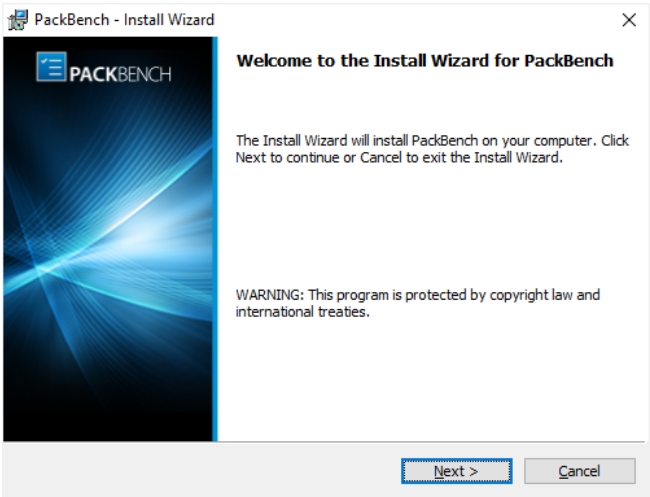
The PackBench installation resources come along as an MSI file with additional documents. Before the application is installed on a device, some preparations are needed:

1. Please make sure to have your order number or license file at hand. Either one will be provided by our support team, or your Raynet sales representative.
2. The machine has to meet the system requirements described within the chapter *System Requirements*.
3. A Windows User with sufficient rights for installations has to be logged in.
4. Close all dispensable applications during the setup routine execution.
5. Ensure that the machine contains all necessary prerequisite software as mentioned in chapter *prerequisite software*.

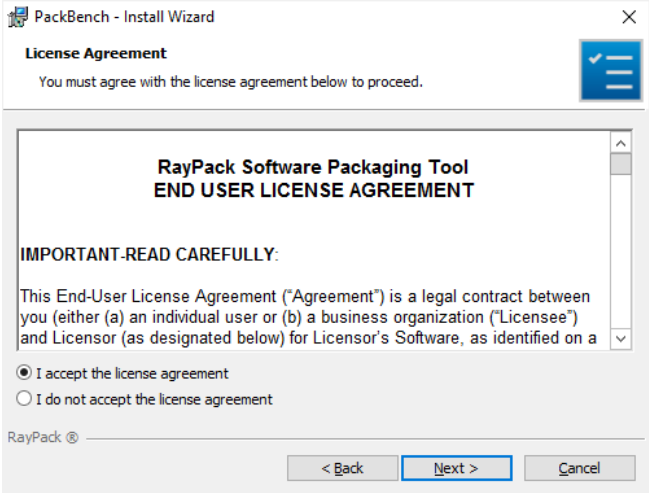
**Be aware:**
If no other configuration has been made, PackBench is using port 23624 by default. Please ensure, that this port is not blocked by a firewall.

## Installing PackBench

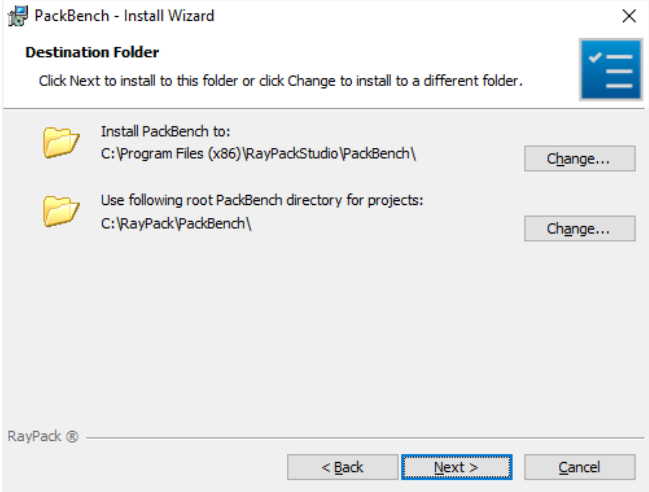
Launch the PackBench setup with a double-click on the MSI file and wait for the Welcome Screen to be prepared.



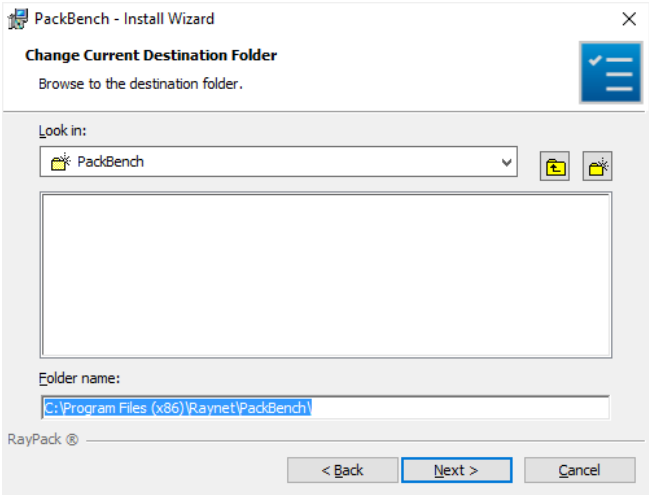

Click **Next >** to proceed.




In order to be allowed to install PackBench, the End User License Agreement has to be accepted. To do so, select the upper radio button and click **Next >** to proceed.

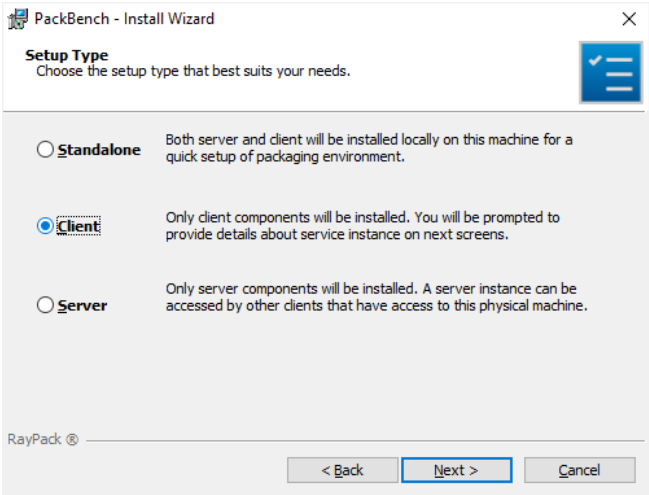



Please choose the destination folder and the root directory for projects by either keeping the suggested default or clicking on the **Change...** button to select another target directory.




If a custom destination folder has to be defined, the dialog displayed above has to be used. Please use the icons to navigate to the desired installation directory or to create a new one where necessary. As soon as the right target is defined, click **OK** to return to the Destination Folder screen. It will be updated to display the custom destination recently selected.

On the Destination Folder screen, click **Next >** to proceed.

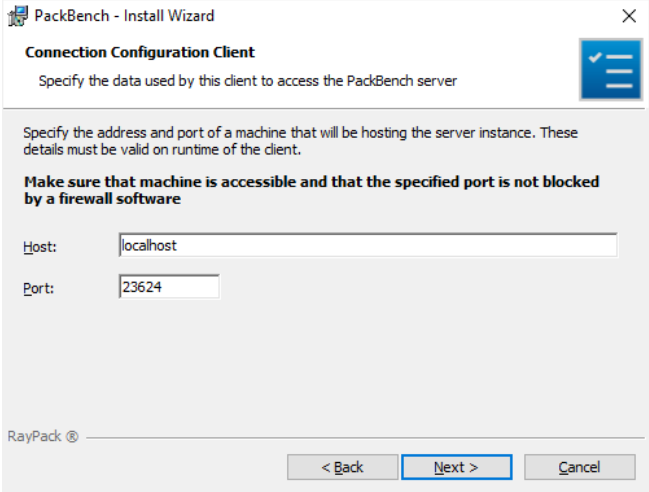



The Setup Type screen is displayed, presenting the three options available for PackBench installations:

- **Standalone** installs required components on a same machine. This is the simplest configuration, requiring only an access to an SQL server..
- **Client** installs the client component only. The client will need access to a server. This type of installation requires that there is a server running PackBench server components.
- **Server** installs the server component only. This instance can be accessed by all clients that have access to this physical machine. This type of installation requires configuration details of SQL database being in use.

Press **Next >** to continue.

### Connection Configuration Client screen



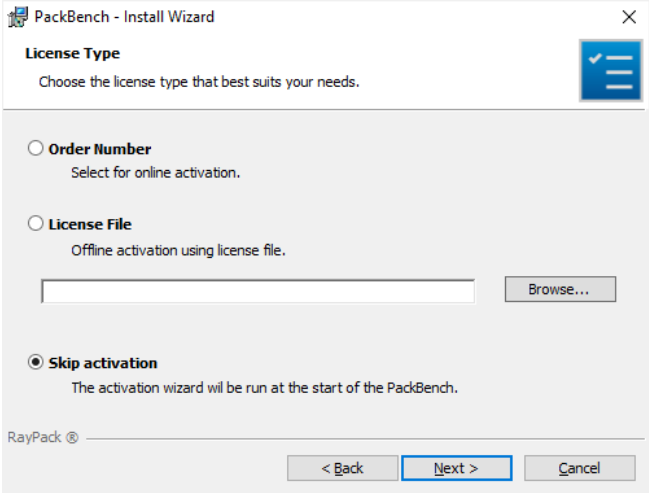

In the Connection Configuration Client screen the connection to the device which hosts the PackBench server is configured. In this screen there are two fields which are mandatory for the configuration of the connection:

**Host**: The PackBench server needs to be defined here. The server can be specified either by its name or by its IP address.

**Port**: Port: The port for the connection has to be entered here. Please ensure that the port is identical to the one that was used during the set up of the PackBench server.

Click on **Next >** to proceed.

### The License Type screen

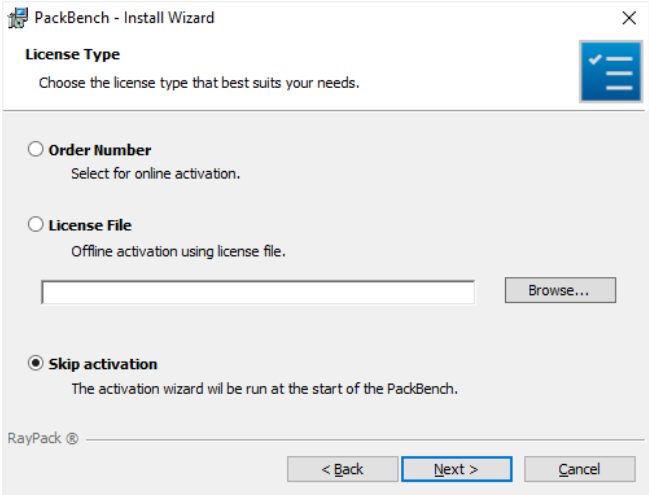


The License Type screen provides options to either activate the PackBench instance via order number and online activation service, by using an already prepared license file (`*.rsl`), or to skip activation for now. If the activation is skipped, it will need to be performed later when PackBench is launched.

To use an already existing license file, which most likely has been provided by the Raynet support team, the **Browse...** button has to be clicked. Use the controls of the system browser dialog to navigate to the `*.rsl` file and select it with a click on the **Open** button.

Click **Next >** to proceed.

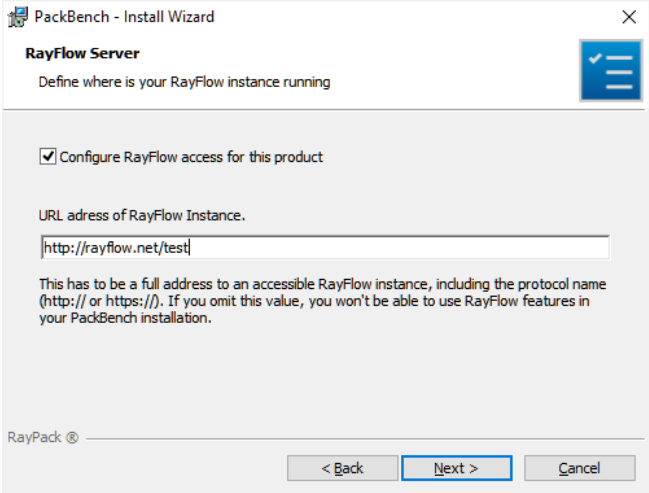
If activation by order number has been selected during the previous step, the Customer Information screen is displayed:




Please enter your individual PackBench Order number and provide required user information, such as E-Mail, user name, and company name. The information will be used to verify the order number during the upcoming installation execution procedure.

Click **Next >** to proceed.

### RayFlow screen

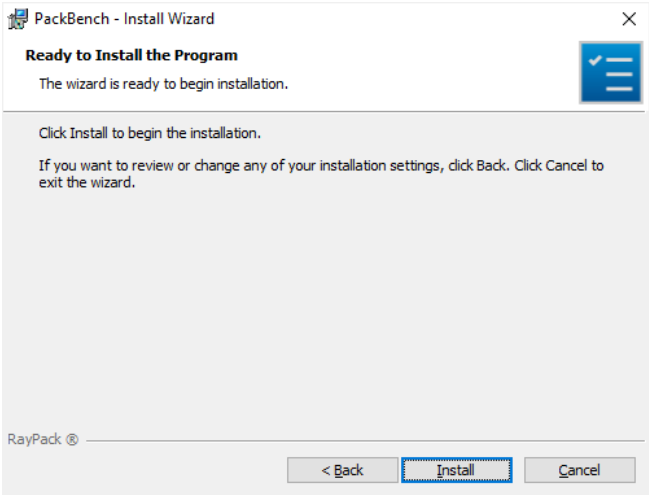



If PackBench is used together with RayFlow, tick checkbox **Configure RayFlow access for this product** and enter the full RayFlow URL here. More information about RayFlow can be found in chapter *Working with RayFlow*.

**Note:**
If no RayFlow URL is entered, all RayFlow-related functionalities will be unavailable. In order to configure RayFlow instance after the installation is over, perform steps described in advanced topic *Changing RayFlow server*.

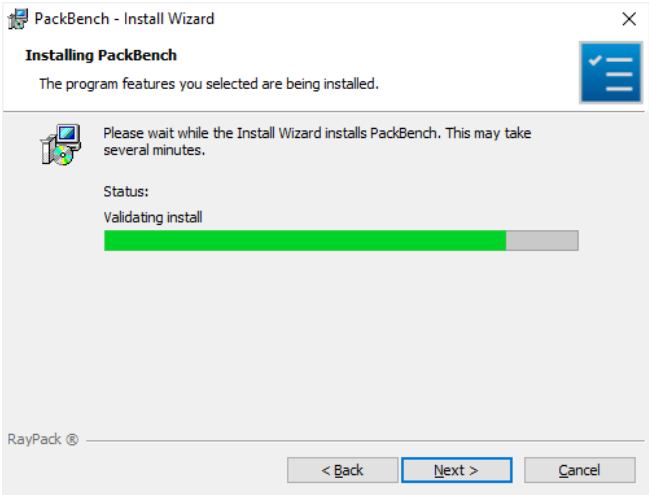
Click **Next >** to proceed.

### Ready to Install the Program screen

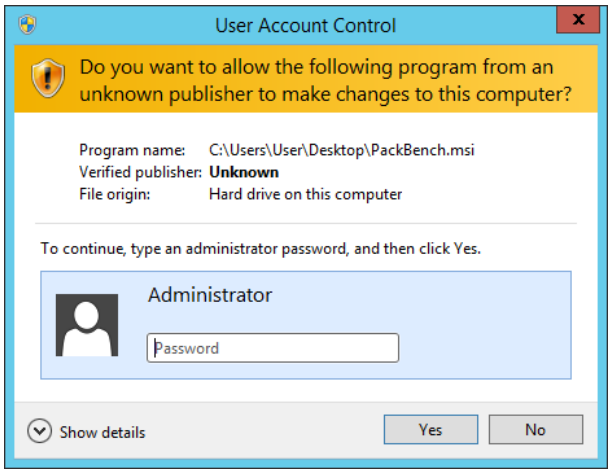



All required settings and properties are defined and PackBench is ready to be installed. Click **Install** to start the process.

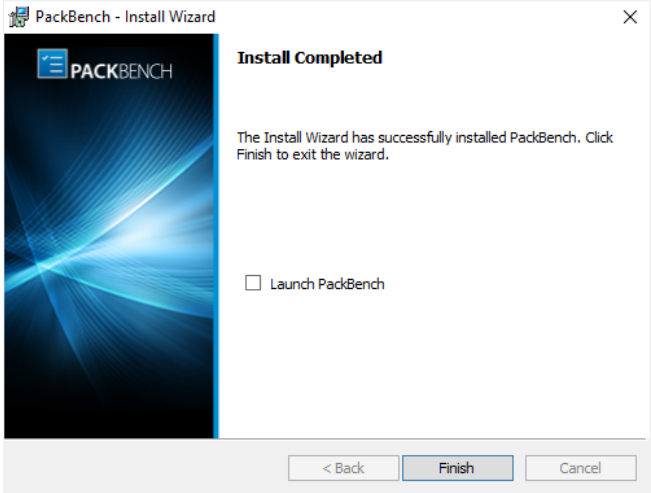
A Progress Indication dialog is displayed as long as the installation steps are executed.




Whilst the installation is executed, Windows may present a User Account Control message like the one shown below:




Click **Yes** to perform the installation.




As soon as all required measures are done, the Install Completed dialog is presented. Activate the checkboxes to immediately launch PackBench and to read the Release Notes with essential information about the requirements of the installed product version, as well as listings of new features and issues which have been solved or documented.

Click **Finish** to exit the setup.

If the **Launch PackBench** option has been activated, the application is loaded immediately. During the software launch, another Windows User Account Control message may be displayed.

Please click **Yes** to proceed if this message occurs.

As soon as the *Home Screen* of PackBench is displayed, your packaging tool is ready for action.

# Security and required firewall configuration

## Requirements for Standalone Product

Standalone installation of PackBench requires that the port **23624** is opened. The port is used for a local TCP/IP communication between client UI part and a local proxy to access the SQL server.

The following executable files are concerned (they can both be found located in the root installation folder, by default `C:\Program Files (x86)\RayPack\PackBench\`):

- Client executable is `PackBench.exe`
- Local proxy is `PackBenchService.exe`

**Be aware:**
`PackBenchService.exe` is not meant to be started manually. By doing so, a warning

message will be shown instead. `PackBench.exe` should always be used as the application entry point.

## Requirements for Server Components

Server installation of PackBench requires the port **23624** is opened. The installation deploys a Windows Service which listens on this port to handle incoming and outgoing communication with clients.

**Be aware:**
The default port `23624` can be changed during the installation.

The name of the Windows Service installed by PackBench is `PackBenchService` and it is started from `PackBenchService.exe` from the main installation folder, by default `C:\Program Files (x86)\RayPack\PackBench\`. The service is by default set to auto-start mode, disabling or stopping the service causes that the incoming requests are not processed, and as such clients cannot communicate and work at all. The service runs by default as user **Network Service**.

When using integrated Windows Authentication, the service has to be allowed to read and write into the database created by PackBench installation. This is a checklist of things that have to be verified::

1. The Network Service user has to exist as a SQL Server login.

2. The login has to use PackBench as default database.

3. The login has a user mapping to the PackBench database.

4. The login has to be member of the `db_datareader` and `db_datawriter` for PackBench database.

## Requirements for Client Components

Client installation deploys a thin client `PackBench.exe` to the main installation folder, by default `C:\Program Files (x86)\RayPack\PackBench\`). This executable has to be able to access the URL of PackBench service via the port number `23624`.

**Be aware:**
The default port `23624` can be changed during the installation.

# Product activation

The product can be activated using one of the following methods:

- Directly within installation
  - By supplying the order number
  - By supplying an already generated license file (`.rsl` format)
- When the product is started for the first time.

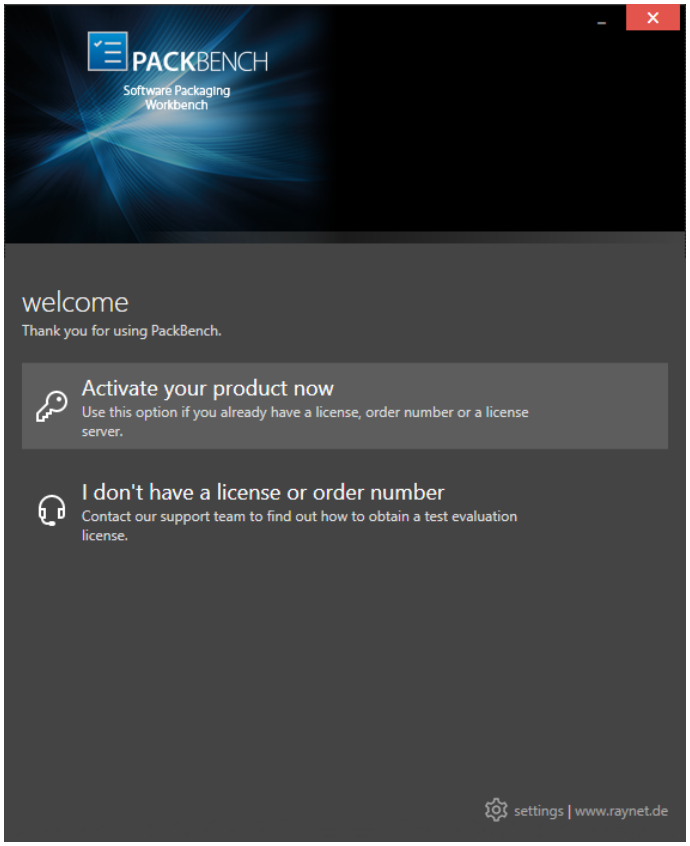
If PackBench detects that no valid license is present on start-up, the license activation wizard will be shown after starting the main executable. The tool can be also started manually, by executing `Raynet.LicenseActivation.exe` from the main installation folder.

# License wizard

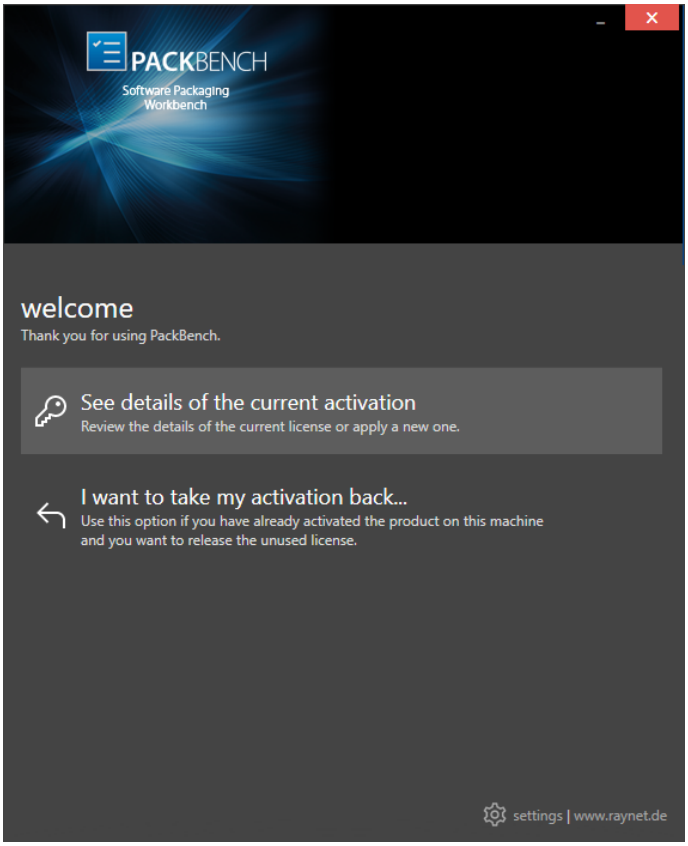
This section describes the usage of the License Activation Tool.

**Be aware:**
The license is needed for the PackBench clients only! For the PackBench server no license is needed!

On the initial start of PackBench, the License Activation Tool is shown. If the need to transfer an existing license arises, the License Activation Tool can be started manually. There are a variety of ways in which a license can be activated and below they are described in detail.



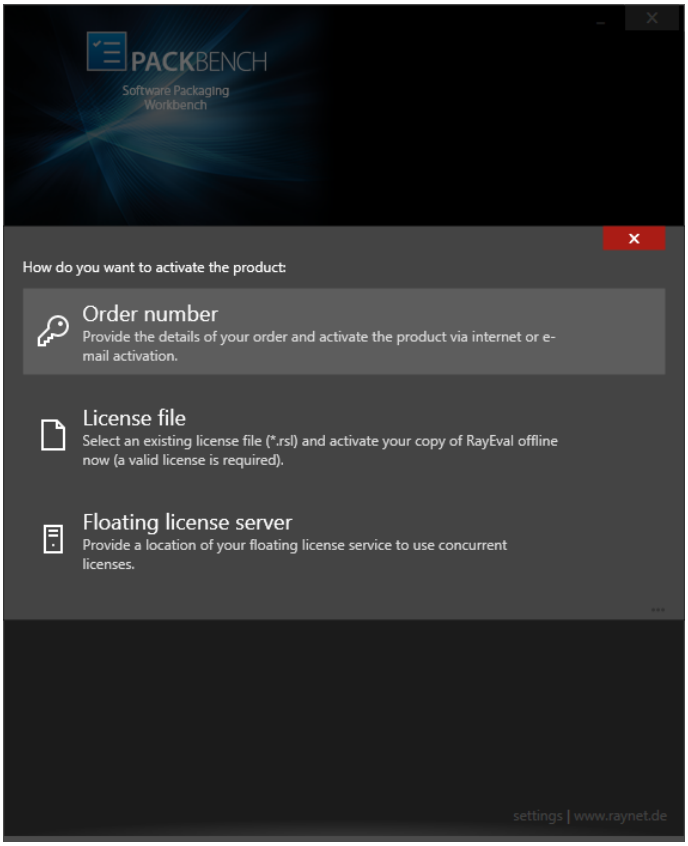

First time activation




The main screen when the product has already been activated

**Activate your product now**

This option should be used to activate the product using one of the following methods:




- **Order number**
  Online activation using a valid order number received from Raynet (recommended for most users).

- **License file**
  Offline activation using a license file (`.rsl`) received from Raynet.

- **Floating license server**
  Activation using a local floating license server.

### See details of the current activation

This options shows the details of the current activation. This option is only visible if the product has already been activated or if a floating license server has been configured.
This option also allows one to reactivate the product using a different order number or different floating license server connection details.

### I don't have a license or order number

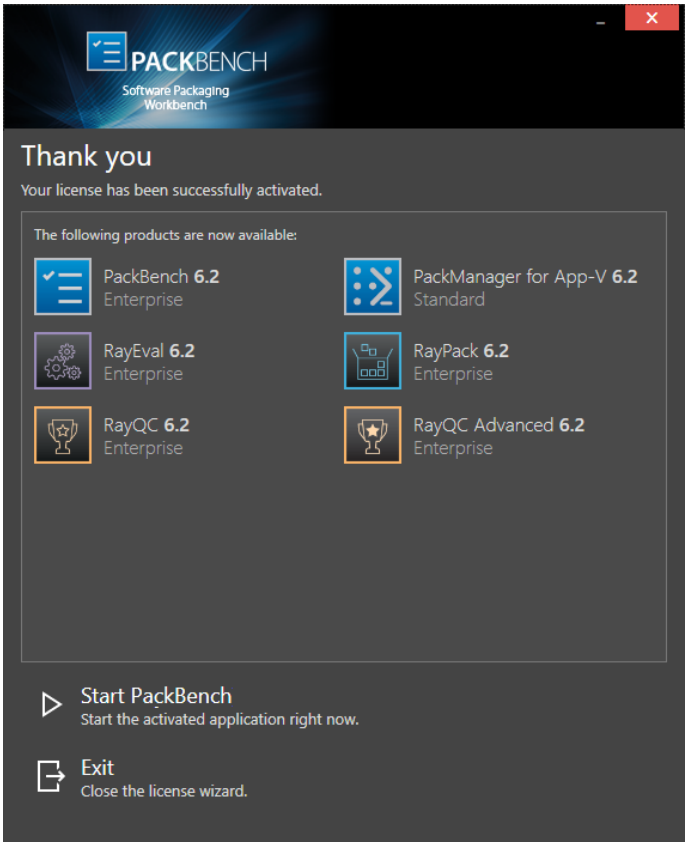
Choose this option if there is neither a license nor an order number available. For in-depth information please read this section. This option is only visible if the product has not been activated yet.

### I want to take my activation back...

Use this option to deactivate a currently licensed version of PackBench. For in-depth information

please read this section. This options is only visible if the product has already been activated.

Once the license file has been generated or copied to the correct location the following will be shown...



**Note:**
Depending on the license, a different amount of available products may be shown, as pictured above.

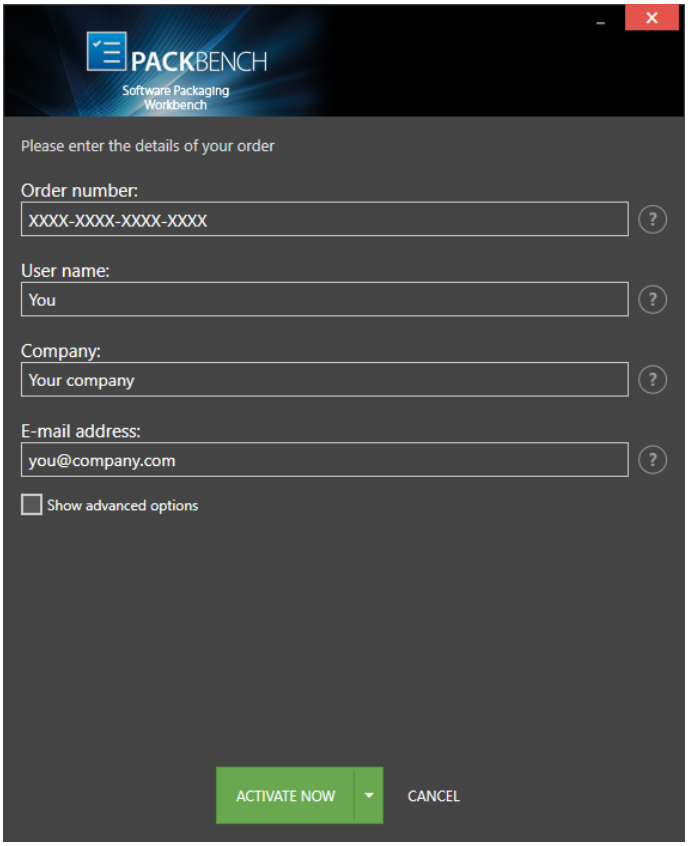
Then the option of starting PackBench or just closing the activation wizard is made available.

## Troubleshooting

If any issues arise during the activation process, please contact our help desk to receive assistance in activating PackBench.

# Order number

PackBench can be activated either directly online or via email once the order number has been delivered. The activation process generates a license file (`*.rsl`) that is created (or must be copied) to the installation directory of PackBench (in the same location as the `PackBench.exe`). When performing an online activation, sufficient permissions must be readily available to allow the creation of the license file in the installation directory. The activation **binds** the license to the machine on which it was activated on. This is the only time that an active connection to the internet is required (if activating online).




Choosing the **ACTIVATE NOW** button, connects to the Raynet license server using the

information provided and will dynamically generate a license file. Choosing the **ACTIVATE MANUALLY** button will open a dialog as shown here. Choosing the **CANCEL** button will abort the activation process.

# Order details

**Order number:**
This is the unique order number received when PackBench has been purchased. If it is necessary to recover the order number, please contact our sales team.

**User name:**
This is the name of the user that is activating PackBench. It does not need to be the same name used to order PackBench.

**Company:**
This is the name of the company for which PackBench will be licensed. This name will appear in the License and Edition view of PackBench.

**E-mail address:**
This is the email address of the person that performs the activation. We respect the privacy of our customers, this email address will only be used by Raynet and only when there are any problems or important information regarding the license.

# Advanced Options

On choosing the advanced options check box, extended information and possibilities of the licensing and activation of PackBench are shown.

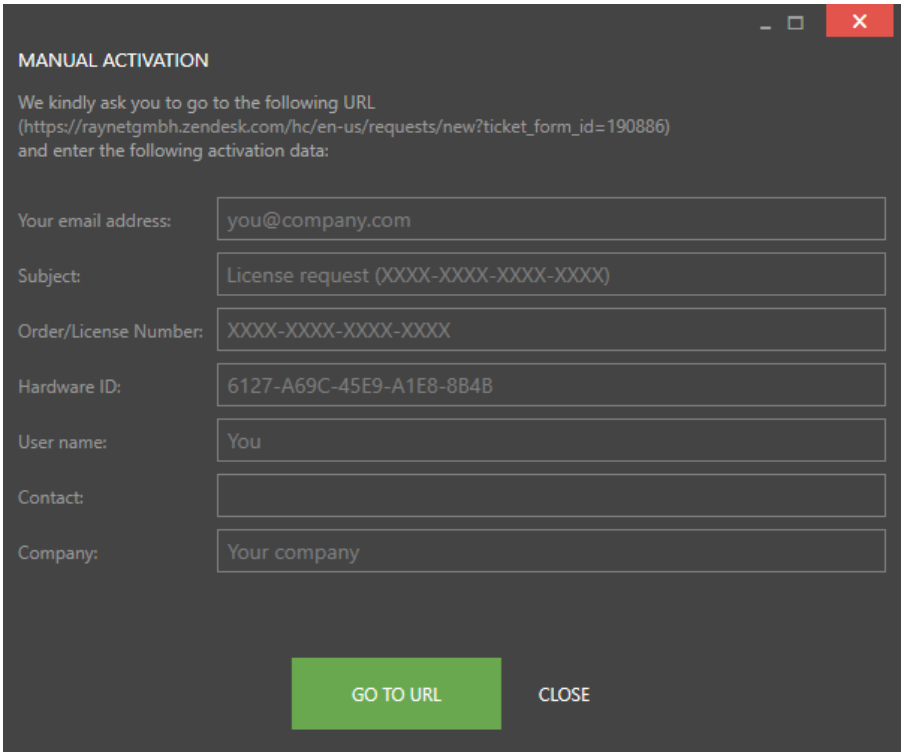
**Hardware ID:**
This is an ID calculated based on the hardware on which the activation is taking place on. The ID is unique, but cannot be used to personally identify a user. It is used to generate the license for the machine on which the activation process is carried out on.

**Transfer the license**
If this option is selected, the order number and details may be used to activate PackBench on a second machine, that has differing hardware (which obviously has a different Hardware ID). This assumes that PackBench has been deinstalled from the machine on which it was previously activated on. The transfer license functionality is logged on our license servers and is periodically checked to ensure that no abuse is made of this functionality.

If the license transfer is part of a regular maintenance and can therefore be prepared and scheduled, it is highly recommended to use the deactivation function first, to disconnect license and packaging machine. This is the standard way for transferring licenses. The option offered here is intended for unscheduled transfers, required if a machine, for whatever reason, cannot be accessed or used operational any longer.

# Manual Activation

On choosing the manual activation, the dialog shown below is displayed.

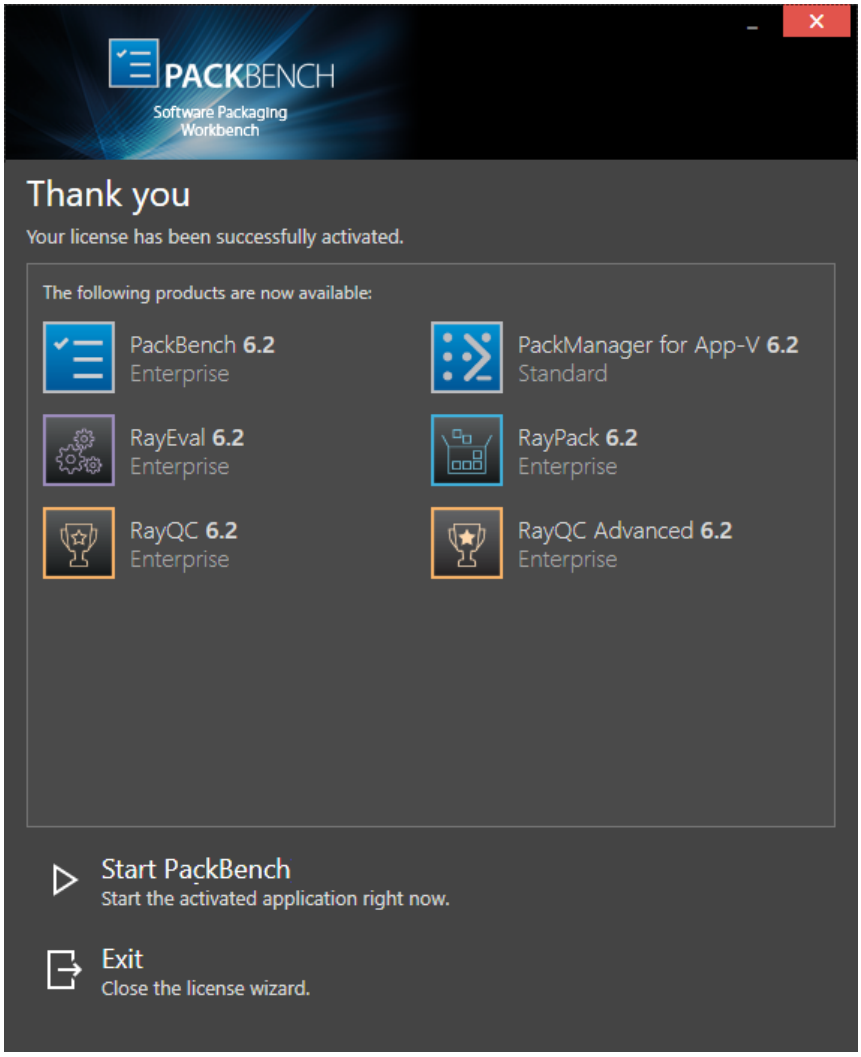



This basically shows the contents of the ticket form that will be opened at Raynet. If there is an internet connection available on the machine, click on the **GO TO URL** button to open the URL shown in the top of the window in the default browser of the system. After a File Order has been opened in the Raynet Support Panel, a license file will be delivered. Information of how to use this file are available here.

If no internet connection is present on the machine on which the activation process is taking place, copy the contents of the dialog onto a machine which has an internet connection and use the URL on that machine. On receiving the ticket, a license file will be generated and sent back. Information on how to use the license file can be found here.

**Tip:**
Please ensure that when copying the information from the **MANUAL ACTIVATION** dialog everything is added as shown above.

Once the license file has been generated the following will be shown:




**Note:**
Depending on the license, more available products may be shown. As an example, see the image above.

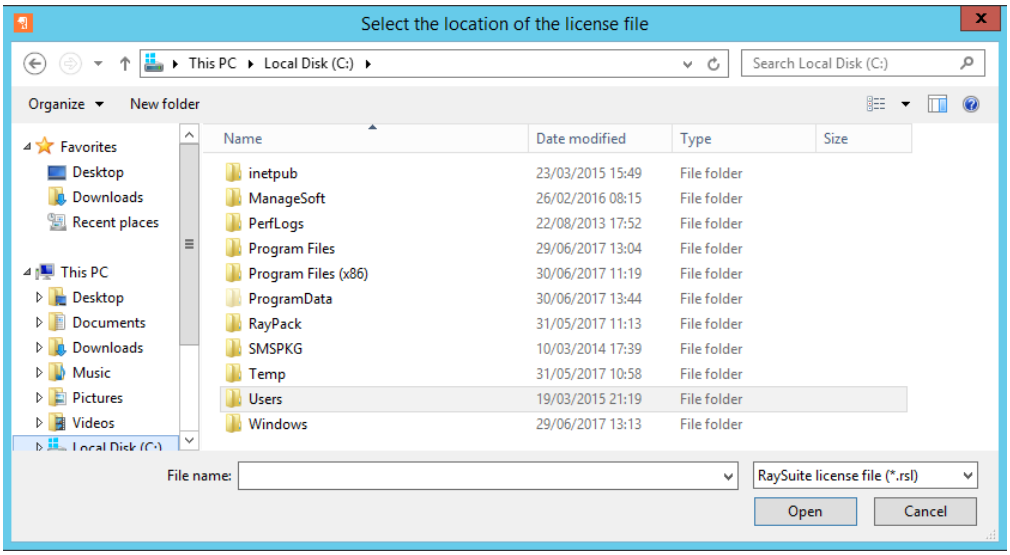
The option of starting PackBench or just closing the activation wizard are available now.

# Troubleshooting

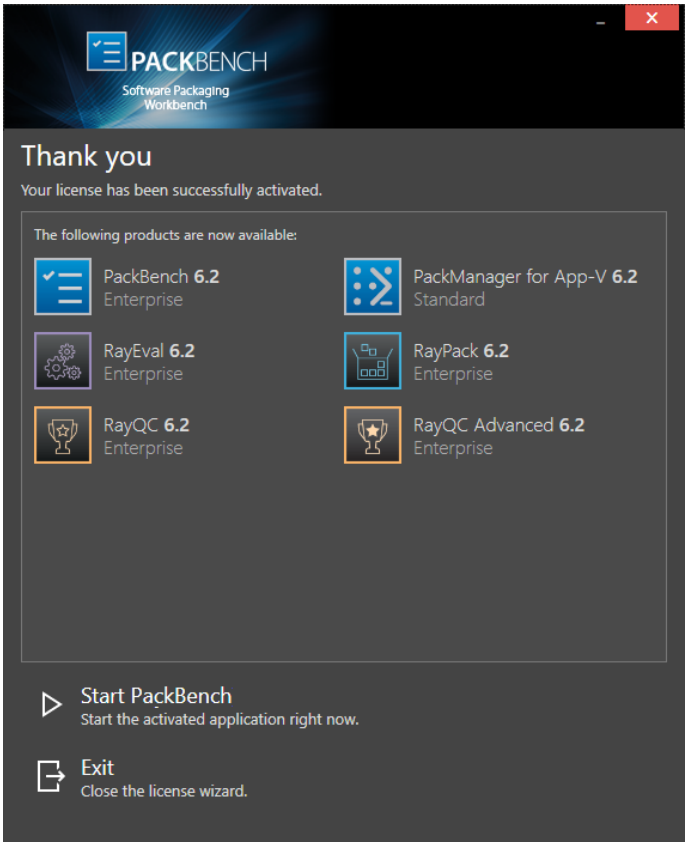
If there are any problems during the activation process, please contact our help desk for receiving assistance in activating PackBench.

# License file

If a license is already available, or a license file has been received as a result of activating PackBench via e-mail, then all that is required is to copy the license file into the installation directory of PackBench (the directory in which the PackBench.exe resides). Clicking on the **I have a license** button on the **License wizard** dialog opens a dialog box which allows one to choose the license file. Once chosen, the file will be copied automatically to the PackBench installation directory. Please ensure that sufficient permissions to allow the creation/copying of a file to the installation directory of PackBench are available.




Once the license file has been copied to the correct location the following will be shown:




**Note:**
Depending on the license, more available products may be shown. As an example, see the image above.

The option of starting PackBench or just closing the activation wizard are available now.

## Troubleshooting

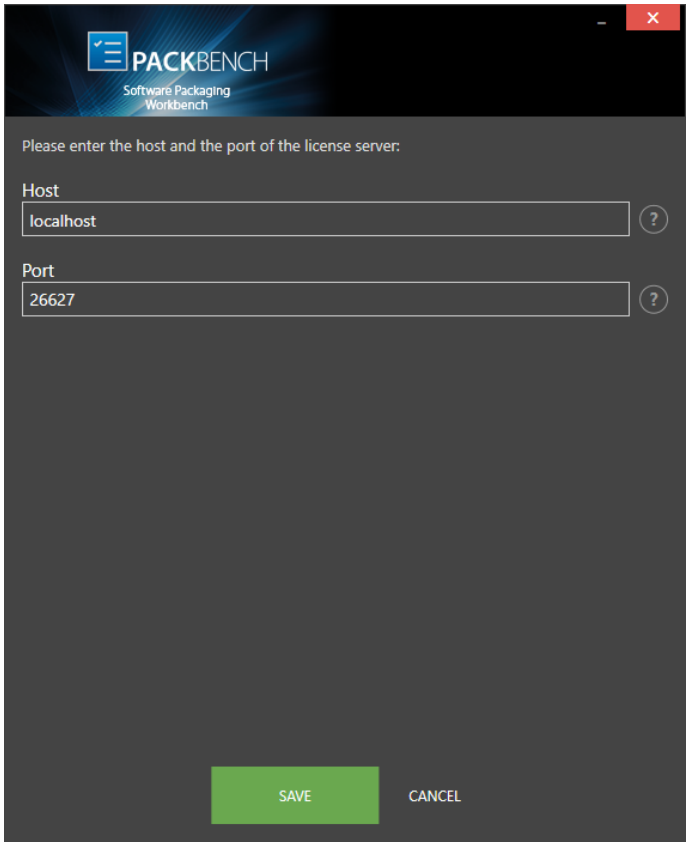
If there are any problems during the activation process, please contact our help desk for receiving assistance in activating PackBench.

# Floating License Server

PackBench can be activated using a local floating license server. This requires that the server component is installed (the installation is available separately from the product installer).

Once the server is configured, the following details are required from the server administrator:

- Server name or IP address
- Configured port (by default 26627)




Enter the required values and confirm them by clicking on the **SAVE** button. The server will be contacted once to verify the correctness of the data. If the server is not available at that time, an option will be presented to write the data anyway.

Once the connection details are saved, please restart the product to activate it using the floating license server.

# I do not have a license or order number

If neither a license or order number is available, then just simply register with Raynet to download an evaluation license for PackBench. This allows potential customers to test and work with PackBench before purchasing. Choosing **I don't have a license or order number** opens the Raynet website in the default browser, allowing potential customers to download an evaluation copy of PackBench.

# I want to take my activation back

Deactivating an existing license for PackBench may be required if the packaging machine used has to be switched. Whenever there is a scheduled migration, e. g. when a virtual machine is transferred in a way that affects the **Hardware ID**, or when a physical machine is no longer used for packaging purposes, deactivating the license is the right thing to do.

## To Deactivate a Licensed PackBench Installation

1. Launch PackBench and open the license and edition tab of the **about** area.

2. Click on the **Open the license wizard** button on the lower left hand side of the application window.

3. Use the option **I want to take my activation back...**

4. Enter the **order number** that was originally used to activate PackBench on the current machine. It was part of the resources and information material delivered during product purchase.

5. If required, adjust the user name already entered into the input field **User name**. The users who activate and deactivate an installation do not necessarily have to be the same.

6. Click on **DEACTIVATE NOW**.

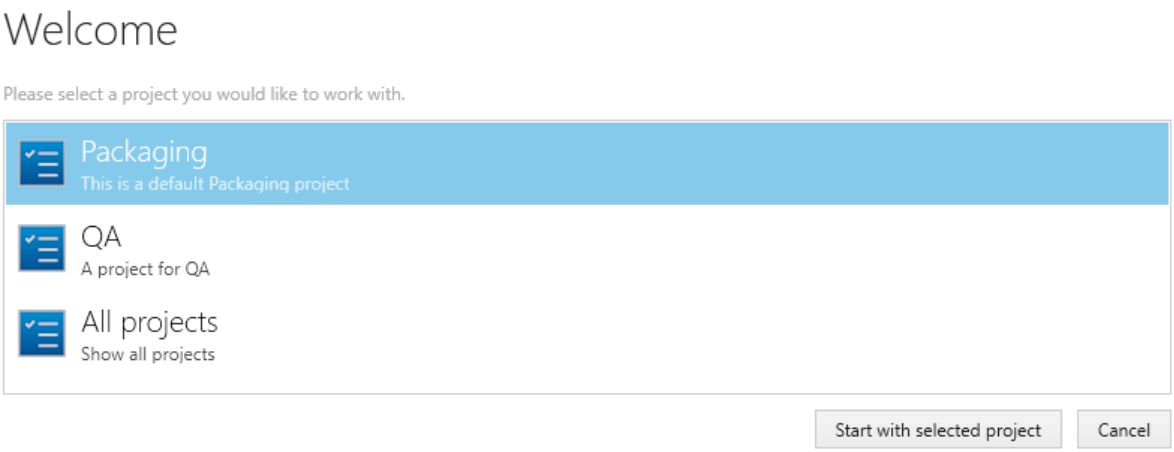
   The license wizard will connect to the Raynet licensing server and send the deactivation information. On success, the number of licenses available for activation, which are bound to the used order number, is incremented by one. With this new free license it is possible to activate any PackBench installation, on the current machine or any other.

## Troubleshooting

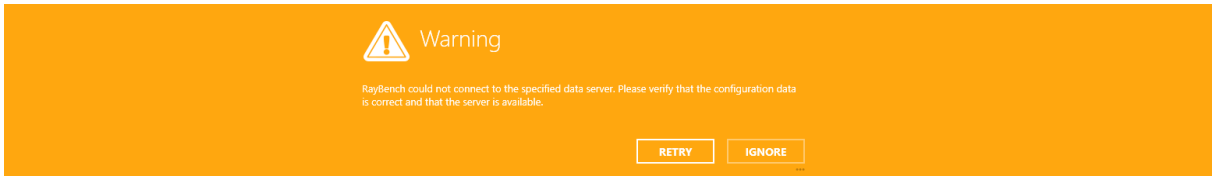
If any problems during this process occur, please contact our help desk for receiving assistance in deactivating PackBench.

# Basic Concepts

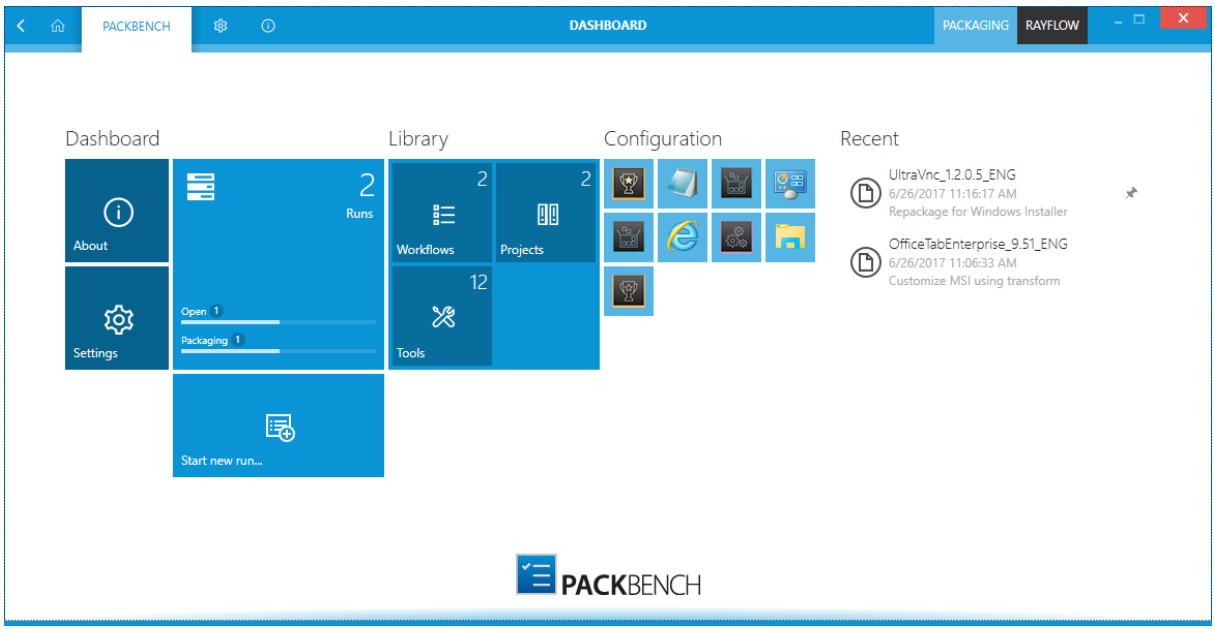
Once PackBench is installed on a machine, launching the program executable form the application installation directory (e.g., `C:\Program Files (x86)\PackBench\PackBench.exe`) invokes the PackBench **Welcome** screen. Select the target project and click on the **Start with selected project** button to open the project and continue to the *home screen*.




If the connection to the database is not available when PackBench is started, the following error message will appear. It is possible to launch PackBench in an offline mode by clicking on the **IGNORE** button. If this is done, only the **settings** and the **about** tile will be available and all other tiles will be grayed out. Furthermore, a red warning which states **could not connect** will be shown in the upper right corner where normally the selected project is shown. By clicking on this warning, PackBench will try to connect to the database again.



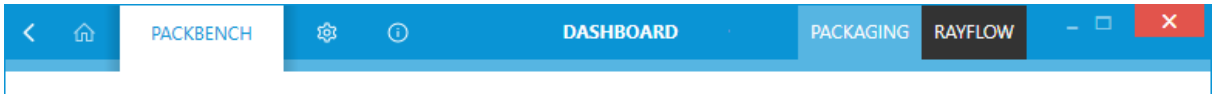

For those who have been previously working with other RaySuite applications, the layout of the home screen should be familiar, since it is a common interface type for all products of the RaySuite family.




Once PackBench is up and running, the application screen contains some basic areas which are always available, even though the actual content of an area varies from view to view.

# The Main Toolbar

Throughout PackBench the Main Toolbar is shown. Depending on the contents of the view shown, the Main Toolbar adds or removes menu items dynamically. The items shown below are always present on the Main Toolbar.




## Home

With this button it is possible to return to the *Home Screen*.

## View History

With a left-click on the arrow button, users navigate one step back within the history of recently opened views. Right-clicking the arrow displays the recently visited views and allows to return to a specific view from that list.
This view history is limited to those views without project relation or with relation to the currently opened project. Thus, returning back to a view is not possible if it is called for a project that is no longer opened.

## View Title

The view title specifies which content is currently shown as part of the active application context

module.

### Window Title

The window title displays the current scope of activity.

### PROJECT

The currently active project is shown here. Left-clicking on the project opens the **Welcome** screen where the active project can be chosen.

### RayFlow

Shows the name of RayFlow user currently connected within this instance. Read more about using PackBench with RayFlow in chapter *Working with RayFlow*.

### Standard Window Controls

The standard window controls allow to minimize, maximize, resize, and close the application window. The availability of each control follows the Windows schema for standard controls as known from any desktop application.

### Application Context

This shows the main application context status which is PackBench.

### SETTINGS

Loads the *Settings* view for PackBench.

### ABOUT

Contains information about the currently active product instance.

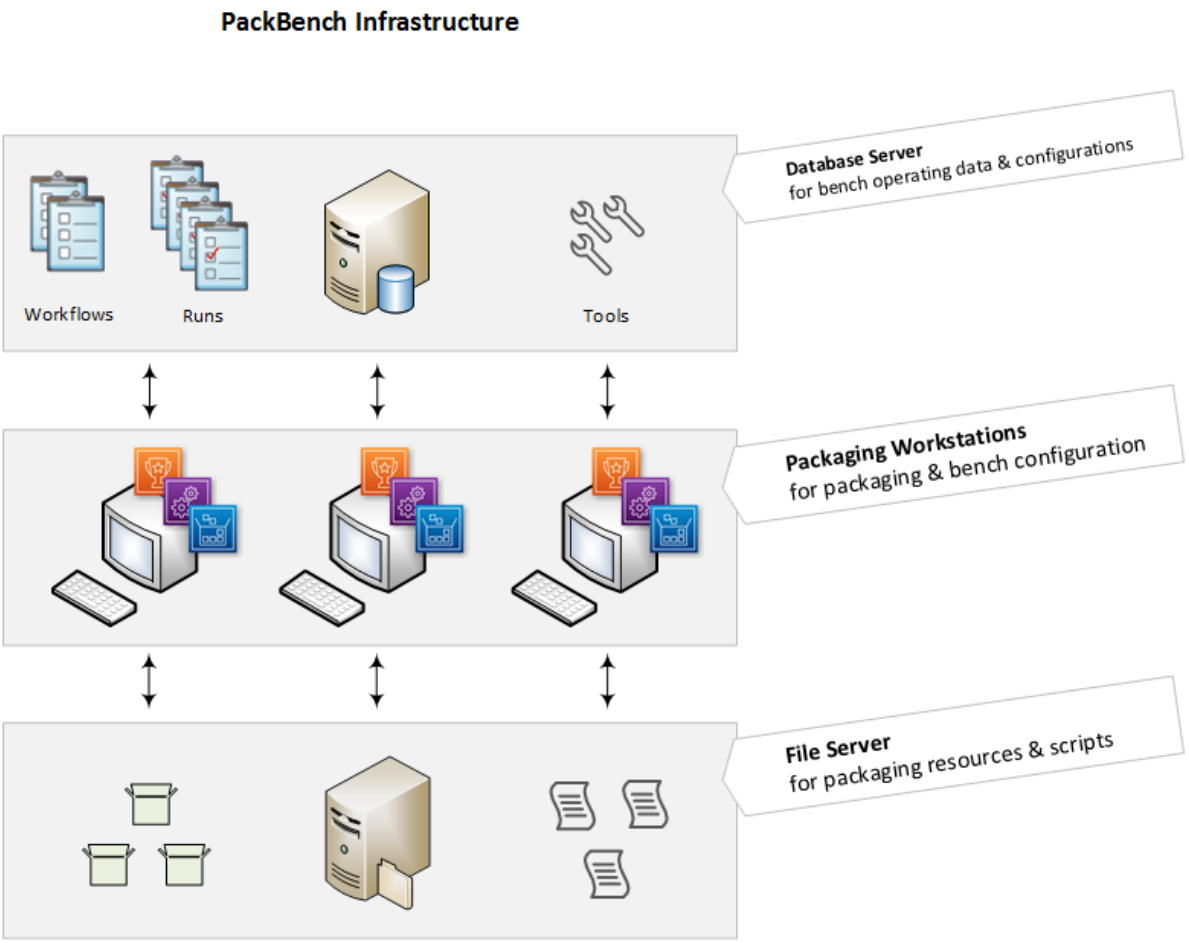
## The Content Area

The content area is the core for the actual application activity. According to the active application context it contains the dashboard (as shown in the screenshot at the beginning of this topic) or other functions of the application.

# Application Architecture

PackBench is architectured around a central database server and several clients which provide the user interface for the management and the execution of packaging related tasks.

A typical PackBench architecture will look like the infrastructure displayed in the following image.



# Standard Objects

There are some essential objects which need to be considered for successful PackBench utilization:

- **Projects**: A project is a logical unit which groups a dedicated set of workflows. A bench project may resemble a real project, such as an operating system migration project or typical workflows required for servicing a specific customer, business unit, or packaging team. Projects have no other function or technical dependencies. To learn more about projects in PackBench read the chapter *Configuring Projects*.

- **Workflows**: Workflows are templates that describe ordered sequences of tasks required for performing specific tasks. Each workflow template belongs to a specific project. Within the PackBench each workflow has a name and may optionally be equipped with a description and a help file. When a workflow is created, the user defines a set of variables (see below), which are used to save information that is either needed to perform the tasks, or are derived as results from task execution. Workflow tasks may either be executed in a strictly guided sequence from A to Z or in a free mode that allows arbitrary execution. Users may instance as many runs from each workflow template as required for the packaging factory operation. To learn more about workflows in PackBench read the chapter *Configuring Workflows*.

- **Runs**: When a packager starts working on a new task, he browses the project repository of workflow templates and instances a run from the template that fits the current order. Thus, a run is a copy of a specific workflow which is filled with information and executed step by step. Whilst a template is a generic description of what to do in which order, the runs execute and document the actual activity for a single packaging order. Runs are usually executed by one member of a packaging team, but may as well be taken over, for example when team organization requires a stand-in. Each step of a run may be due or done, whilst the run itself may have an overall status (e. g. open, complete, on hold), which may be assigned by the packager as he works his way through the single tasks. To learn more about runs in PackBench read the chapter *Workflow Runs*.

- **Variables**: Variables are required to be able to move from a general and more or less abstract workflow template to the actual instance values for a specific packaging task. Variables are defined as part of the workflow and filled with values during run executions. There are different types of variables differing according to their information content: free text or numeric input, checkbox (single or list), date selector, combobox, and file or folder path selector.
  Some of the variable types may be defined with optional input validation settings. Workflow tasks may make use of logic execution elements, so called tools (see below). These tools may utilize and modify variable values of a workflow run by referencing their specific variable name. There are default variables available for every workflow, e. g. PackageName, Type, Vendor. To learn more about variables in PackBench read the chapter *Editing Workflows*.

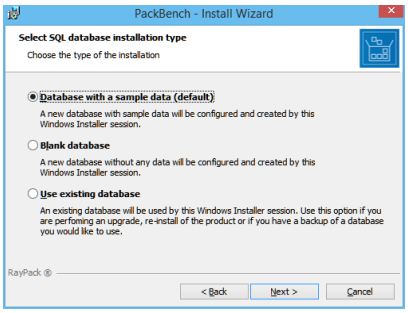
- **Tools**: PackBench intends to especially speed-up packaging related tasks. To do so, it does not only orchestrate tasks and documents their execution status, but at the same time improves their execution by integrating specialized tools. A tool can be any kind of executable, such as an external application or script. When a tool is configured for usage within the PackBench, it may be parameterized with values from variables, which are defined per workflow template.

Typical parameters would be the resource type and location, applicable templates, and the like. Tools help to make sure complex tasks are performed exactly as required, even when they are performed by less experienced packagers. Tools multiply and preserve valuable packaging know-how, and make it available for each job a packaging factory team takes over. Tools therefore improve the overall quality of the factory results as well. To learn more about tools in PackBench read the chapter *Configuring Tools*.

# Quick Start Guide

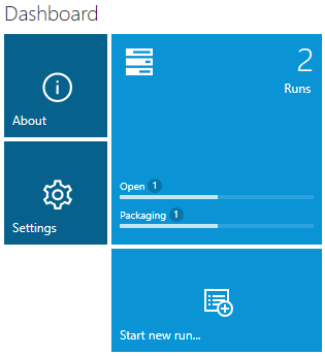
Follow these steps to discover PackBench after the product is already installed.

1. Install product using sample data (see *Product Installation* for more information) to get several out-of-the-box tools and example workflows.

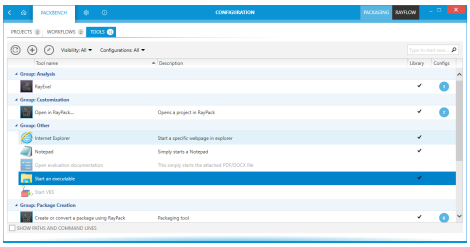



2. Start the PackBench client by double clicking on `PackBench.exe`.

3. The home screen will be shown.




4. Go to *Configuration > Tools* and ensure all necessary tools are present. If any custom tools/scripts are needed, create them using instructions from chapter *Creating Tools*. It is also possible to tweak existing tools as described in *Editing Tools*.



5. Go to *Configuration > Workflow* and review the default workflows. *Create a new workflow* that matches processes in your organization, or edit existing one. It is possible to *clone existing* workflows to work on a fresh copy.
   a. Define *custom variables* that should be used to describe the process (for example, package

revision, packager name, additional details etc.).

b. *Link tools* from the library to particular steps in the workflow. Certain parameters for workflow-specific behaviors can be overridden. The *variables* can be used to do the work using input from user.

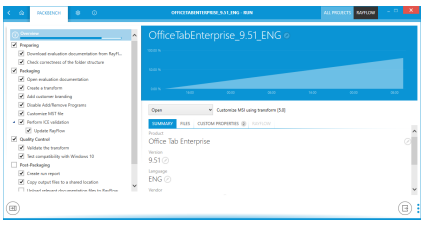
6. Go to *Configuration > Projects* and adjust the details of the current project to satisfy certain naming conventions and general settings requirements.
   a. If there is the intention to use RayFlow and RayFlow URL has not been configured during installation, go to *project properties* to the RayFlow tab and set up a connection between RayFlow and PackBench projects.

7. Once the configuration is done, typically the configuration will be rarely visited (unless there are adjustments to existing processes). However, bear in mind that after starting a run of a workflow, it copies all workflow details, so that any future changes are not affecting already running workflows.

8. *Start a new run* to execute the workflow.
   a. Sources can be specified on installation time. As a source, a single file, or a single folder, or a task/file from RayFlow can be selected.
   b. If no sources should be specified right now, this step can be skipped and it can be *done later* (however, RayFlow assignment can be only configured from the *New Run* wizard).
   c. Provide any additional details for which will be asked. Some of them may be already inserted if the package sources have been selected.

9. Once the wizard is finished, a basic folder structure in projects folder will be automatically created.

10. While in run, it is possible work through the checklist, execute associated tasks and tools, change variables, generate reports etc.




11. Any files or data are saved while going through steps.

12. Once all steps are executed, the work is done and the result are files and information created according to the process guidelines.

13. If RayFlow has been configured, upload result files and documentation back to RayFlow.

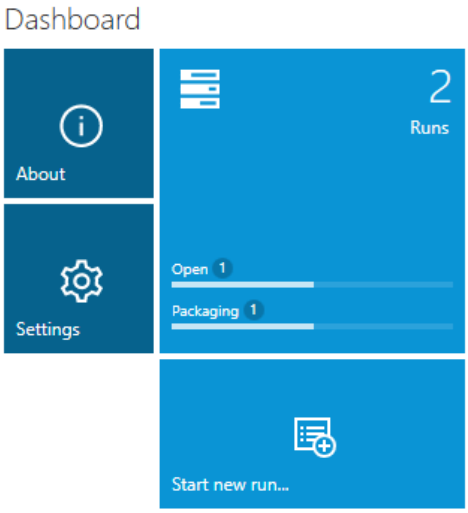
14. Refer to *advanced topics* for more detailed insight and to learn how to tweak PackBench even more.

# Home Screen

The Home Screen of PackBench contains a Dashboard, a recent, a tools and a configuration section. Whenever the application is launched without a specific file trigger, this view is the starting point of a new PackBench session. Just like other RaySuite products, the PackBench Dashboard contains tiles which link to the key features and areas of this module.

## Dashboard

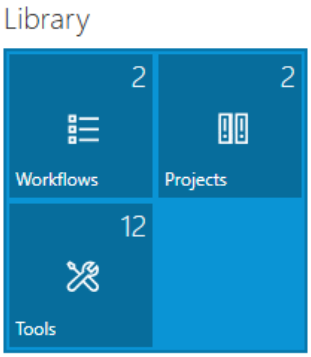
The dashboard of PackBench contains four different tiles and each of the tiles on the dashboard opens a specific view or starts a wizard:




- **About:** Link to various built-in and online resources is available under the about view.
- **Settings:** After installation of PackBench, a user should visit the settings view to verify if the connection to the SQL database is working.
- **Runs**: With this tile the **WORKFLOW RUNS** view is opened. An overview of all runs can be found here. The number of the runs in the project and a diagram of their states is shown on the tile.
- **Start new run...:** Clicking on this tile opens the **START NEW RUN** wizard.

# Configuration

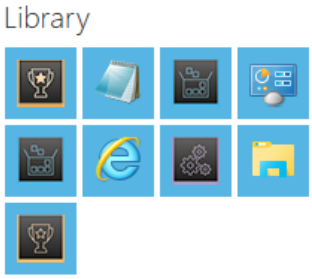
The configuration part of the Home Screen contains the **configuration** tile. This tile consists of an overview which counts the workflows, projects, and tools currently available in the PackBench instance. Clicking on the **configuration** tile will open the **CONFIGURATION** view.




For more on how to configure PackBench and the **CONFIGURATION** view refer to *this section*.

# Library

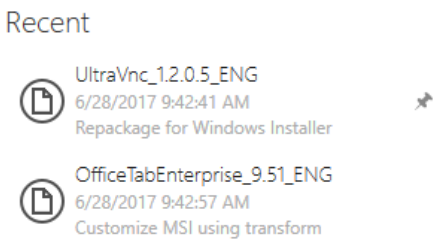
The library consists of several small tiles, each representing a different tool:




By pressing an icon, the tool will be automatically started. It is also possible to hover the mouse over the icon to see the name and description of underlying tool.

It can be configured which tools are shown here (and what entry points they trigger) by changing the value of VISIBLE/HIDDEN flag in the configuration section. Hidden tools are never shown in the library.

# Recent




The **Recent** section of the home screen shows a list of top five most recent runs. Selecting an item from this list will open it directly. You can pin and unpin items to the list by using a small pin icon shown next to each item.

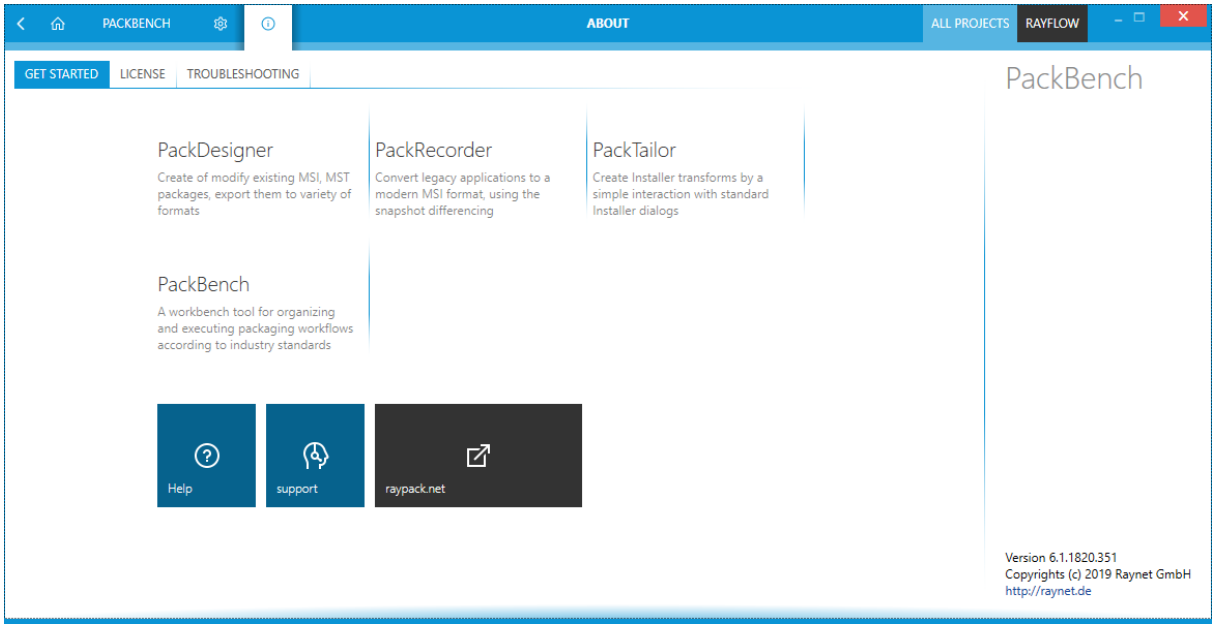
# About

Choosing the **About** tile from the *Home Screen* displays the about area. It contains the get started view, providing information about the individual tools of PackBench, Samples and links to various resources. Additional supportive views regarding *license and edition* as well as *troubleshooting*.

An alternative access to the about section is provided by clicking the about tab displayed within the main toolbar, which is present within all PackBench editor views.

## Get Started

Choosing the **get started** tab from the *Home Screen* reveals this view in the about section. It contains information about the individual tools of PackBench, Samples, and links to various resources. Additional supportive views regarding *license and edition*, as well as, *troubleshooting* are available by clicking on the view tabs.



# Meet RayPack

These are the five main tools that RayPack comprises of:

- PackDesigner
- PackRecorder
- PackTailor
- PackBench

The PackBench client is separated from the set and can be started as a standalone application. To use other modules, start main RayPack executable.

## Get Started

This opens the get started help file, which includes tips & tricks how to configure and use PackBench.

## Support

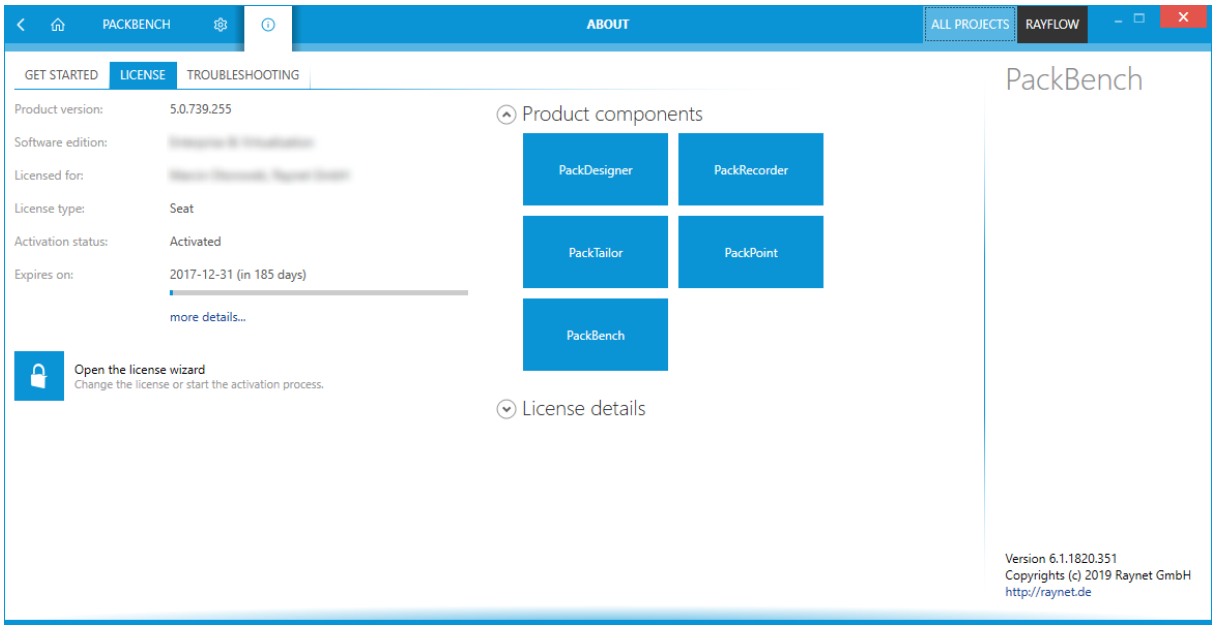
Opens the web site for Raynet product support.

## Www.raypack.net

RayPack on the web.

# License and Edition

The *about* area contains the License and Edition view, providing all usage relevant license information about the current product instance. Additional supportive views regarding *get started* guidance, as well as, *troubleshooting* are available by clicking on the view tabs.




Here, a variety of information is shown regarding the current version of PackBench, license information and the installed features that are currently licensed. License information can be entered using the license wizard by choosing *Open the license wizard*.

## PackBench Components

Depending on the license that you have purchased, the various add-on modules that are available for that particular license are displayed in the **"PackBench components"** section.

## License Details

Depending on the license that has been purchased, the various features that are available for that particular license are displayed when expanding the **"License details"** section.

# License File Management

After the activation of PackBench, an encrypted license file (`.rsl`) is generated and saved in the root folder where the program has been installed.

Subsequent runs of the application will require no internet connection / activation as long as a valid license file is present in any of the following folders:

1. The main installation folder (for example `C:\Program Files (x86)\RayPack\PackBench`).
   - When running a portable copy, the path from where the main executable has been started applies.
2. `\Raynet\Licenses` folder in Common 32-bit Program Files directory (for example `C:\Program Files (x86)\Common Files\Raynet\Licenses`).
3. `\Raynet\Licenses` folder in Application Data directory (for example `C:\Users\Administrator\AppData\Roaming\Raynet\Licenses`).
4. `\Raynet\Licenses` folder in Program Data directory (for example `C:\ProgramData\Raynet\Licenses`).

The required extension of a license file name is `.rsl`.

The license file is portable and will not be removed when PackBench is uninstalled. A back-up of the license or a copy on the stick in the portable installation location allows to start PackBench, given that the machine is the same as used to activate the product.

## Resolving Multiple License Files

In case more than one license is found, the following logic is applied

1. The order in which folders are probed is equal to the standard probing described in the above section.
2. Should two or more license files be found in the same level, the one that is active and valid for the current machine wins. If more than one license is active and valid, the first one wins (note: the order in which license files from a folder are probed may be not deterministic)
3. Should no license in the same level be valid and active, the next folder is processed
4. If no valid and active license has been found in any of the probed folders, the program will not start and the activation wizard will show up.

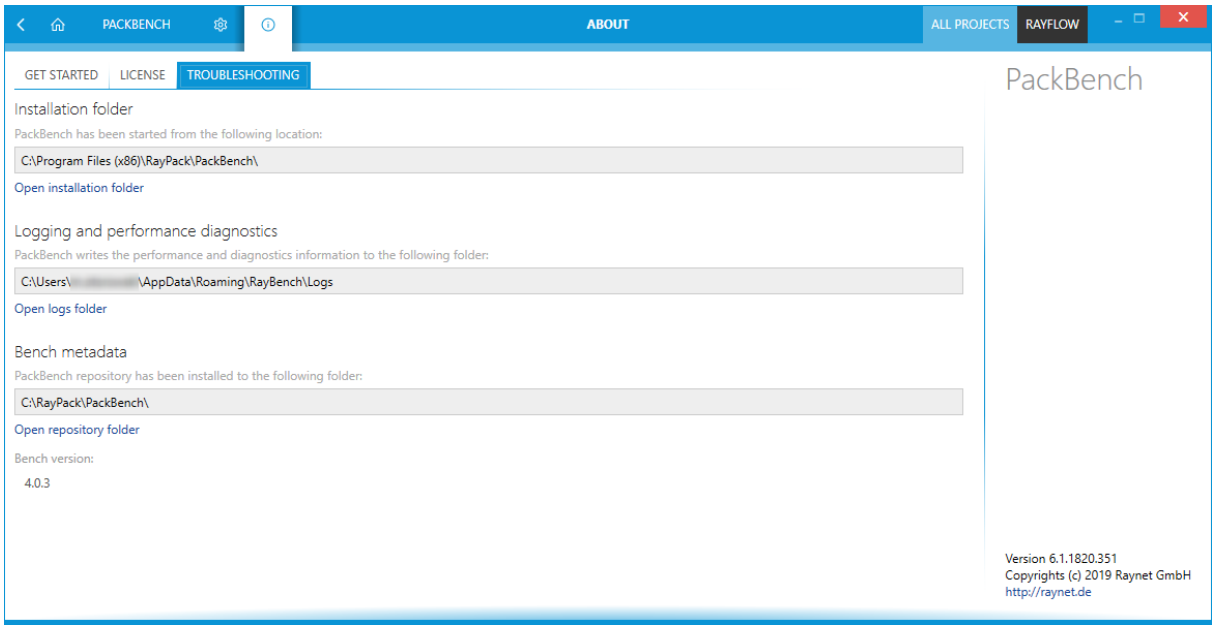
## Logging and Troubleshooting

Detailed information about the order and which licenses are applied on start-up can be found in the PackBench logs. Refer to the Troubleshooting section for more information.

# Troubleshooting

The *about* area contains the Troubleshooting view, providing handy information about the crucial system paths defined for the current product instance. Additional supportive views regarding *get started* guidance, as well as, *license and edition* details are available by clicking on the view tabs.

Any information displayed within this tab is read-only. It is provided to ensure transparency about the current instance settings, which is vital for proper troubleshooting measures in case of support relevant issues.




## Installation Folder

This path shows where the currently running instance of PackBench is located. Click on the **Open installation folder** link to open a windows explorer instance at the displayed location for further instance resource review.

# Logging and Performance Diagnostics

PackBench is by default configured to write log files with information about each product work session. A new log file will be generated for every launch of the current application instance. These log files contain vital information for any troubleshooting or help desk measure, such as environment information about the physical machine PackBench resides on, the steps performed during a session, and exception details thrown by the application in case of operational issues. Please make sure that the target directory displayed here provides sufficient disk space and is accessible for the currently logged in user. Click on the **Open logs folder** link to open a windows explorer instance at the displayed location for further instance resource review.

# Bench Metadata

The path to the PackBench repository and the version of the current PackBench installation will be shown here. Click on the on the **Open repository folder** link to open a windows explorer instance at the displayed location for further instance resource review.

# Settings

Choosing the settings button from the *Home Screen* allows users to make changes to the configuration.

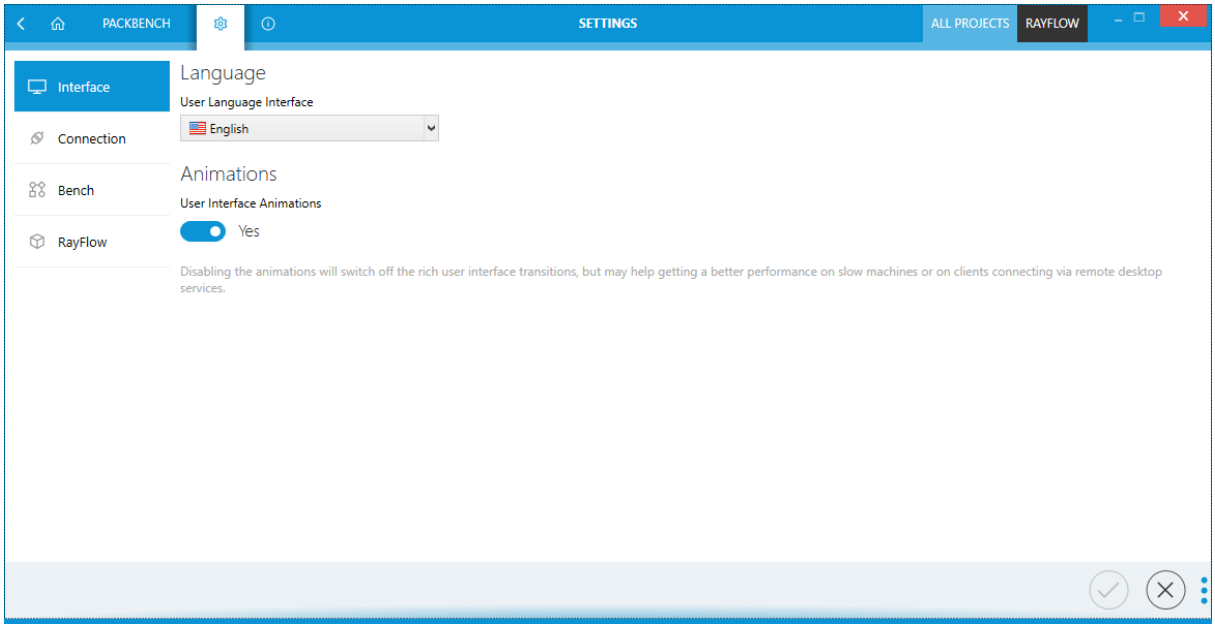
An alternative access to the settings section is provided by clicking the settings tab displayed within the main toolbar, which is present within all PackBench editor views.

## Interface

This section of the **SETTINGS** contains the options to define the language that is used throughout PackBench and the option to switch the animations on and off.

### Language

The language option enables users to choose the language that should be used for the PackBench application. the language is chosen by selecting one of the options that can be found in the drop-down menu which is available under the **User language interface** option.



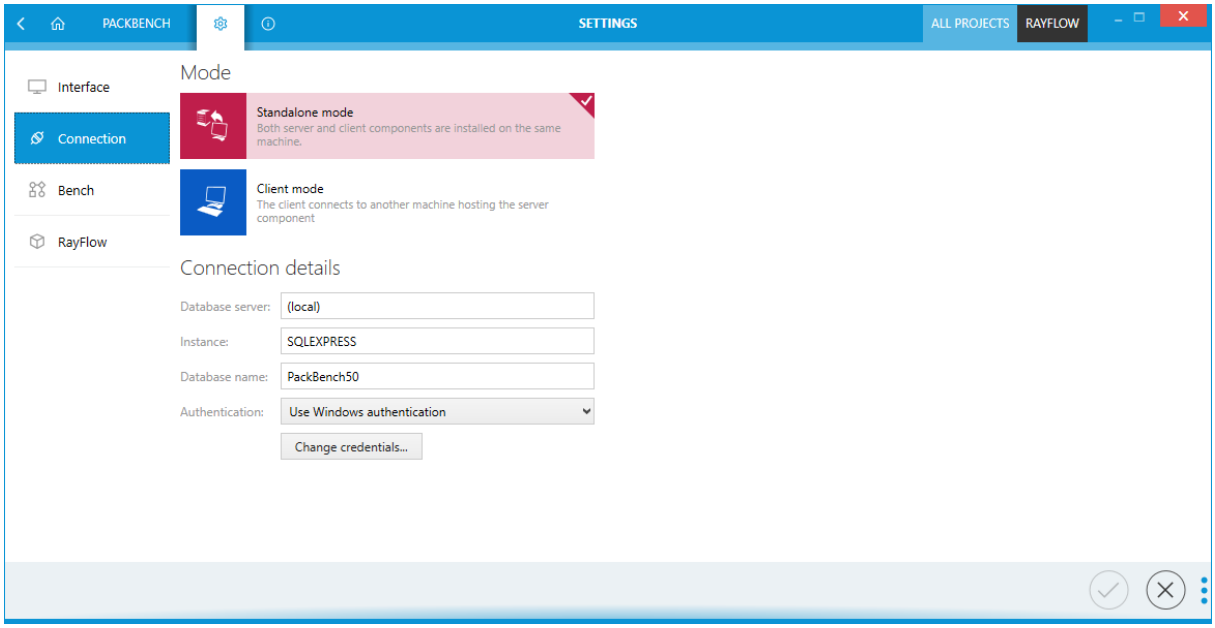

Currently there is only one language available for the user interface of PackBench: The available language is English.

### Animations

When PackBench is going to be used via remote connections, it may save precious bandwidth and performance if the user interface animations are deactivated. They are active by default, but may be deactivated by setting the slide toggle **Enable user interface animations** to **No**.

# Connection

This section of the **SETTINGS** contains information on the connection to the PackBench server.




## Mode

The following modes are available for PackBench:

- **Standalone mode**: Choose the Standalone mode if PackBench has been installed as *Standalone*. This is the case if both, host and client are installed on the same physical machine, therefore the client does not need to connect to another physical machine on which the server is installed.
- **Client mode**: Choose the Client mode if host and client have been installed on different machines. The client will need to connect to the machine which is hosting the server.

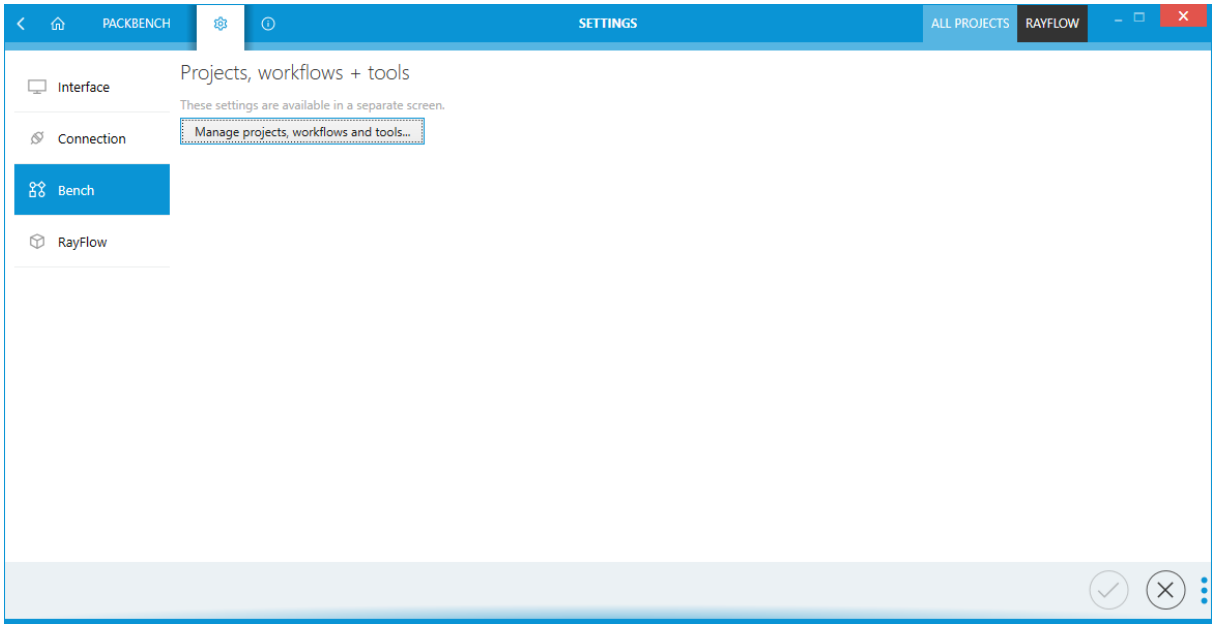
## Connection Details

This is where the connection to the machine hosting the server is configured. There are two options available.

- **Server**: The SQL server needs to be defined here, if not already defined during the installation process. Please specify the server together with an instance (if an instance is required). The server can be specified either by its name or by its IP address.
- **Port**: The port for the connection has to be entered here. Please ensure, that port has to be identically with the one that has been used during the set up of the PackBench server. The port can either be changed by manually typing in a new port or by using the arrow buttons to count up or down.

# Bench

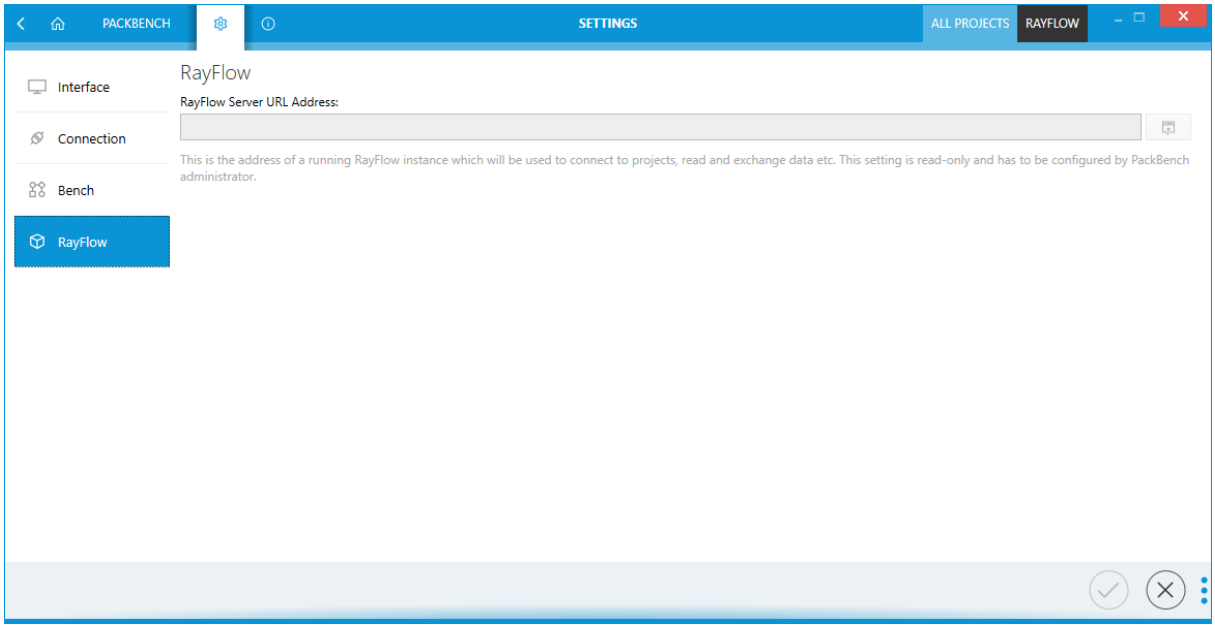
This is where the settings of the projects, workflows, and tools can be found.




Clicking on the **Manage projects, workflows and tools...** button will open the **CONFIGURATION** view, which can also be opened by clicking on the config tile on the **dashboard**. For more information on the configuration of projects, workflows, and tools refer to the *Configuration* chapter.

# RayFlow

This is where the RayFlow server setting can be verified.



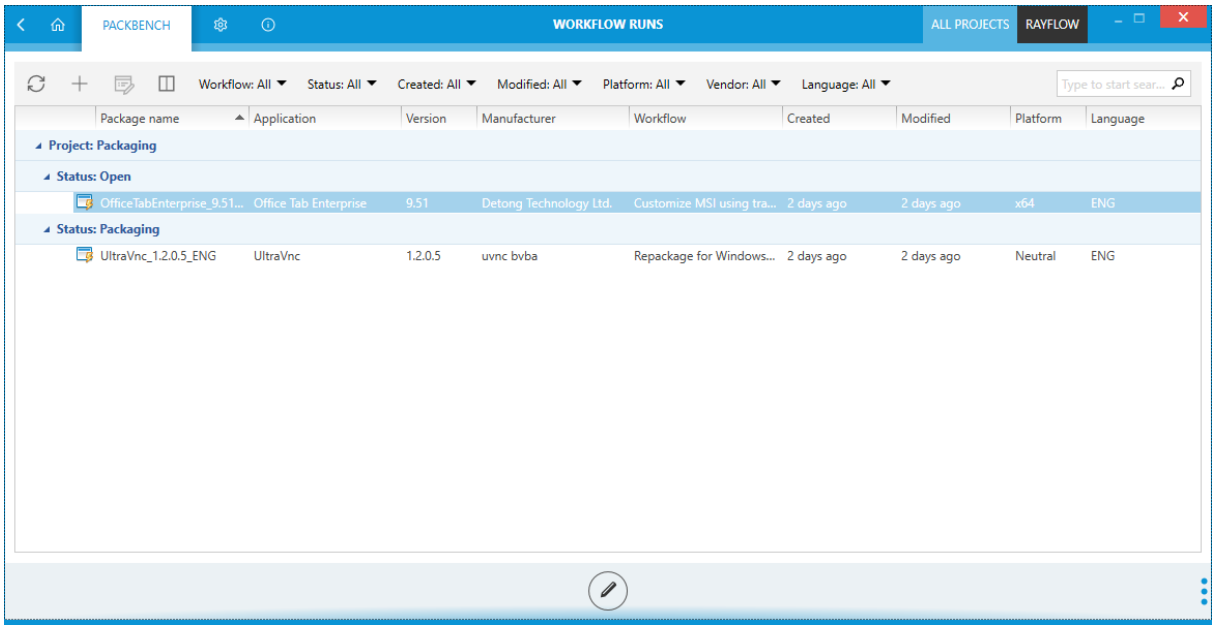

The RayFlow URL is a read-only setting. It is configured per-instance during the installation of the product. Advanced topic *Changing RayFlow server* shows steps to be done in order to change the URL manually.

# Workflow Runs

This chapter explains how to create runs and how to work with runs inside of PackBench.

## Managing Runs

The **WORKFLOW RUNS** overview is opened by clicking on the **runs** tile on the Dashboard.




The **WORKFLOW RUNS** view shows all the runs that are part of the selected project. It is divided in a content area containing the runs and an action bar which is located on top of the content area.

### Refresh this View Button

The **Refresh this view** button is located in the action bar and can be used to refresh the view.

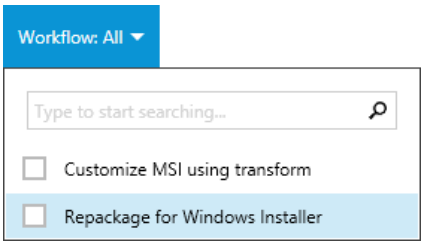
### Start New Run Button

The **Start new run** button is located in the action bar and can be used to open the **START NEW RUN** wizard and create a new run. For more information on this refer to the *Starting New Runs* section of this document.

## Edit Currently Selected Run Button

The **Edit currently selected run** button can be used to open the selected run in the details view. For more information on this refer to the *Completing Runs* section of this document.

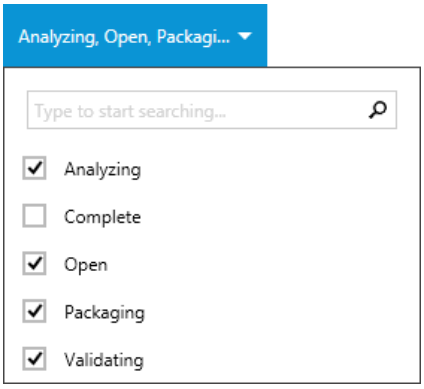
## Workflow




The **Workflow** button contains a dropdown menu with an integrated search function. A list of all workflows of the runs which are currently shown in the content area will appear. This list can be searched by using the keyword search field integrated into the dropdown menu to search for workflows. This search field works like the keyword search field in the tab. To learn how to use the keyword search field continue with this *section*.

After selecting workflows from the dropdown menu only runs which use the selected workflows will be shown in the content area. Workflows are chosen by enabling or disabling the checkbox related to the project. If no workflow is selected, all workflows will be shown. By default, no workflow is selected. The title of the option will show the selected worklfows.

## Status

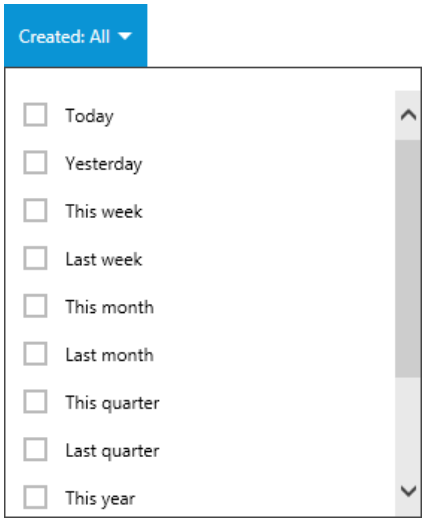



The **Status** button contains a dropdown menu with an integrated search function. A list of all status of the runs which are currently shown in the content area will appear. This list can be searched by using the keyword search field integrated into the dropdown menu to search for status. This search field functions like the keyword search field in the tab. To learn how to use the

keyword search field continue with this *section*.

After selecting a status from the dropdown menu only runs which are currently in the selected status will be shown in the content area. Status are chosen by enabling or disabling the checkbox related to the project. If no status is selected, all status will be shown. By default, only non-finished statuses are active. For example: **Analyzing**, **Open**, **Packaging**, **Testing**, and **Validating** are selected as status in the above screenshot. The title of the option will show the selected status. For more information look at this *section*.

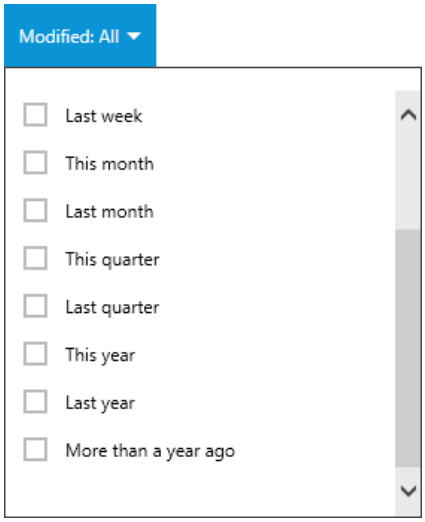
## Created




The **Created** button contains a dropdown menu which can be used to define which runs will be shown in the **WORFLOW RUNS** view. Available options in this menu are **Today**, **Yesterday**, **This week**, **Last week**, **This month**, **Last month**, **This quarter**, **Last quarter**, **This year**, **Last year**, and **More than a year ago**. Enable or disable an option by using the checkbox related to the option.

After selecting an option from the dropdown menu only runs which match the selected option will be shown in the content area. Options are chosen by enabling or disabling the checkbox related to the option. If no option is selected, all options will be shown. By default, no option is selected. The title of the button will be changed to the selected options.

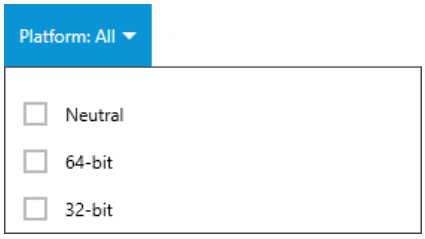
## Modified




The **Modified** button contains a dropdown menu which can be used to define which runs will be shown in the **WORFLOW RUNS** view. Available options in this menu are **Today**, **Yesterday**, **This week**, **Last week**, **This month**, **Last month**, **This quarter**, **Last quarter**, **This year**, **Last year**, and **More than a year ago**. Enable or disable an option by using the checkbox related to the option.

After selecting an option from the dropdown menu only runs which match the selected option will be shown in the content area. Options are chosen by enabling or disabling the checkbox related to the option. If no option is selected, all options will be shown. By default, no option is selected. The title of the button will be changed to the selected options.

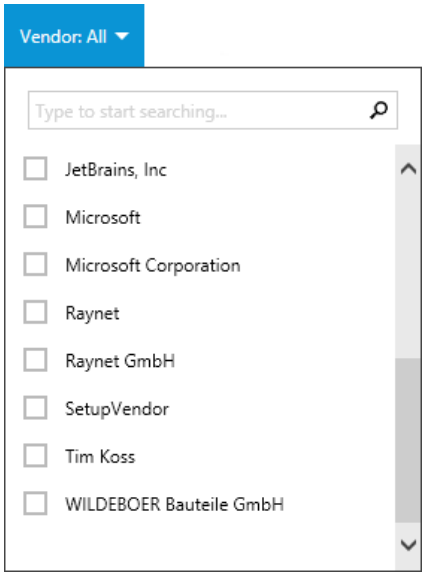
## Platform




The **Platform** button contains a dropdown menu which can be used to define which runs will be shown in the **WORFLOW RUNS** view. Available options in this menu are **Neutral**, **64-bit**, and **32-bit**. Enable or disable an option by using the checkbox related to the option.

After selecting an option from the dropdown menu only runs which use the selected option will be shown in the content area. Options are chosen by enabling or disabling the checkbox related to the project. If no option is selected, all options will be shown. By default, no option is selected. The title of the button will be changed to the selected options.

## Vendor




The **Vendor** button contains a dropdown menu with an integrated search function. A list of all vendors of the runs which are currently shown in the content area will appear. This list can be searched by using the keyword search field integrated into the dropdown menu to search for vendors. This search field functions like the keyword search field in the tab. To learn how to use the keyword search field continue with this *section*.

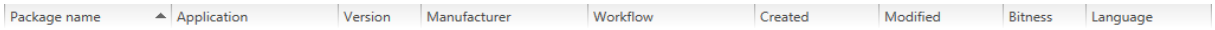
After selecting a vendor from the dropdown menu only runs which use the selected vendor will be shown in the content area. Vendors are chosen by enabling or disabling the checkbox related to the project. If no vendor is selected, all status will be shown. By default, no vendor is selected. The title of the option will show the selected vendor.

## Keyword Search Field

The keyword search field can be used to filter the runs. Type the keyword into the search field. The runs will be filtered for the keyword while typing.

For example, type "sample" into the search field an only runs which contain the word "sample" in any column will be shown.

## Sorting




It is possible to sort the content by column. Select the column and click on it to sort the runs alphabetically. An arrow will appear next to the selected column header. To change the sort order simply click on it again. The arrow will indicate the order by either pointing upwards or downwards.

The columns themselves can be interchanged by drag and drop. Simply drag the column header

and drop it at the target position.

### Switch to Details View Button

The **Switch to details view** button is located on the swipe-bar and can be used to switch to the details view. To learn more about the details view continue to the next chapter.

## Working with many Projects

If no default project is selected, the workflow view contains additional elements:

- Ability to filter items by project name
- Grouping by project name

All other functions remain the same.

# Starting Runs

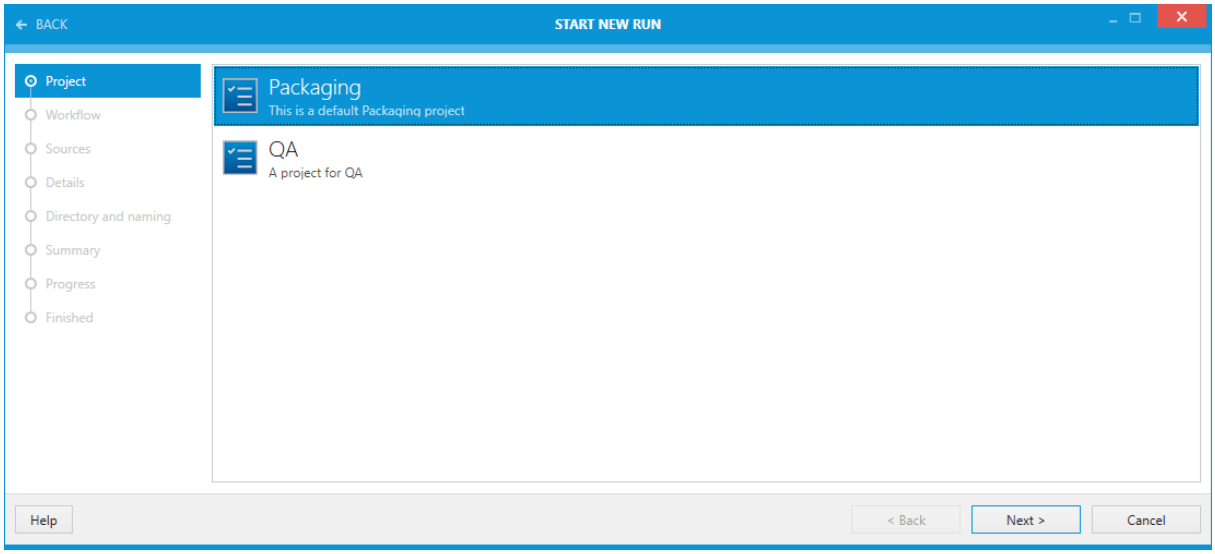
To start a new run open the **START A NEW RUN** wizard by either clicking on the **start new run** tile on the Dashboard or by clicking on the **Start new run** button in the **WORKFLOW RUNS** view.

## Using the New Run Wizard

The **START NEW RUN** wizard is the tool which is used to create new runs in PackBench.

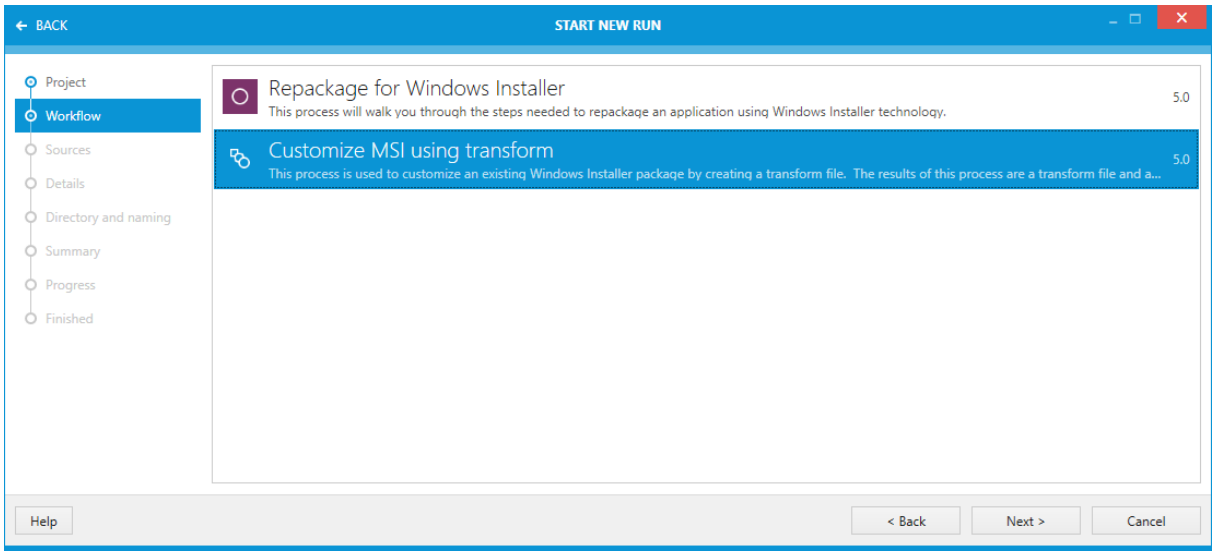
### Project Page

**Note:**
This option is only available if there is no project which is currently selected as a default one. Otherwise, the selected project is used implicitly.




On the **Project** page the project for the run is chosen. The selection determines the availability of workflows in the next screen.
Click on the **Next >** button to proceed to the **Workflow** page.

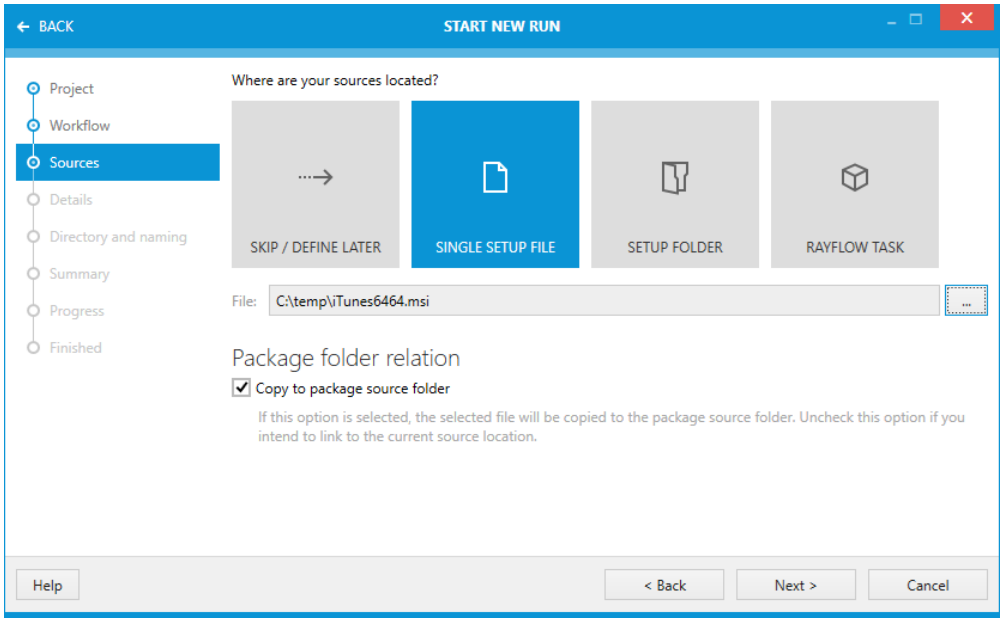
## Workflow Page




On the **Workflow** page the workflow for the run is chosen. In this example, the available workflows are **Repackage for Windows Installer** and **Customize MSI using transform**.

Select the target workflow and click on the **Next >** button to proceed to the **Sources** page.

## Sources Page




On the Sources page the sources for the new run are selected. Select the type of source using the

radio buttons. The following options are available:

- **SKIP / DEFINE LATER**: If this option is chosen, no file or folder is selected and all configuration in this step can be skipped.

- **SINGLE SETUP FILE**: If this option is chosen, the **Open** dialog appears. Select the target source file and click on the **Open** button. After this, the file can still be changed by using the browse button [...] to browse to the target file. Under the **Project folder relation** option, choose if the selected sources should be copied into the project or if the current location of the sources should be linked. If the location should be linked, disable the checkbox. If the sources should be copied into the project, ensure that the checkbox is enabled. By default, the sources are copied into the project.

**Be aware:**
There is a special handling for this option if you select an `.msi` file as a source. Even though the option says that PackBench determines which other files are required (for example external uncompressed files, CAB archives etc.) and copies them together with the selected source file if some sources are missing, the errors are silently skipped and only shown in the product's log.

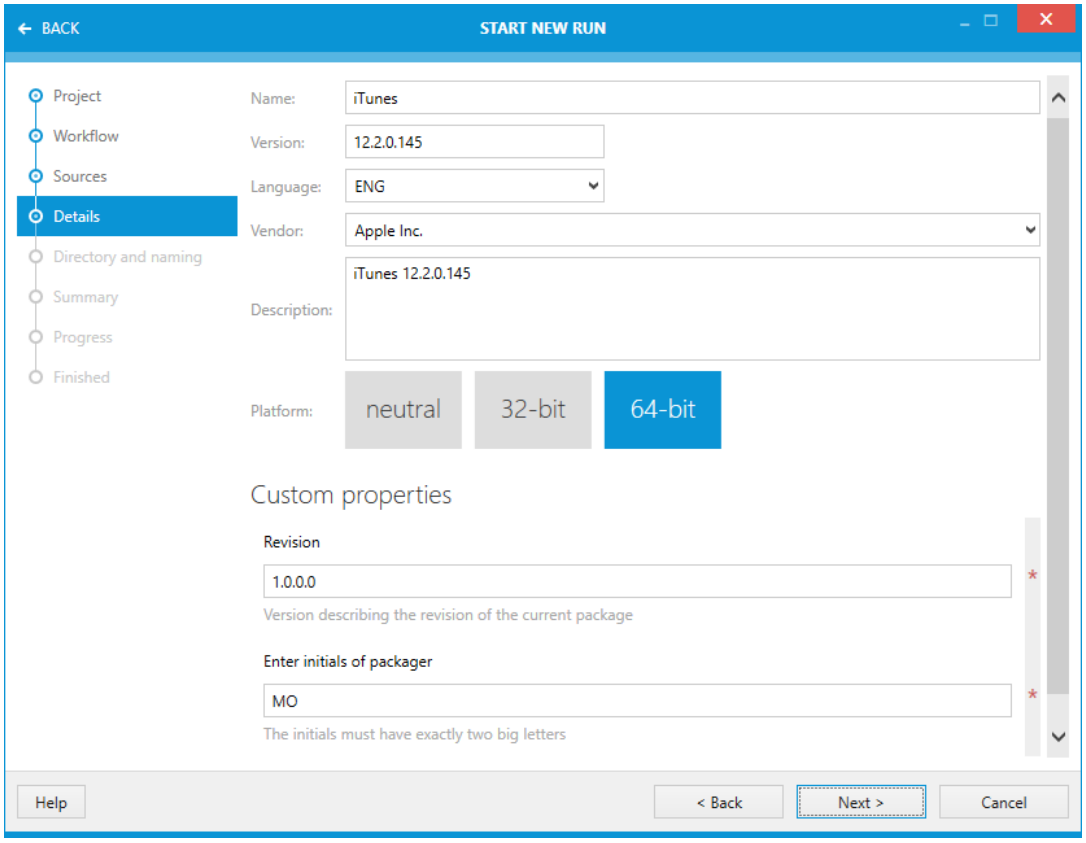
- **SETUP FOLDER**: If this option is chosen, the **Select Folder** dialog appears. Select the target source folder and click on the **Select Folder** button. After this, the folder can still be changed by using the browse button [...] to browse to the new target folder. Select the executable or file which should be used by default to trigger the installation under the **Entry Point** option. Then, under the **Project folder relation** option, choose if the selected sources should be copied into the project or if the current location of the sources should be linked. If the location should be linked, disable the checkbox. If the sources should be copied into the project, ensure that the checkbox is enabled. By default, the sources are copied into the project.

- **RAYFLOW TASK**: With this option a task or a file within a task can be selected directly from RayFlow. More information about working with RayFlow can be found in chapter *Working with RayFlow*.

**Note:**
This option is only available if the currently selected project has been configured to use RayFlow. Refer to chapter *RayFlow* for more information about setting up a RayFlow connection.

Click on the **Next >** button to proceed to the **Details** page.

## Details Page




On the **Details** page, the details of the new run are defined. The following options are available on this page. Most of the fields will be filled in automatically, based on the package that has been chosen. All values can also be configured manually:

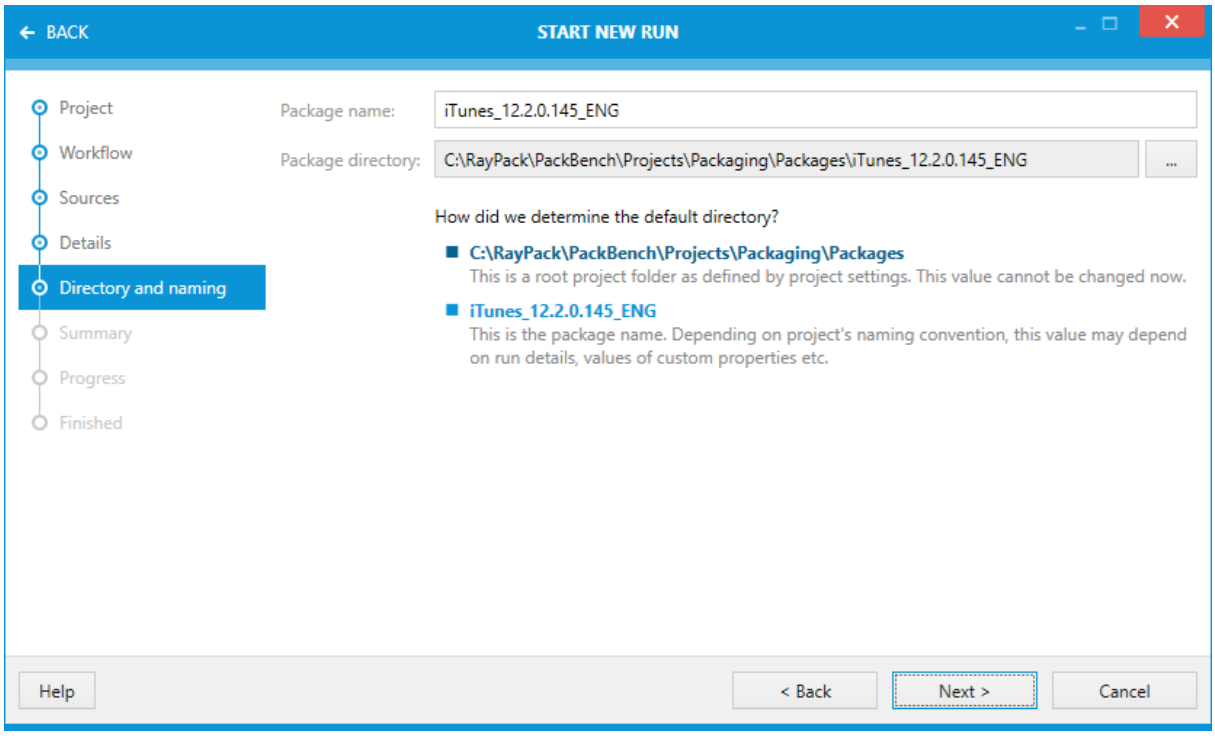
- **Name**: This is the name of the application.
- **Version**: This is the product version.
- **Language**: This is the language of the product.
- **Vendor**: This is the vendor/manufacturer of the product.
- **Description**: A description for the package can be entered here. For the default value, see the screenshot above.
- **Platform**: This is where the target platform for the package is configured. There are three options available: neutral (the package can be used on both architectures), 32-bit (only), or 64-bit (only).

Furthermore, the Custom properties are configured on this page. The content of the **Custom properties** page varies, depending on the workflow that has been selected for the run. The page shown above shows the Custom properties for the **Repackage for Windows Installer** workflow used in this example. Mandatory fields will be marked with a red * and fields with invalid entries

will be highlighted in red and an error message will describe why the value is invalid.

When all necessary data has been entered click on the **Next >** button to proceed to the **Directory and naming** page.

## Directory and Naming Page

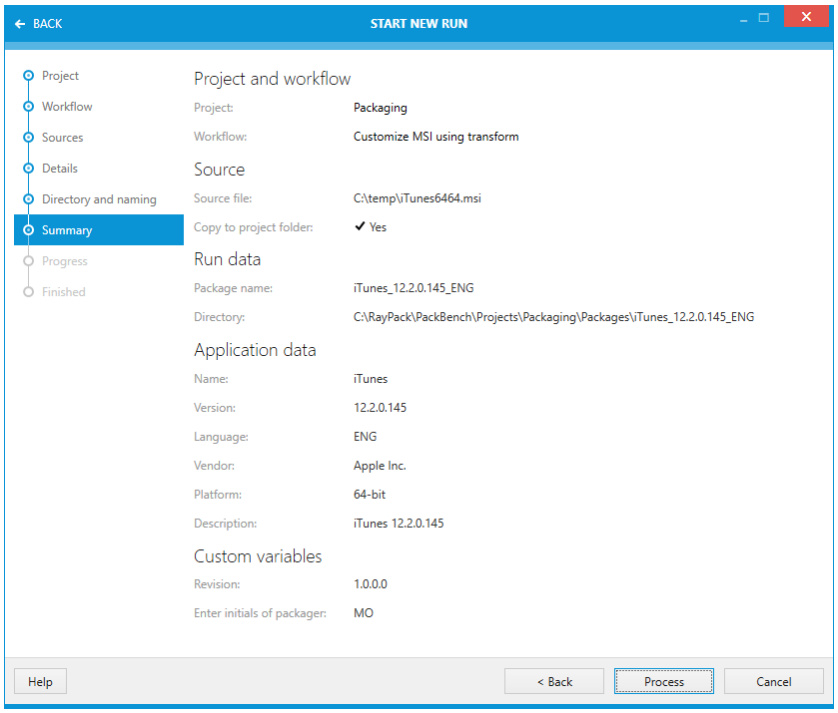



On the Directory and naming page, the directory in which the run is stored is defined under the **Store project in the following directory** option. It can be changed by opening the **Select Folder** window through clicking on the Browse button [...]. In the **Select Folder** window, browse to the target folder and click on the **Select Folder** button.

**Note:**
If the option to copy the sources to the project folder has been chosen and a source file or folder has been selected, the required files will be copied into a subfolder of the project as defined in the *Advanced tab* of the project properties.

When all necessary data has been entered click on the **Next >** button to proceed to the **Summary** page.

## Summary Page




The summary shows all configurations that have been made for the new run. Verify the entries and click on the **Process** button for the new run to be created.

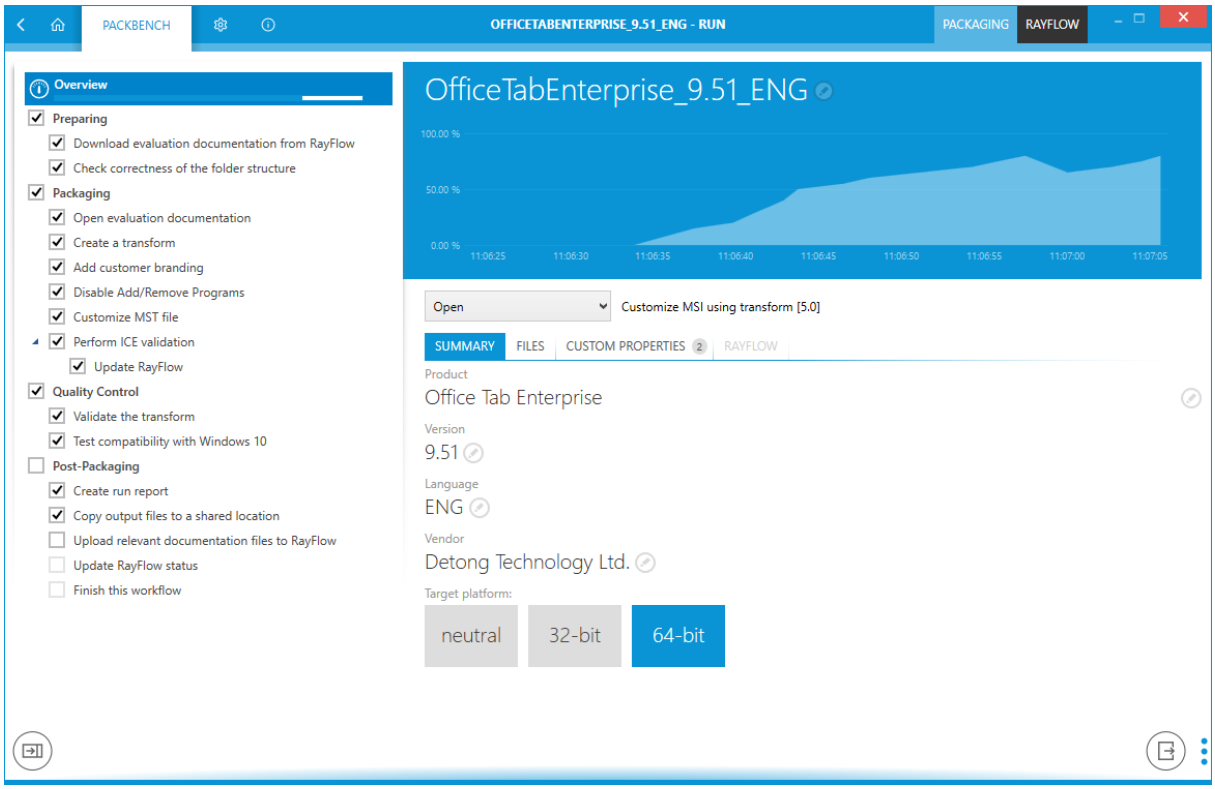
## Progress Page and Finished Page

On the **Progress** page the progress of the creation of the run is shown. When the creation process is finished the page will automatically change to the **Finished** page.

The run has now successfully been created and the wizard can be closed by using the **Finish** button. The new run will be opened automatically.

# Working with Runs

Runs are completed in the details view.




The view consists of three panels:

- A tree list of all tasks available in the workflow
- A contextual panel, showing the details of the currently selected item.
- A sidebar (hidden by default) showing open runs and library of tools

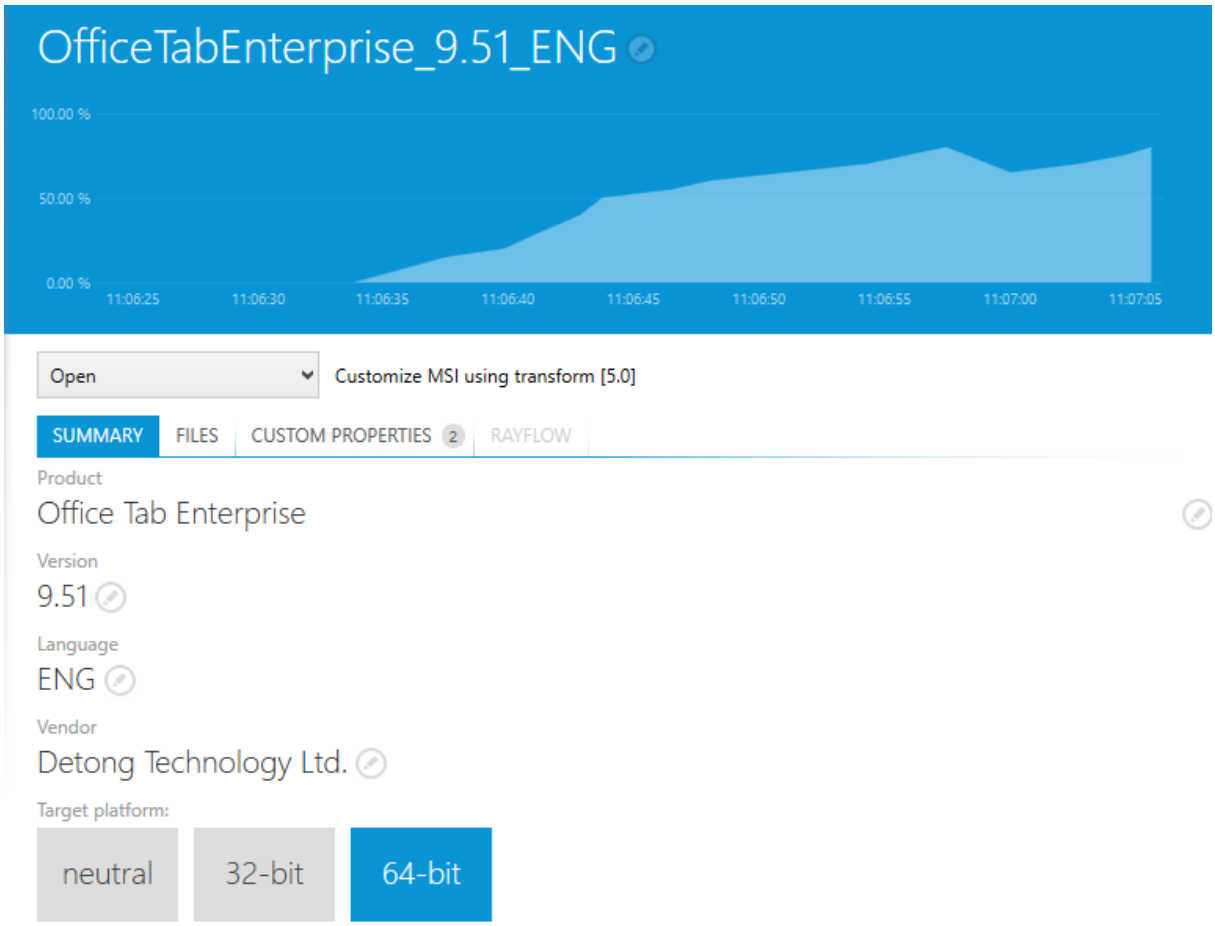
The right panel displays the overview of the run if the element **Overview** is selected on the list. Otherwise, it shows the details of the currently selected task.
Press the **Sidebar** toggle button to expand or collapse the sidebar view.

Top panel contains a chart which represents visually the progress over time. This information cannot be changed, but will be updated automatically by <%PRODUCTNAME% as statuses of tasks are changed. The bottom panel contains a button that - when clicked - generates a report about the current state of the run. More details about reporting is available in the chapters *Generating reports*, *Reports*, and the advanced topic *Customizing reporting templates*

The following topics contain more information about overview and task view:

- Overview
- Task view
- Sidebar

# Overview

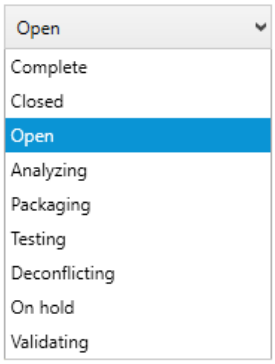



## Header

Above the three tabs the name and the version of the workflow chosen for the run, the name of the run, the progress, and the status of the run are shown.

## Status

An overall status for the run itself can be assigned and changed by using the dropdown menu which can be opened by clicking on the status field.

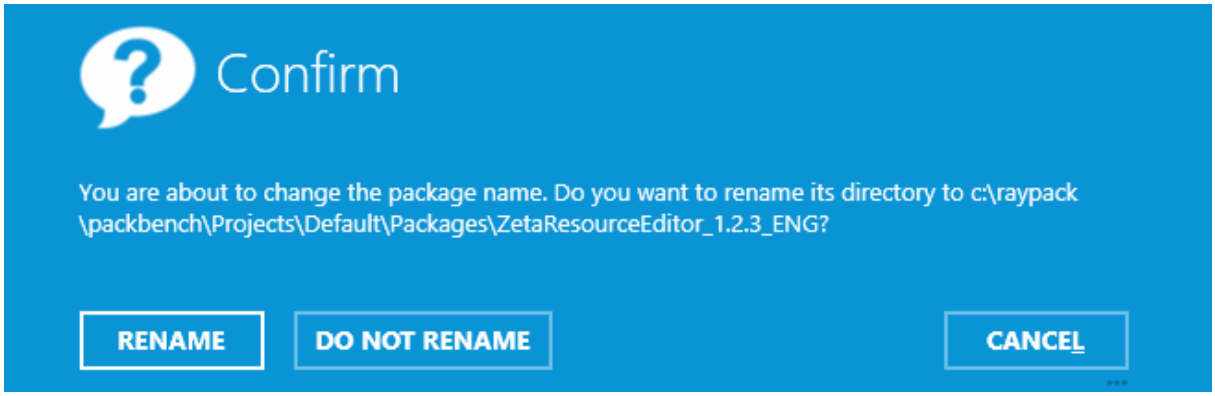



The screenshot above shows the different options which are available in the dropdown menu.

## Changing Run (Package) Name

**Note:**
Changing the name of the package is not recommended and usually should be avoided. Additionally, its availability depends on workflow configuration. Chapters *Strict mode and editing limitations* and *Advanced Tab* contain more information about how the editing may be limited in the run.

In order to change the package name, simply click on it in the header. An editor will be revealed, allowing to enter a new name. Press **OK** to accept it, or **Cancel** to discard changes. If the name of the run's folder is the same as the previous package name, PackBench asks if the folder has to be renamed as well:




For more information about direct value editor, refer to chapter *Direct value editor*

# Summary

The **SUMMARY** tab displays the following information about the package:

- **Standard properties overview**
  Shows the list of standard properties, which depending on the configuration may be editable.

# Changing Standard Properties

**Note:**
Availability of this option depends on workflow configuration. Chapters *Strict mode and editing limitations* and *Advanced Tab* contain more information about how the editing may be limited in the run.

The following standard properties can be changed within the Summary view:

- **Product**
  This is the product which is contained in the package.

- **Version**
  This is the version of the product.

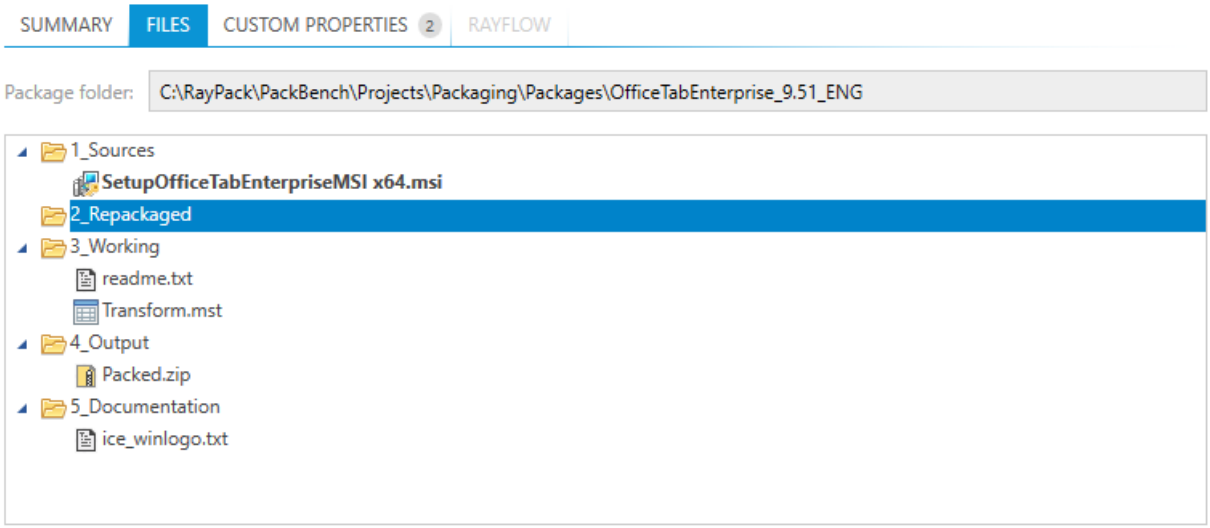
- **Language**
  The language of the product is shown here.

- **Vendor**
  The name of the packages vendor.

- **Target platform**:
  The highlighted field represents the target platform for the package. The available options are neutral, 32-bit, and 64-bit.

For more information about direct value editor, refer to chapter *Direct value editor*

# Files

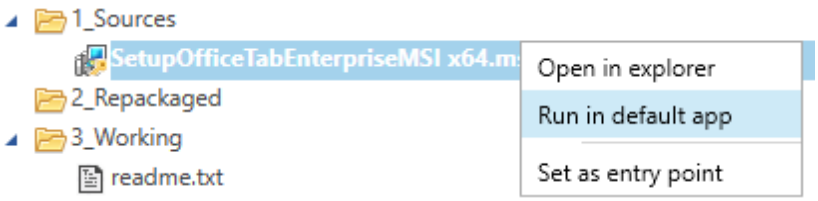


In the **FILES** tab the source folder and the package files for the current run are defined.

At most one file in the list at a time can be marked with a special icon:

meaning that the file is an entry point of the package. For some workflows it may be a source file, source MSI package etc. There may be also no entry point per package, this is possible if the option **SKIP / DEFINE LATER** was used in the *New Run wizard*.

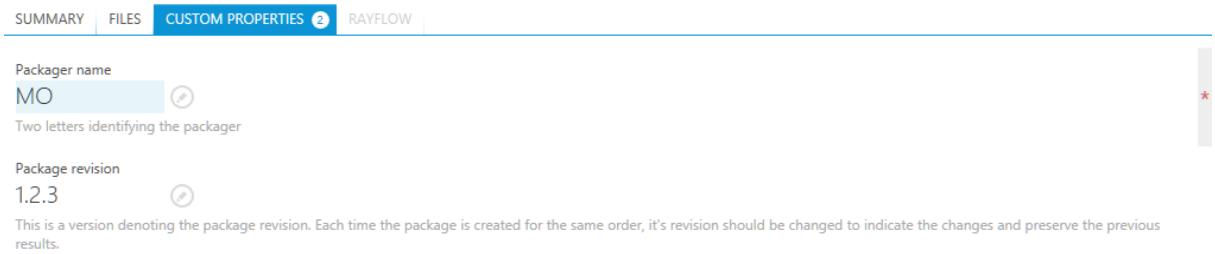
Right-click on an object inside of the structure to open the context menu.




- If the context menu has been opened on a folder the option to **Open in explorer** is available. Choosing this option will open an explorer window which shows the respective folder.
- If the context menu has been opened on a file, the following two additional options are available:
  - **Run in default app** will open the selected file in the default application (for example Microsoft Word for `.docx` files)
  - **Set as entry point** updates the information about the main entry point file of this package. This is the same option as the **Entry point** configured in the *New Run wizard*.
- When working with RayFlow, the additional option **Upload to RayFlow...** is available. For more information about RayFlow, read chapter *Uploading files*.

Once the user goes through the workflow, and more files are created (including but not limited to documentation, package files, supporting files etc.) the list of files may grow.

# Custom Variables




In the **CUSTOM PROPERTIES** tab the custom variables defined for a workflow are shown. The header of the tab shows the total number of custom properties that are part of the run. Mandatory properties are marked by a red star. If mandatory properties are missing or filled with an invalid value they are highlighted in red.

# Changing Custom Properties

**Note:**
Availability of this option depends on workflow configuration. Chapters *Strict mode and editing limitations* and *Advanced Tab* contain more information about how the editing may be limited in the run.

In order to edit a custom variable, simply click on its control element (a textbox for a string variable, a checkbox for a boolean variable etc.). The value is commited immediately.

For more on custom properties refer to the *Custom Variables* chapter and the *Custom Properties page* section.

## RayFlow

This tab is available only when the package has been associated with a RayFlow task during its creation (see *Starting runs* for more information).

For more information about RayFlow, refer to chapter *Working with RayFlow*.

## Generating reports

When the button **Export to PDF Report...** is clicked, a report about the current state of the run will be generated and automatically saved in the documentation subfolder of the current package source folder.

**Note:**
The subfolder name and the exact name of the file are *configurable per project basis*.

After the report file has been generated, it will be automatically opened in the default app (depending on the type of the file).

**Be aware:**
The format of the report (PDF, DOCX, etc.) is determined by the extension of the report file name. See *project configuration for more details*.

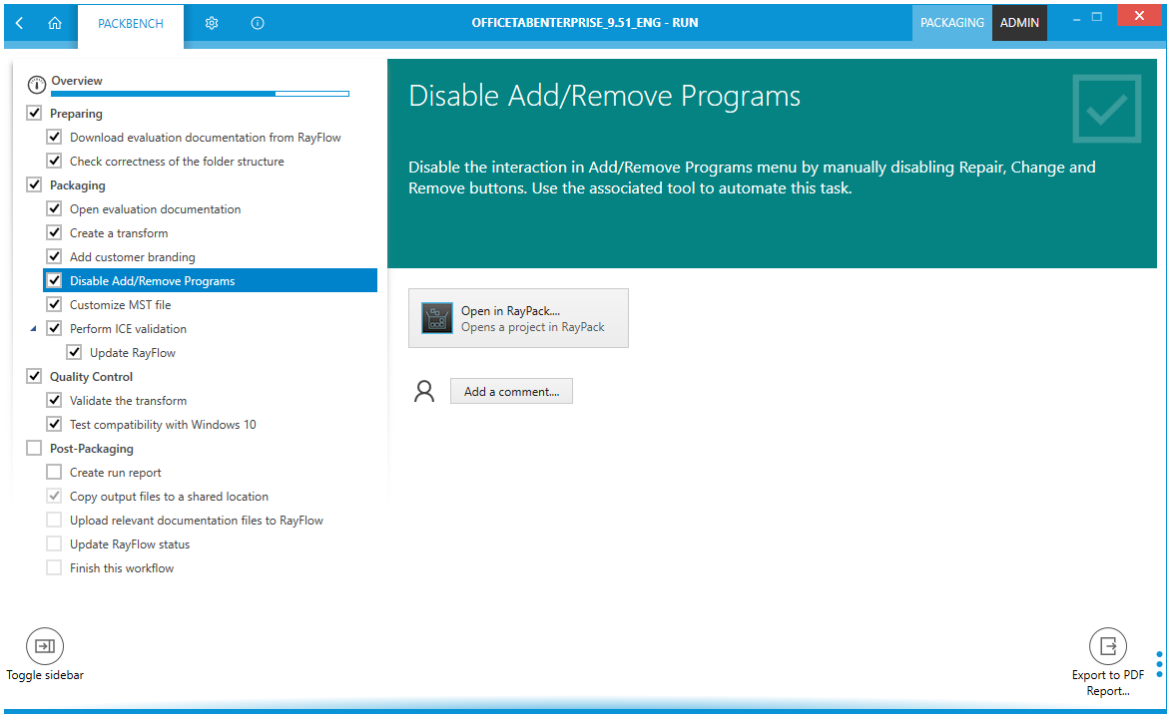
### Customizing Reports

The reports are customizable on a per project-basis. Advanced topic *Customizing reporting templates* contains detailed information on how to prepare a custom template, and the chapter *Reports* explains how to enable the template on a project basis.

# Task View

## The Details View

The details view consists of a swipe-bar, an Overview of the current run and the details for the task currently selected in the run.




Depending on how a workflow is configured a run needs either be completed one step after the other or it can be completed by either completing or skipping all tasks without a specific order.

A run is completed when all tasks have either been completed or skipped. A task can be skipped by right clicking on the task. A context menu will open which offers the option to set the task to either **Open**, **Done**, or **Skipped**. If strict mode is activated, the menu can only be opened if all previous tasks are either in the status **Done** or **Skipped**. It is possible to set a task to **Done** or **Open** by left clicking on the task. The same restrictions apply as if opening the context menu.

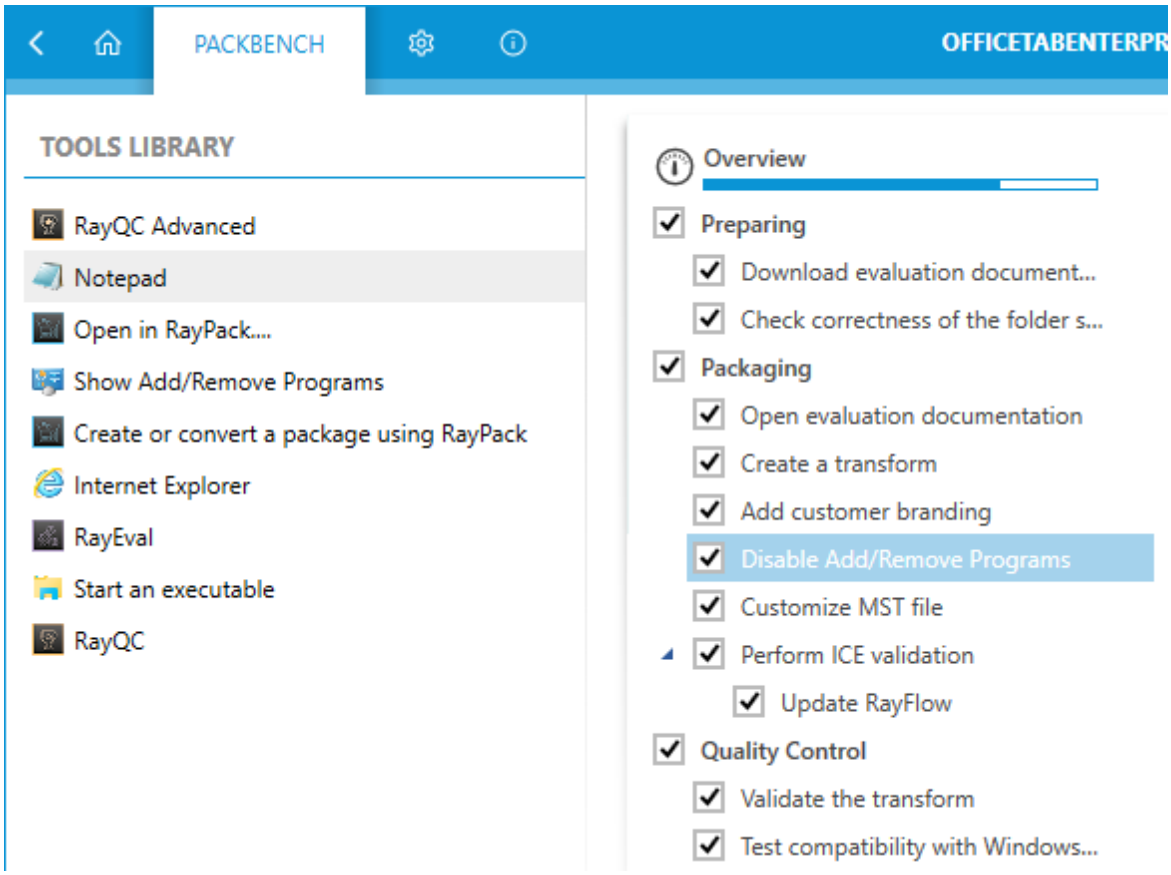
**Be aware:**
In strict mode, if a task is set from **Done** to **Open** which is followed by tasks which are also set to **Done**, the tasks which follow will stay in the state of **Done** and it is not possible to change the state of the tasks until all previous tasks are set to either **Done** or **Skipped** again.

It is possible to add comments to a task during a workflow run by clicking on the **Add a comment...** button. If a comment is already existing, it is not possible to add a second comment to the task. A task can only have one comment at a time, but it is possible to edit the comment by clicking on it.

Tools which have been integrated into a task are started by clicking on them.

# Sidebar

The **Sidebar** contains the library of tools.




- In order to start a tool, click its name.
- In order to hide the sidebar, press **Sidebar** toggle button:

# Strict Mode and Editing Limitations

A run may limit editing capabilities in three different ways:

- Strict mode may be enabled, limiting the order in which the tasks can be processed.
- Editing of standard properties may be disabled.
- Editing of custom variables may be disabled.

## Strict Mode

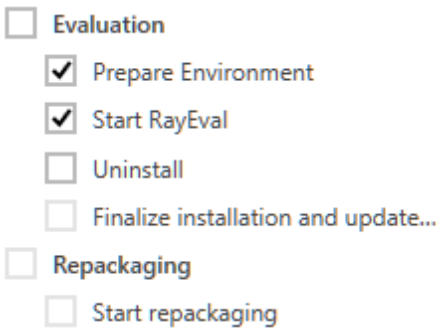
Using strict mode means that in order to proceed to any workflow task and mark it as done, all

previous tasks have to be finished (or skipped) beforehand. Additional rules enforced by using the strict mode are:

- The first workflow task is always enabled.
- If a task is marked as done or skipped, the next task on the same level is enabled
- If a parent task is enabled its first sub-task is enabled as well..
- If a parent task is disabled, all its subtasks are also disabled.
- In order to proceed to the next task, all previous tasks including their sub-tasks have to be done or skipped.

If strict mode is disabled, none of these rules apply and users are generally free to mark the completion as they want.

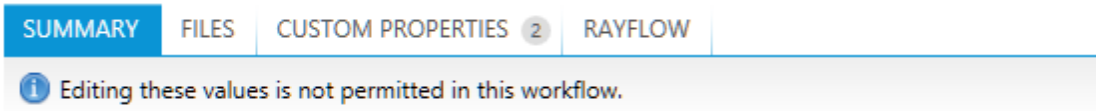
In the following example:




- The **Evaluation** is enabled as it is the first task in the list.
- **Prepare Environment** is active, because it is the first sub-task in an enabled task.
- **Start RayEval** is enabled because the previous task has been completed.
- **Uninstall** task is enabled because the previous tasks have been completed.
- **Finalize installation and update...** is disabled, and will be enabled only when **Uninstall** is finished.
- **Repackaging** is disabled because not all previous tasks have been finished. It will be enabled when **Evaluation**, **Uninstall**, and **Finalize...** are finished.
- **Start repackaging** is disabled because its parent is disabled.

## Disabled Editing of Standard Properties

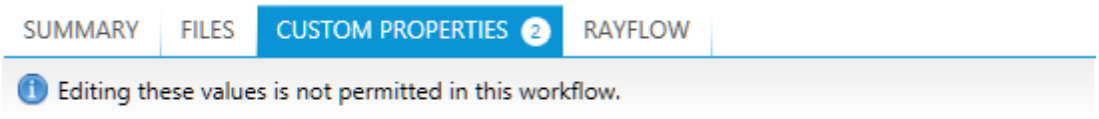
By default, standard properties (like name, version, manufacturer, package name etc.) are read-only once a run is started. This means they can be only edited in the **New Run** wizard and further changes are disabled due to the consistency rules. User will be presented with the following note in the Run view:



This option is defined on workflow level. If the option is disabled, standard properties can be edited even after the run has been started.

## Disabled Editing of Custom Variables

By default, custom variables are editable even once a run is started. Workflow creator can disable this function, which will make all custom variables not editable anymore and the following message will be shown in the Run view:




This option is defined on workflow level. This option is disabled by default, which means that by default all custom variables are editable.

**Note:.**
Variables shown in the task view are still editable, even if the editing has been disabled globally.

# Deleting Runs

Runs can be deleted from the *Runs view*. By removing a run, its metadata and history of changes will be permanently removed. However, any content saved to the package folder (including files created by tools, documentation, folders etc.) will remain.

**Note:**
Deleting runs is generally not recommended. When runs are finished, their status should be changed accordingly, so that the finished runs can be easily filtered out in the *Runs view*.

## In Order to Remove a Run...

1. In the *Runs view*, select the run to be deleted.

2. Right click to open a context menu.

3. Click on **Delete**

4. Confirm the deletion of the run.

# Working with RayFlow

## Introduction

RayFlow is a workflow process management tool with the ability to support diverse workflow processes. The possibility to be customized to fit the user's needs and requirements makes it one of the most efficient and user friendly workflow management tools. This guide shows how to configure and manage RayFlow, so that IT departments can stay ahead, save time, increase productivity, and decrease IT costs.

RayFlow is based on the client-server architecture in which all the information, data, and configuration is stored on the RayFlow server. Users work on this server remotely through the RayFlow web and Windows-based clients.

**Where to find the latest information about RayFlow**

For further information on RayFlow, including its features, functionality and latest updates visit www.raynet.de.

**Enabling RayFlow features in PackBench**

RayFlow connection data is stored in the instance database. The minimal configuration requires entering the URL address containing a valid and running instance of a RayFlow server.
See *RayFlow configuration* for information on how to configure this feature.

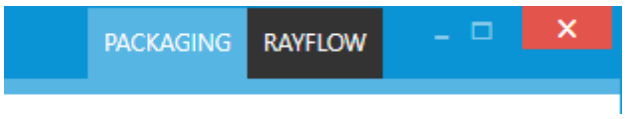
## Signing-in to RayFlow

Saving and opening files from RayFlow requires that the current user is signed-in to the RayFlow instance specified in the *configuration screen*.

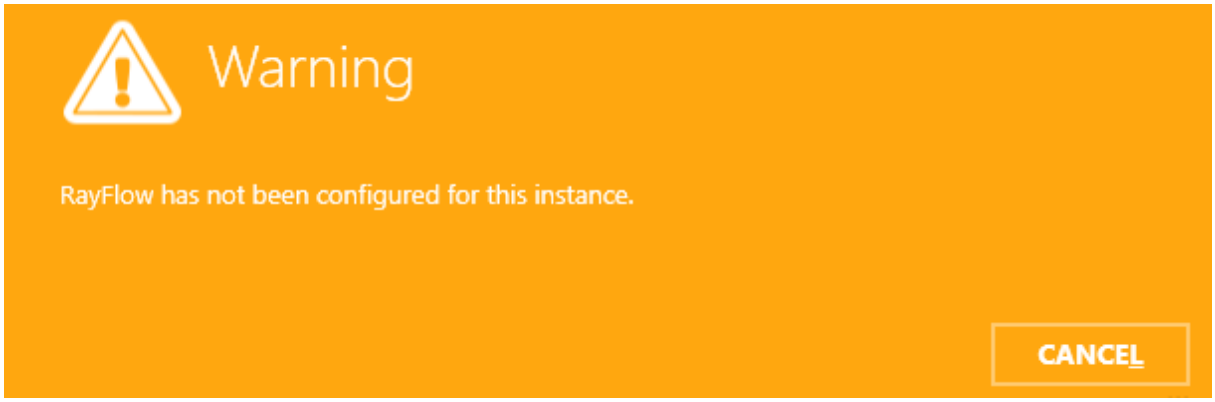
Sign-in procedure makes sure that:

- User has permissions to a specified RayFlow project
- User has permissions to see/edit required RayFlow tasks

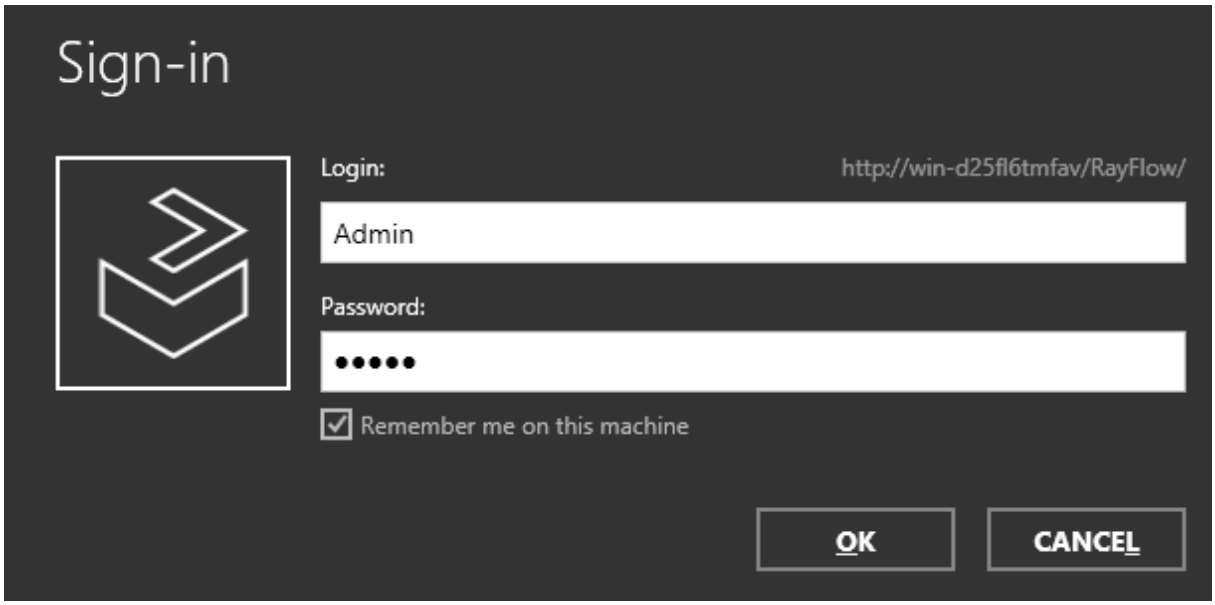
In order to sign-in, press the RayFlow button, located in the top right corner of the screen.




If the server is not valid or not specified, the following warning will be shown:




Otherwise, the sign-in overlay will be displayed over the current window:




The overlay contains the following information:

- The URL address of the current RayFlow instance, as configured in the profile settings
- The text fields for user name and user password
- A checkbox to remember the credentials on the current machine

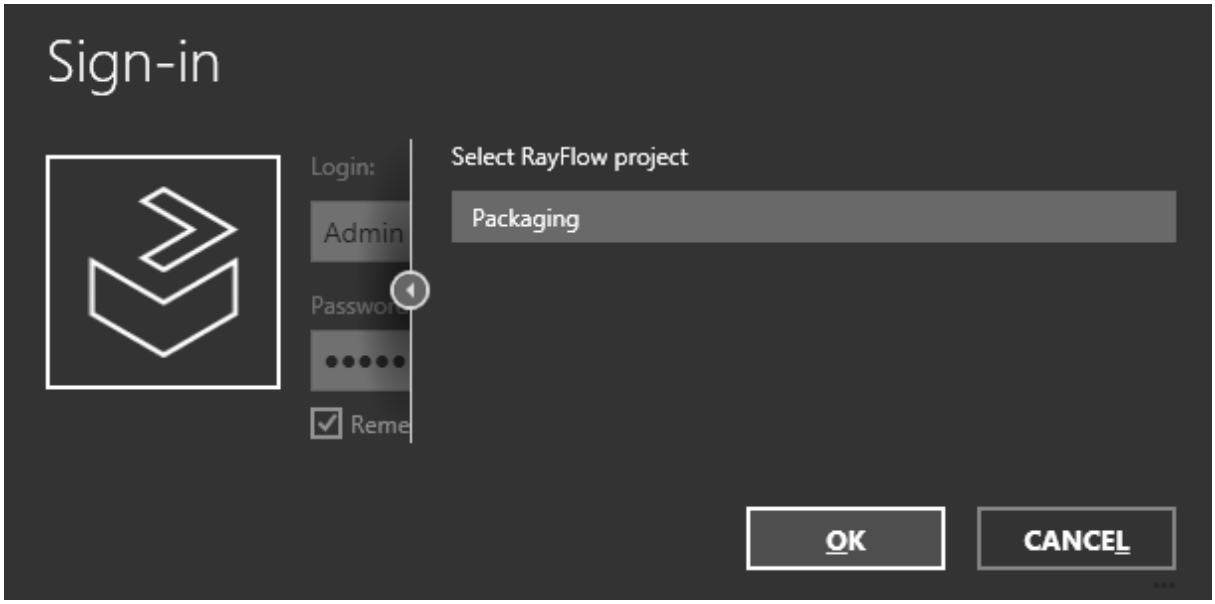
**Note:**
RayFlow credentials have to be delivered by the local RayFlow administrator.

Login and password are required to sign-in to RayFlow. Make sure the checkbox **Remember me on this machine** is marked before pressing OK to save the credentials of a successful login.

Once the credentials are verified, a selection of projects available in the current RayFlow

instance will be shown.

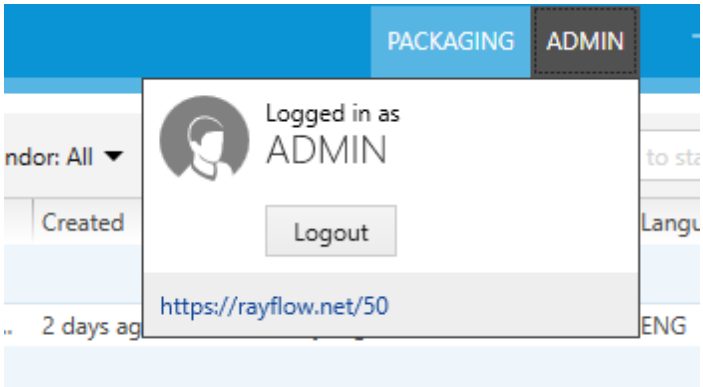
**Note:**
Only projects to which the current user has view permissions are displayed.




Once the project is selected the sign-in procedure is complete, and certain PackBench functions (like opening and saving files to RayFlow are available).

Once authenticated, the RayFlow button in the top right corner of the screen changes its color and displays the user name of currently authenticated user. After clicking on it additional details are shown, including:

- Current project
- An URL address of the current instance
- *A button to logout*



# Privacy Concerns

When the option **Remember me on this machine** is enabled, each time an user signs-in to RayFlow a small cookie is stored on his machine.

The cookie is stored in a predetermined location:

```
C:\Users\<user>\AppData\Local\RayFlow\UserToken.cookie
```

The cookie contains no sensible information (for example it does not contain the user password).

The cookie contains the following encrypted information:

- Security token used to re-authenticate in RayFlow
- ID of the selected project
- The URL server of RayFlow instance
- User name

Once the authentication cookie is deleted, all saved login information is lost.
The information is automatically deleted when the *log out* button is pressed.

# Creating Runs from RayFlow Tasks

Any RayFlow task can be used as a source of a run's files and metadata. RayFlow tasks remain assigned to the PackBench run through its whole lifetime, allowing to upload files, send comments, change data fields etc.

**Note:**
RayFlow task can be only assigned in the *New Run* wizard. After a run is started, it is not possible to assign or reassign RayFlow tasks anymore.

## Prerequisites

There are a few prerequisites that must be met in order for the following steps to work.

1. There is an accessible RayFlow instance running in the network, or available online
2. The project that is going to be used has to be mapped to a RayFlow project. This can be configured in the Project Configuration dialog. See more details in chapter RayFlow.
3. The user needs to be signed-in to RayFlow. Contact the RayFlow administrator for the credentials. In order to sign-in, follow the steps described in chapter Signing-in to RayFlow.

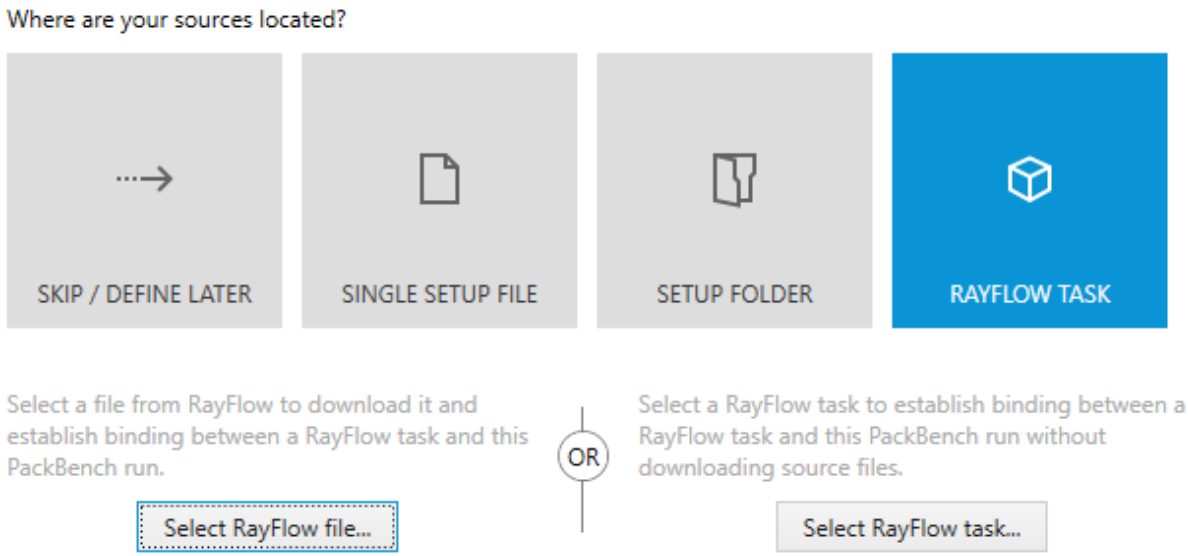
## In Order to Start a New Run from RayFlow...

In order to start a new run and to associate it with RayFlow task, do the following:

1. In the dashboard, press **Start new run** button.

2. Select the PackBench project (if required) and workflow as described in chapter *Starting runs*.

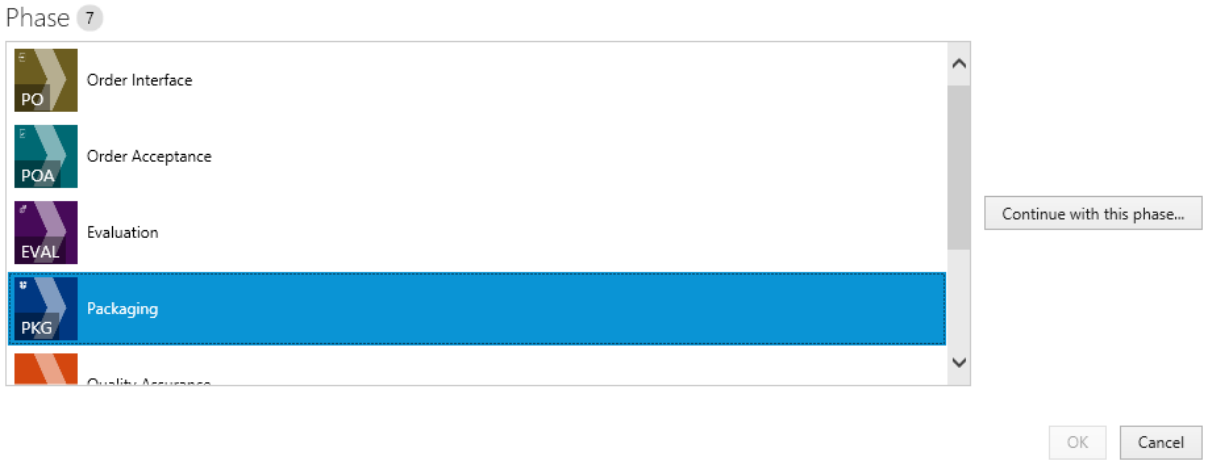
3. Select **RAYFLOW** as a source type.




**Note:**
If the **RAYFLOW TASK** option is not visible, this means that the PackBench project is not configured to use RayFlow. The chapter *RayFlow* describes how to configure it.

4. The following two options are available:
   - **Select RayFlow file...** - opens RayFlow selector prompting for the RayFlow phase, task, and the required file assigned to the task.The selected sources will be downloaded by PackBench and used as package entry point.
   - **Select RayFlow task...** - open RayFlow selector prompting for the RayFlow phase and the task assigned to the run. No sources will be downloaded from RayFlow.

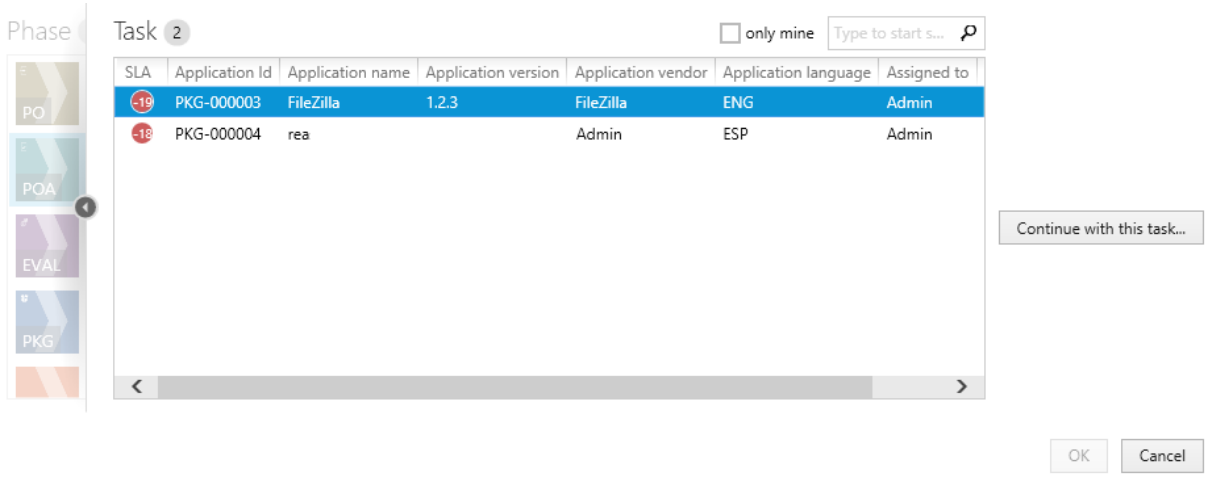
5. A new overlay window will be shown, prompting to select a RayFlow phase, a task, and a file belonging to a task.




6. Select the RayFlow phase which contains the task that is to be opened. The counter above the list shows the total number of phases that are visible to the current RayFlow user.

7. Click **Continue with this phase...**. to proceed to the task selection with the currently selected phase. Press **Cancel** to go back to the backstage menu.

8. Next, select the task that should be opened:



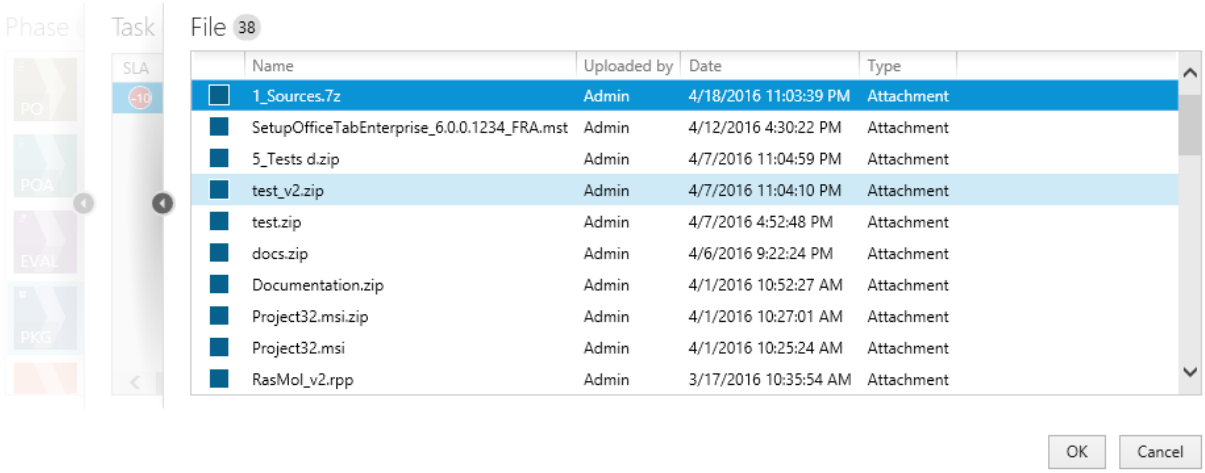

The number displayed above the list shows the total number of tasks that are visible to the current RayFlow user. The results can be narrowed by ticking the **only mine** option, which hides all packages which are not belonging to the current RayFlow user. The task browser displays first 5 data fields from RayFlow to enable easier identification (refer to the RayFlow documentation about setting up the data fields).

9. It is also possible to narrow the results by searching using the search box in the top right corner of the list.

10. (Only if the option **RayFlow task** has been selected)
Once a RayFlow task is selected, press **OK** to accept the selection or **Cancel** to return to the sources selection.

11. (Not available if the option **RayFlow task** has been selected)
Once the required task is selected, press **Continue with this task...** to proceed to the file selection. Press **Cancel** to go to the backstage menu. In order to go back to the phases view, click the arrow on the left side of the list or the partially transparent area containing the tiles representing available phases.

12. (Not available if the option **RayFlow task** has been selected)
On the next screen, select the file to open:




13. (Not available if the option **RayFlow task** has been selected)
The number displayed in the top left corner represents the number of files that are present in the currently selected package.
   - The type column represents the source from which the file can be accessed. Possible values are **Attachment** (file has been physically uploaded to RayFlow) and **Link** (the package is available on a shared location and only the path to the root file is saved in RayFlow).
   - The **Date** and the **Uploaded by** columns denote by who and when the file has been uploaded.
   - The list is sorted by date, the most recent items are displayed on the top of the list.

14. Press **OK** to open the selected file / task. Press **Cancel** to go back to the backstage. In order to change RayFlow task or phase, click the arrow on the left side of the list or the partially transparent area containing the list of tasks.

## Relations Between RayFlow Data Fields and PackBench Properties

Standard properties like the *Application Name*, *Version*, *Vendor*, and *Language* are automatically mapped to RayFlow datafields using standard out-of-the-box configuration. If the default names of the instance are changed, it is important to configure RayFlow data mappings to indicate, which of the new fields are to be reported as name, version, vendor, and language respectively.

Custom variables are mapped to RayFlow datafields by using their names. A matching between a variable and a datafield is automatically recognized if any of the following is fulfilled:

- The name of the custom variable and a datafield in RayFlow are the same (case insensitive).
- The name of the custom variable is equal to the name of datafield in RayFlow, excluding

spaces. For example, `MyVariable1` will be mapped to the `My Variable 1` data field.

# Managing RayFlow Task

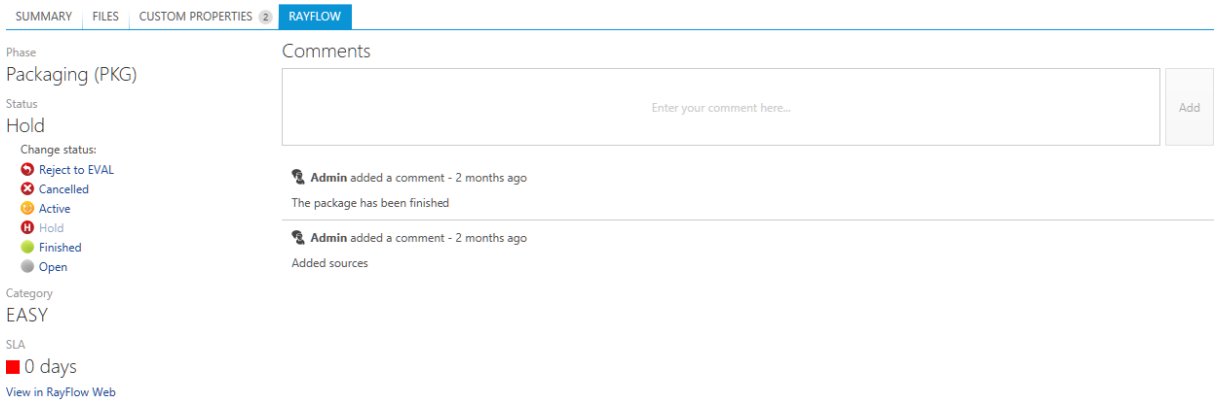
For any run that has been associated with RayFlow, the following actions are available from the *Run view*:

- *Ability to view RayFlow task status, metadata and comments*
- *Ability to upload file(s) to RayFlow*
- *Ability to update RayFlow datafields values based on custom variables*

**Note:**
These functions require that a run has been configured to use RayFlow as its source. Read the chapter *Creating runs from RayFlow tasks* to find out how to do it. Existing runs which haven't been configured to use RayFlow cannot use these functions, and respective buttons and controls will be hidden from the UI.

# Managing State and Comments

Existing runs that have been configured to use a RayFlow task as a source have additional tab **RAYFLOW** available in the *Overview* section.




## Phase

This is the name of the current phase in which the RayFlow task is in. This is a read-only information which can be changed only by using RayFlow triggers (see *Change status paragraph*).

## Status

This is the current status of the RayFlow task. This is a read-only information which can be changed only by using RayFlow triggers (see *Change status paragraph*).

# Change Status

This is the list of triggers available for the current package. The triggers can change the package state and move the package to other phases, depending on RayFlow configuration. If a trigger is disabled it means that it represents the current status, for example in the above picture, as the RayFlow task is already in **Hold** state, the trigger **Hold** is unavailable.

**Note:**
Depending on RayFlow configuration, selected project and phase, the list of available triggers, their names and status may be different. Contact the RayFlow administrator for more details..

# Category

Shows the category of RayFlow task. This is a read-only information.

# SLA

Shows the SLA (*Service Level Agreement*) expressed in days. This is a read-only information.

# View in RayFlow Web Button

Opens a link to the RayFlow task using default system browser.

# Comments

Shows a list of comments and a textbox to add new comments. Already added comments are using human-friendly data representation. Hover the mouse over the human friendly string (for example *2 months ago*) to reveal a full date string.

In order to add a new comment, enter its content to the textbox and press button **Add**.

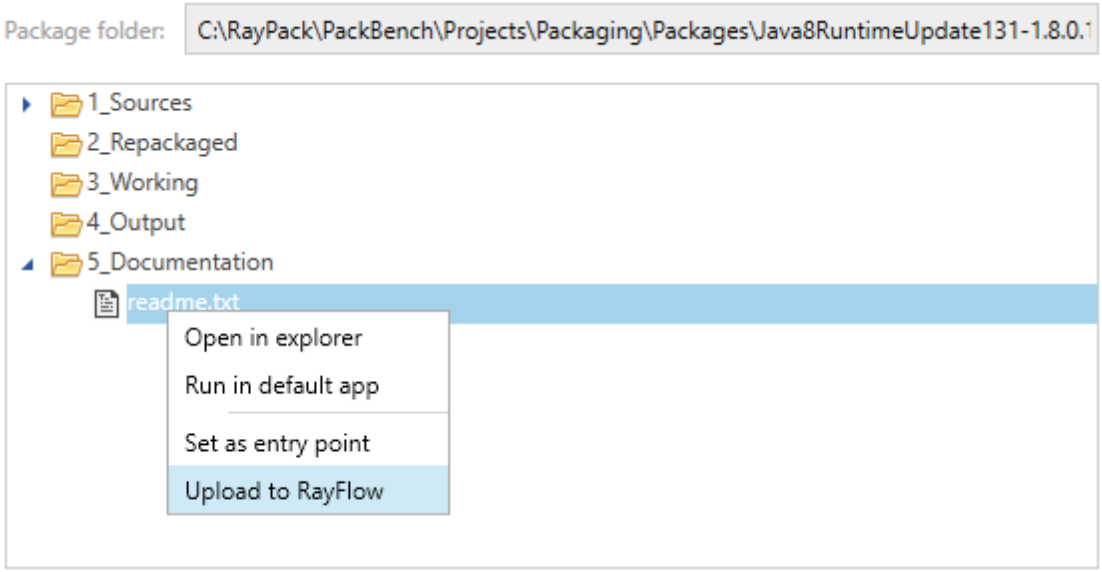
**Note:**
RayFlow Web allows adding rich text comments using HTML. This option is disabled in PackBench, and all comments containing HTML tags are stripped down to a text version. In order to see the comments using HTML formatting, go to the Web version by using **View in RayFlow Web** button.

# Uploading Files and Folders

PackBench provides a simple interface for files upload to the associated RayFlow task.

## Uploading Single Files

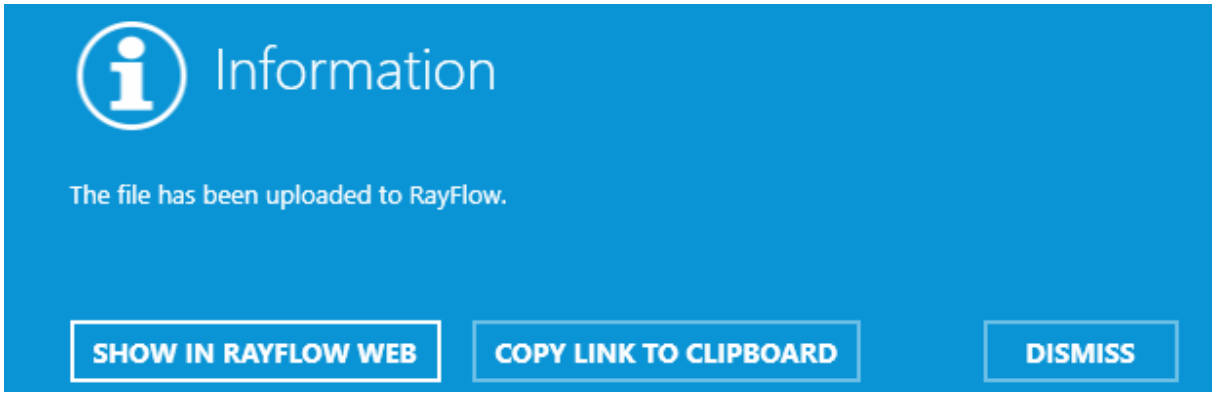
A single file can be uploaded to the associated RayFlow task by selecting menu item **Upload to RayFlow** available on any single file entry within the **FILES** tab:




The upload will start automatically when you press the menu item.

**Note:**
If not yet logged in to RayFlow, PackBench will ask for user credentials..

Depending on the size of the folder, the process may take a while. Once the upload is done, the file will be available in RayFlow.

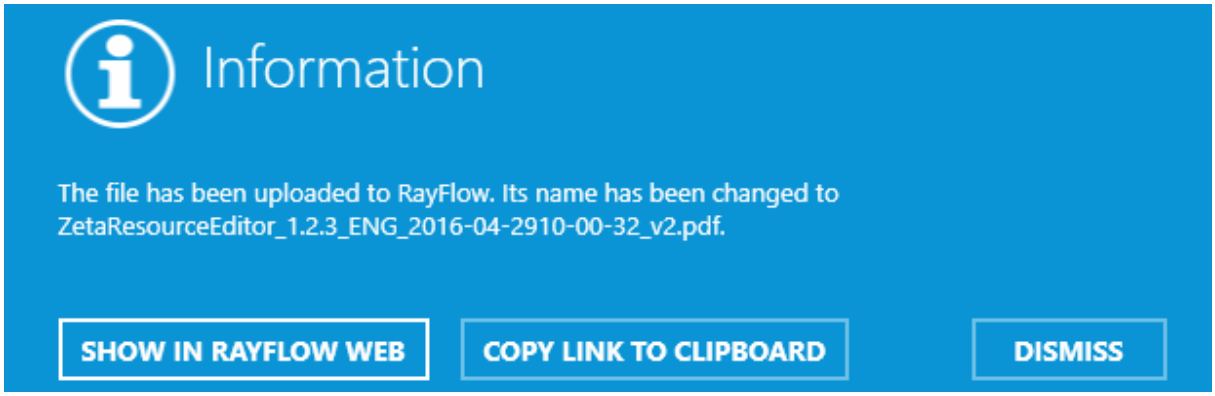


Press **SHOW IN RAYFLOW WEB** to open the task link in the default system browser. In order to copy the download link to clipboard, press **COPY LINK TO CLIPBOARD**. Note that the link may only be accessed by users already signed-in to RayFlow. Press **DISMISS** to close the dialog.

In some cases, RayFlow may additionally adjust the file name. This may happen if one of the following is true:

- File with the same name already exists (RayFlow does not allowed duplicated files within the same task) then suffix _v2 will be added automatically.
- File name contains characters that are invalid for Windows operating system or the name is not valid file name. In that case, the file name is adjusted to be valid and unique.

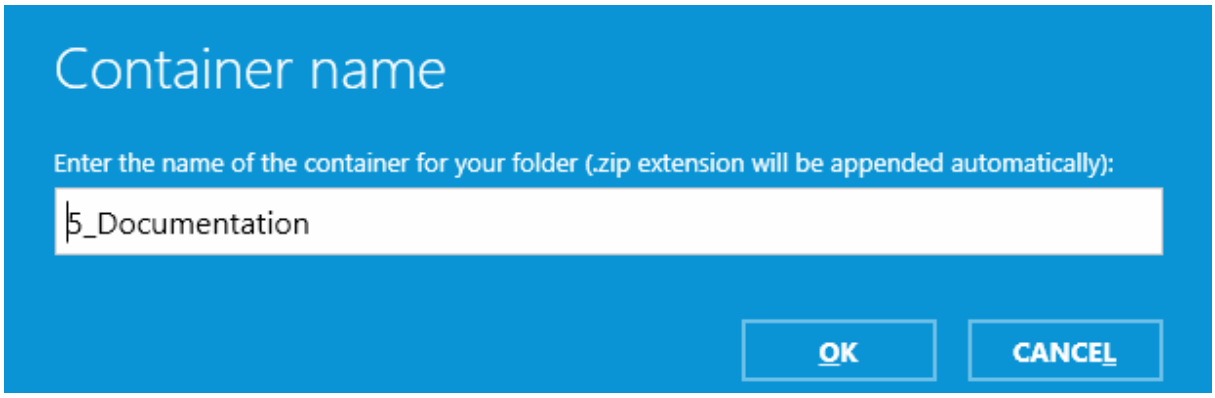
Should any change be done automatically, a message similar to the following may be shown:




# Uploading Single Folders

To upload a whole folder (including its all files and subfolders) select menu item **Upload to RayFlow** available on any folder entry within the **FILES** tab.

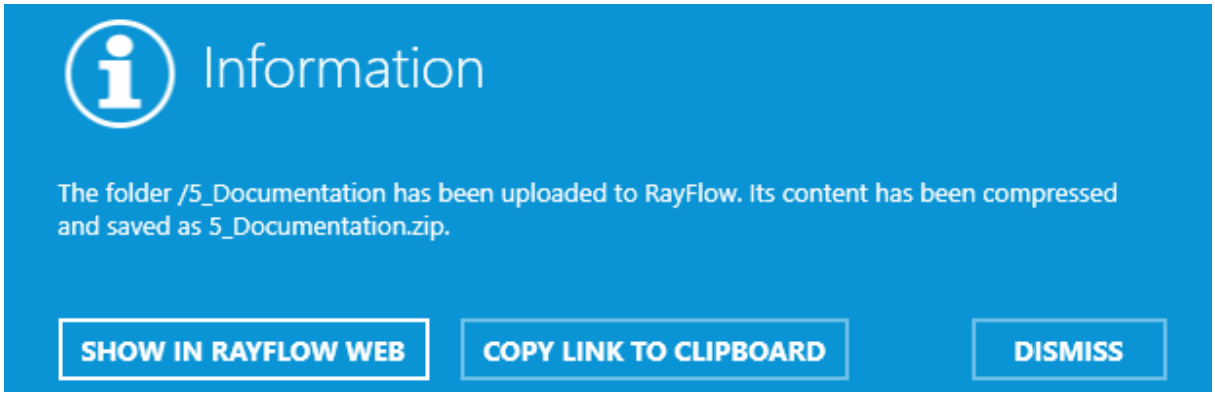
PackBench will ask for a name for the container, which by default is equal to the name of the folder:



When the **OK** button is pressed, the upload will start automatically. Depending on the size of the folder, the process may take a while.

In some cases, RayFlow may additionally adjust the folder name. This may happen if one of the following is true:

- A folder with the same name already exists (RayFlow does not allowed duplicated files within the same task) then suffix _V2 will be added automatically.
- A folder name contains characters that are invalid for Windows operating system or the name is not valid file name. In that case, the folder name is adjusted to be valid and unique.




Press **SHOW IN RAYFLOW WEB** to open the task link in the default system browser. In order to copy the download link to clipboard, press **COPY LINK TO CLIPBOARD**. Note that the link may only be accessed by users already signed-in to RayFlow. Press **DISMISS** to close the dialog.

# Downloading Files

The *New Run wizard* allows the linking a run package with a RayFlow task or a file. Once the run has been started, any additional files can be downloaded from RayFlow and put in an arbitrary folder.

Examples:

- Downloading evaluation documentation (in PDF format) to a package whose sources have been already downloaded from RayFlow
- Downloading additional documents, transforms, supporting files, reports etc.
- Downloading updated versions of sources

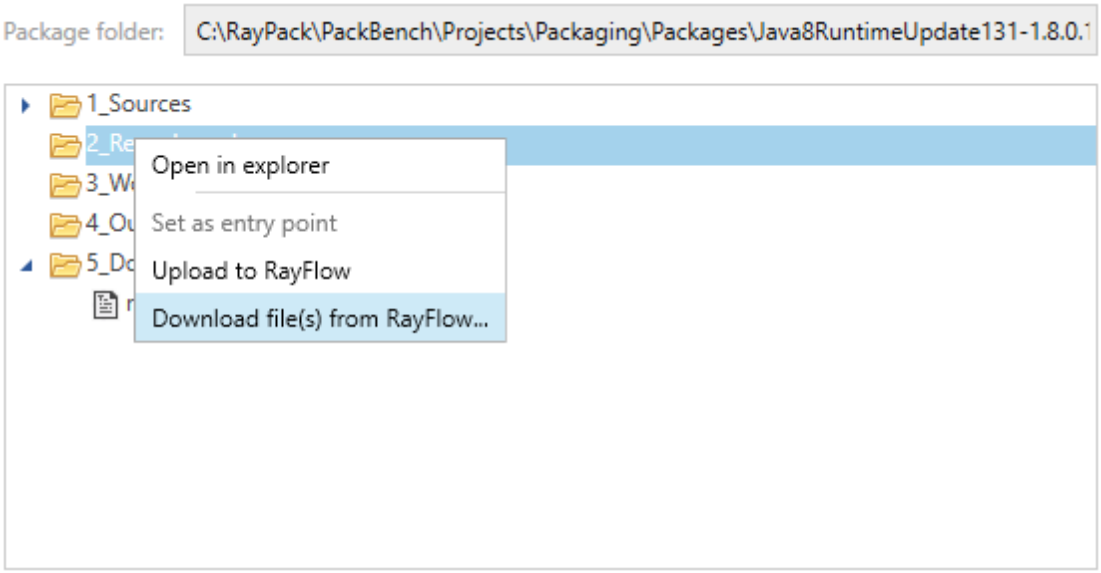
## In Order to Download Additional Files from RayFlow...

1. Go to the *FILES* tab within the *OVERVIEW* section.

2. Select the folder to which the files have to be downloaded. If the folder does not exist yet, use Windows Explorer to create it first and then press **F5** in the file browser to refresh its content.

3. Right-click to open a context menu.

4. Select **Download file(s) from RayFlow...**




5. PackBench will ask to select a file for download. Press **OK** to accept the selection or **Cancel** to return to the file browser.

6. The file will be downloaded and extracted (if its extension is `.zip` or `.7z`).

**Note:**
Any existing files will be overwritten without prompt.

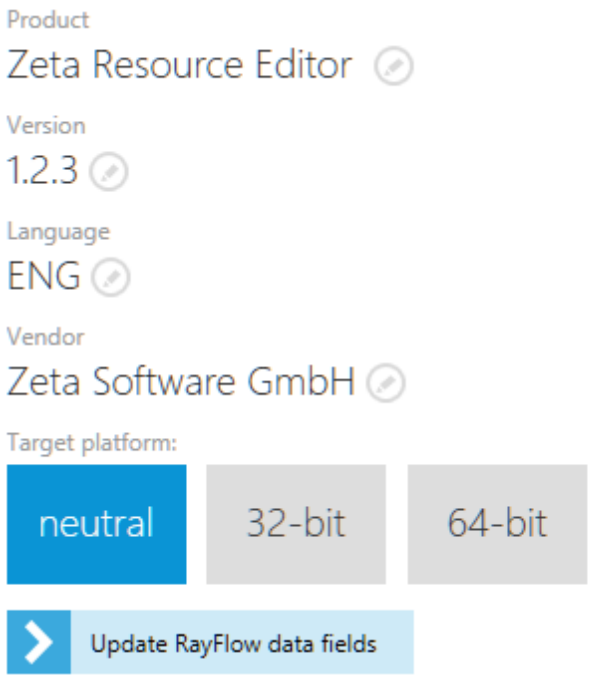
# Updating Data Field Values

If there is a mapping between PackBench custom variables and RayFlow datafields, RayFlow values can be updated on demand.

## In Order to Update RayFlow Values Based on the Current Run...

1. Go to **SUMMARY** tab to synchronize standard properties.

2. Press the **Update RayFlow data fields** button to perform a synchronization back to RayFlow




3. If the button is not visible, this means that the workflow administrator disabled editing standard properties and the update is not feasible. Custom variables can still be synchronized by going to the **CUSTOM PROPERTIES** tab and pressing **Update RayFlow** button.

## Mappings Between Variables and Data Fields

PackBench uses a zero-touch mapping between its variables and RayFlow:

- Standard properties: Application name, version, vendor, and language are automatically mapped to RayFlow datafields: `Application Name`, `Application Version`, `Application Vendor`, and `Application Language`.
- Custom variables are mapped to RayFlow datafields by using their names. A matching between a variable and a datafield is automatically recognized if any of the following is fulfilled:
  - Name of the custom variable and a datafield in RayFlow are the same (case insensitive)
  - Name of the custom variable is equal to the name of datafield in RayFlow, excluding spaces. For example, `MyVariable1` will be mapped to the `My Variable 1` data field.

# Importing Data Fields from RayFlow

Custom variables are automatically mapped to RayFlow datafields using the rules described in the chapter *Updating data field values*. The names must be defined at workflow-creation time. In order to avoid mistakes, PackBench contains a simple function to automatically import RayFlow variables, which ensures that:

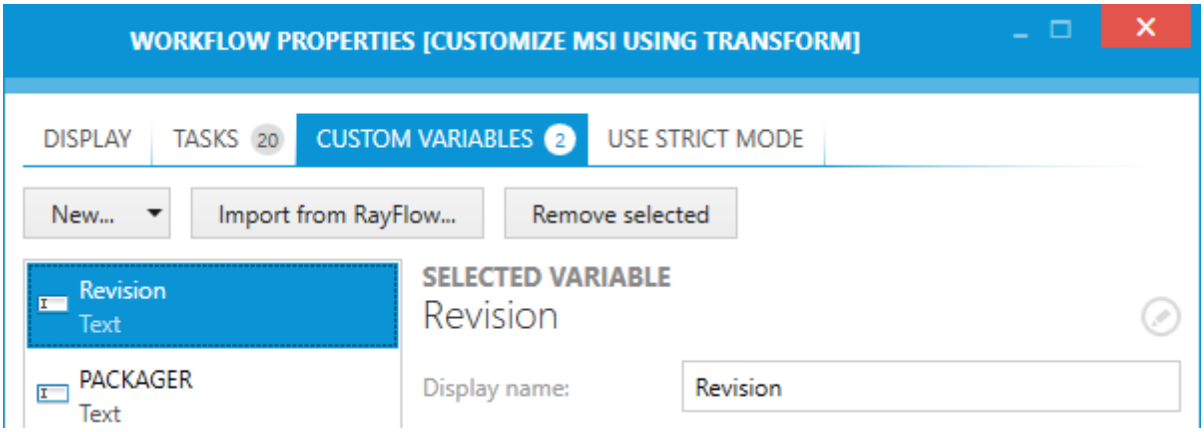
- Variables are of correct types
- Their default values are the same
- Their validation rules are the same
- Their names ensure that the mapping *will be automatically recognized when updating the values back*.

## In Order to Import Data Fields from RayFlow...

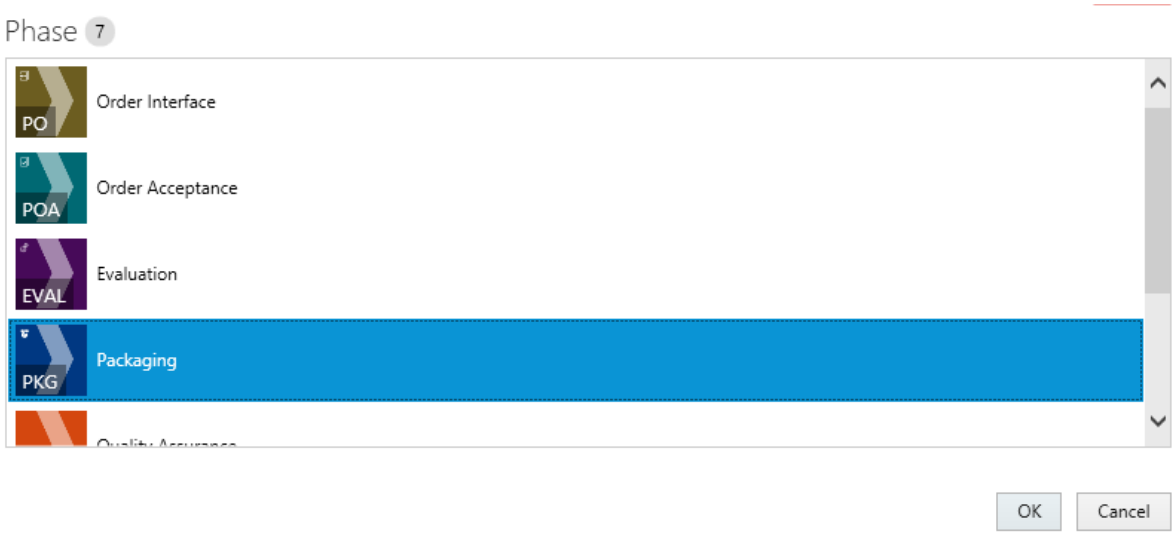
1. Edit any workflow using the techniques described in chapter *Configuring Workflows*.

2. Go to **CUSTOM VARIABLES** tab.

3. Press **Import from RayFlow...** button.

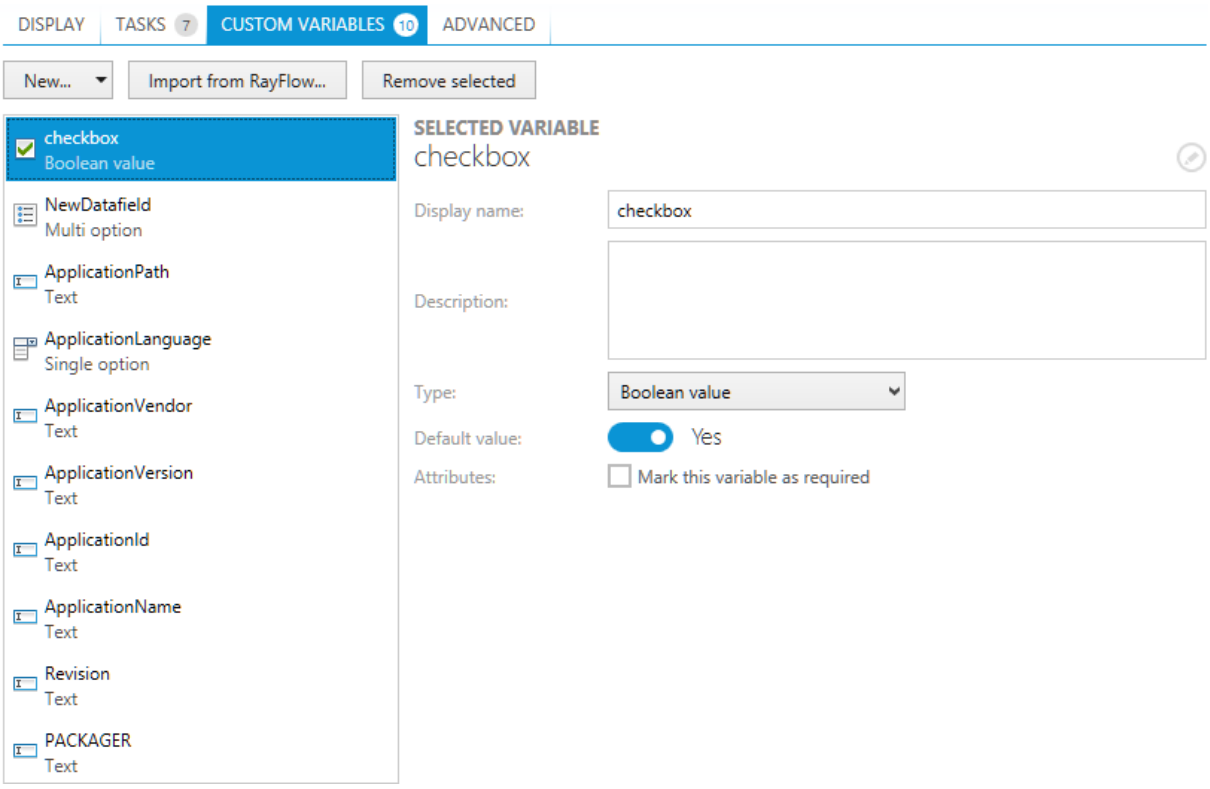



4. You will be asked to select a RayFlow phase which contains required data fields.




5. Press **OK** to import datafields. Press **Cancel** to return back to the editor.

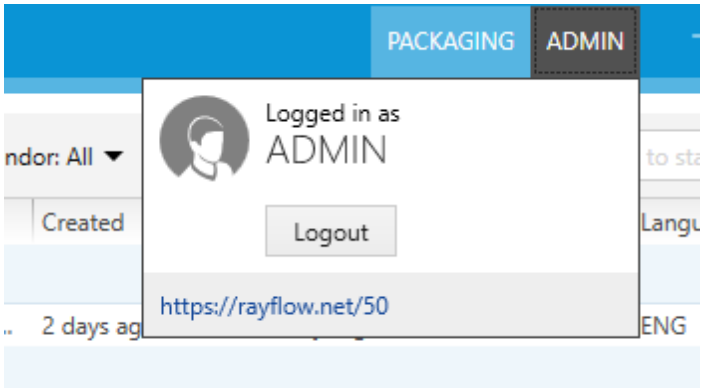
6. All global variables will be imported automatically. Unnecessary ones can be removed by highlighting them and pressing the **Remove selected** button. The variables can be reordered and edited using standard techniques described in chapter *Custom Variables*.




7. Press **OK** or **Apply** to save changed, or **Cancel** to return back to the Workflows view.

# Signing Out from RayFlow

In order to sign-out from RayFlow, or sign-in as another user, press the RayFlow button in the right top corner:




And then click the **Logout** button to sign-out from RayFlow.

# Configuration

This chapter shows how to configure projects, workflows, and tools inside of PackBench.

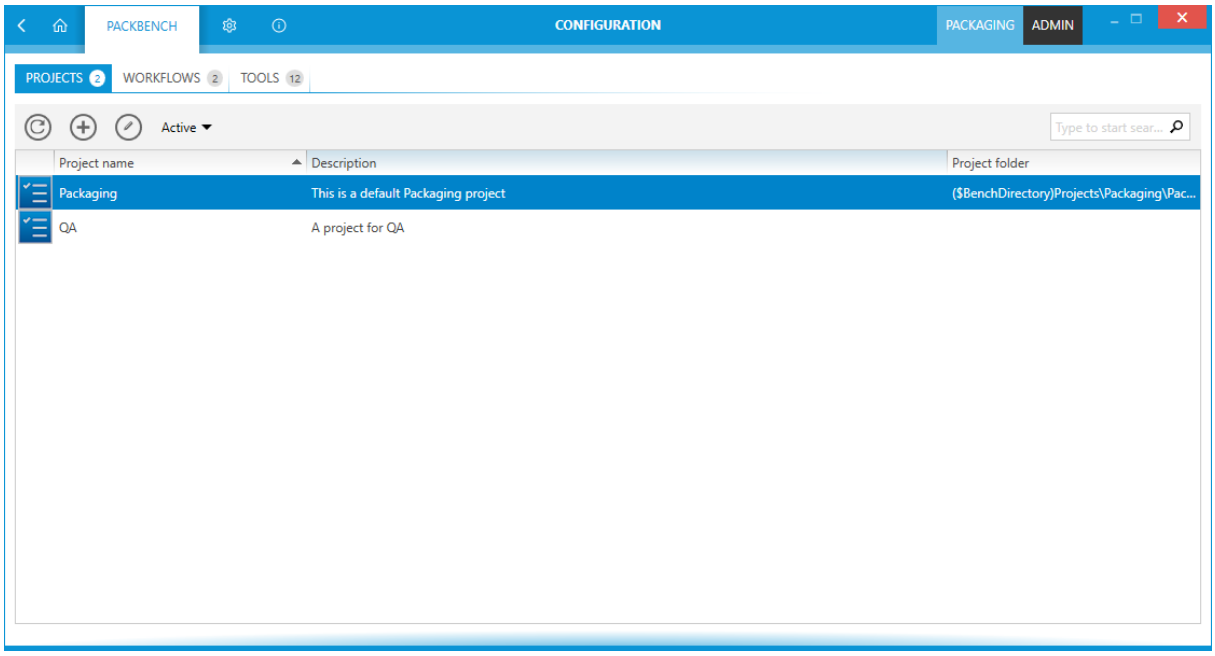
## The CONFIGURATION View

The **CONFIGURATION** view can be opened by using the **configuration** tile in the **configuration** part of the Home Screen. It consists of three tabs, the **PROJECTS** tab, the **WORKFLOWS** tab, and the **TOOLS** tab. At the right-hand side of the name of each tab a number is shown. This number represents the number of projects, workflows and tools that exist in the current PackBench instance.

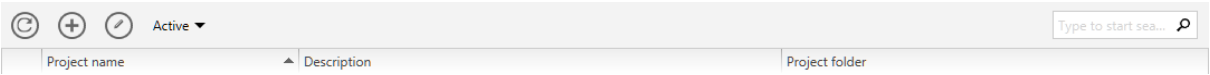
**Note:**
It is not possible to delete projects, workflows, or tools from PackBench. Projects and workflows can be archived and tools can be hidden.

# Configuring Projects

This chapter explains how to create a project with PackBench and how to edit an already existing project.




To configure a project left-click on the **Configuration** tile in the **config** section of the Home Screen. The **CONFIGURATION** view is opened. Select the **PROJECTS** tab inside of the **CONFIGURATION** screen.




The **PROJECTS** tab shows an action bar, the header of the content area, and the content area where the projects are shown.

## Projects Tab Header

The header of the tab is highlighted when the tab is selected. Furthermore, it shows a count of the projects which are currently existing in PackBench. A project cannot be deleted, but it can be archived. An archived project will still be counted.

### Action Bar

There are four different abilities available in the action bar.

## Refresh this View Button

This button can be used to refresh the content area of the **PROJECTS** tab. If any changes have been made to any of the projects, this button should be used to ensure that the changes are shown in the tab.

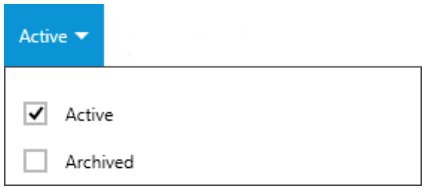
## Create New Project Button

Click this button to create a new project. For more information on this, refer to the *Creating Projects* section of this chapter.

## Edit Tool Button

Click this button to edit currently selected project. For more information on this, refer to the *Editing Projects* section of this chapter.

## Status



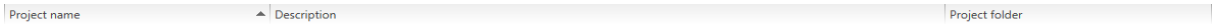

The **Status** button contains a dropdown menu which can be used to define which projects will be shown in the **PROJECTS** tab. Available options in this menu are **Active** and **Archived**. Enable or disable an option by using the checkbox related to the option. If archived projects are shown, they will be grayed out. The title of the option will also change to **Active, Archived** to clarify that all projects are shown. By default, only active projects are shown.

## Keyword Search Field

The keyword search field can be used to filter the projects. Type the keyword into the search field. The projects will be filtered for the keyword while typing.

For example, type "sample" into the search field and only projects which contain the word "sample" in either the **Project name**, **Description**, or **Project folder** column will be shown.

### Sorting



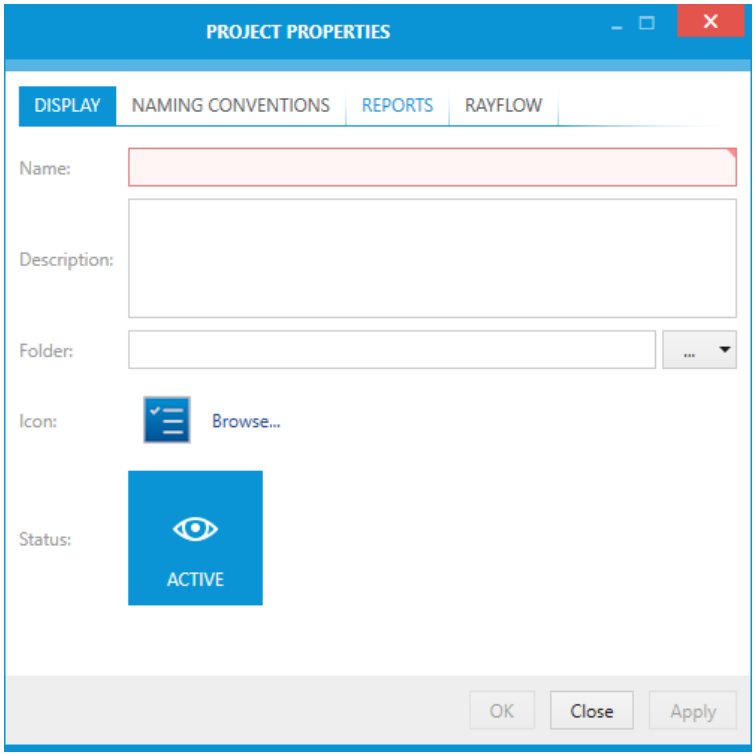

It is possible to sort the content by column. Select the column and click on it to sort the projects alphabetically. An arrow will appear next to the selected column header. To change the sort order simply click on it again. The arrow will indicate the order by either pointing upwards or downwards.

The columns themselves can be interchanged by drag and drop. Simply drag the column header and drop it at the target position.

# Creating Projects

To create a new project open the **PROJECTS** tab in the **CONFIGURATION** view and then either open the context menu and select **New...** or click on the **Create new project** button (⊕).

The **PROJECT PROPERTIES** window will open.


PROJECT PROPERTIES
DISPLAY
NAMING CONVENTIONS
REPORTS
RAYFLOW
Name:
Description:
Folder:
...
Icon:
Browse...
Status:
ACTIVE
OK
Close
Apply

**Note:**
**Name** is mandatory.

Enter the name of the new project into the Name field and click on the **OK** button to save the project and close the window or on the **Apply** button to save the project and configure it further. To learn how to further configure the project refer to the *next chapter.*

# Editing Projects

Projects can be edited by opening the **PROJECT PROPERTIES** of the target project. They can be opened by clicking on the **Properties...** option in the context menu.

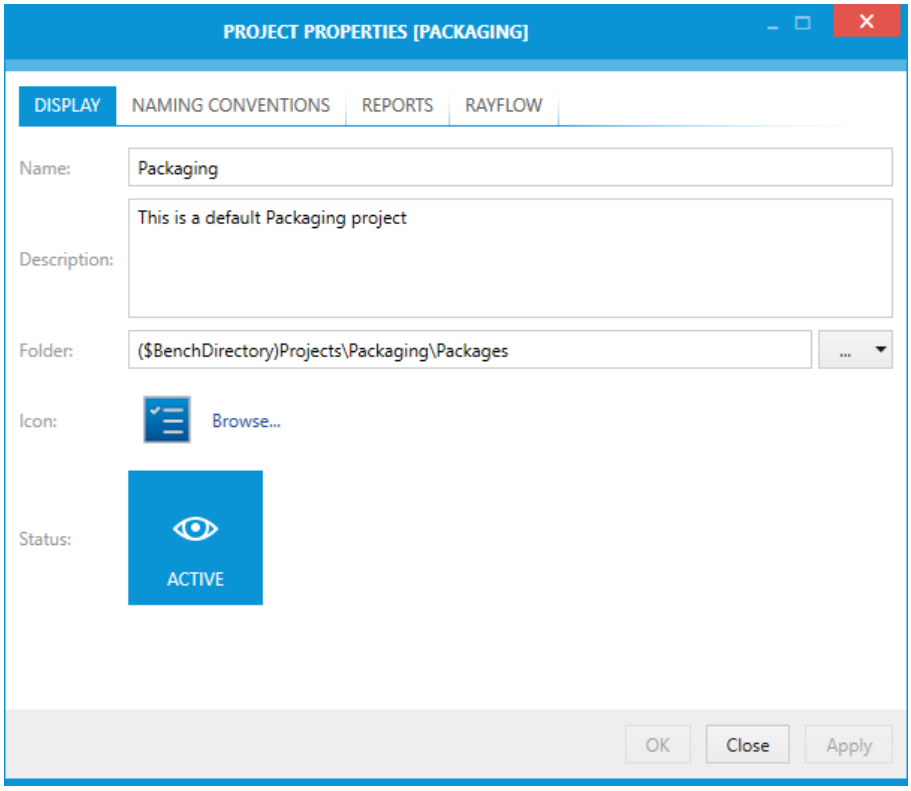
The dialog consist of four functional tab groups:

- *DISPLAY*
- *NAMING CONVENTIONS*
- *REPORTS*
- *RAYFLOW*

Click on the **OK** button to save the changes and close the window, click on the **Apply** button to save the changes without closing the window, or click on the **Cancel** button to dismiss the changes.

## Display

The **DISPLAY** tab contains various settings which define how the project is presented. Additionally, the status and default project path can be defined here.




## Name

Shows the name of the project and can be used to change the name. This field is mandatory for all projects.

## Description

The description of the project can be entered or altered here.

## Folder

The path to the folder which is related to the project is shown here. It can be changed manually or by using the dropdown menu next to the field. The dropdown menu offers to options:

- **From local folder...**: When this option is selected, the **Select Folder** window will open. Browse to the target directory and press the **Select Folder** button.
- **Using variables...**: When this option is selected, the **VALUE EDITOR** will open. For more on the **VALUE EDITOR** refer to the *Value Editor* chapter.

## Icon

This is the icon associated with the project. The icon is shown primarily in the project browser, shown on application startup (if there is more than one project in current product instance).

In order to define a custom icon, press **Browse..** to browse for supported icon files.

**Tip:**
It is recommended to use `.ico` files. PackBench can also extract the icons from executables and use other graphic formats.

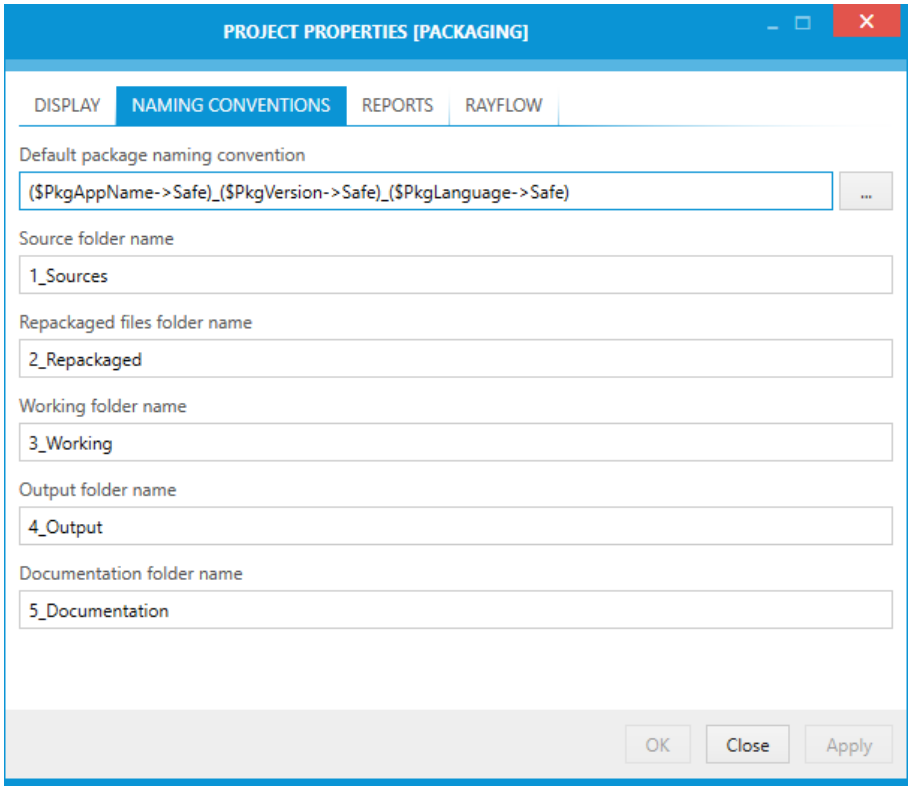
## Status

The status of the project can be defined by clicking on the button. It can be set to either **ACTIVE** or **ARCHIVED**. The current status will be shown on the button and the button will be highlighted if the current status is **ACTIVE**.

Click on the **OK** button to save the changes and close the window, click on the **Apply** button to save the changes without closing the window, or click on the **Cancel** button to dismiss the changes.

# Naming Conventions

The **NAMING CONVENTIONS** tab contains various settings that define how folders are named. The settings here affect primarily the package name and the values of several variables available to tool creators.




## Default Package Naming Convention

This shows how packages for this projects will be named by default. Custom variables can be used to ensure the value is created dynamically as the run's data is collected. Press the browse button [...] to open the variable editor.
For more information about formatted strings, refer to this section: *Variable Editor*

## Source Folder Name

This is the name of the folder where the sources for this project are stored by default.

## Repackaged Files Folder Name

This is the name of the folder where the repackaged files are stored.

## Working Folder Name

This is the name of the working folder.

### Output Folder Name

This is the name of the output folder. All output files will be stored here.

### Documentation Folder Name

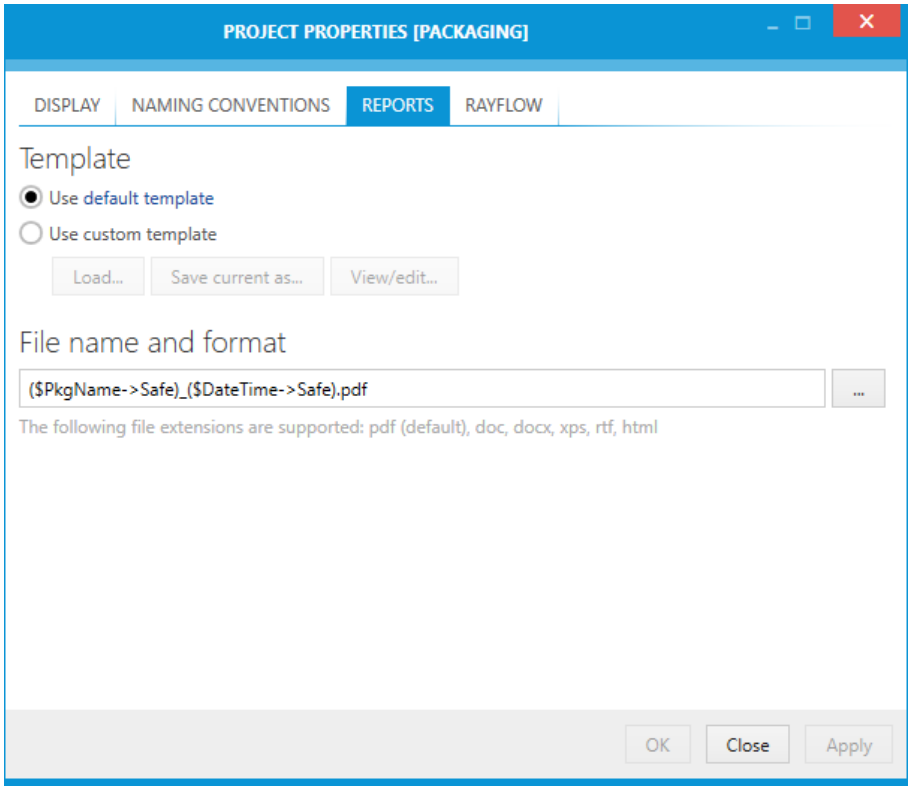
This is the name of the documentation folder. By default, documentation files will be stored here.

**Note:**
The folder names specified in this dialogs are always relative to the package folder. These names are accessible using variables, so it is up to the tool / workflow creator to make sure they are used instead of hardcoded values. Standard tool configuration available out-of-the-box are already aware of them. These values are not resolvable, meaning they must not contain variables.

**Tip:**
By default, these 5 subfolders (**Source folder**, **Repackaged Files folder**, **Working folder**, **Output folder** and **Documentation folder**) are created automatically when a run is started. If this functionality should be disabled and reverted back to the original PackBench 2.1 behavior, then the appropriate option has to be disabled on a workflow basis, as described in chapter *Advanced Tab*.

## Reports

The **REPORTS** tab contains settings that control how reports are generated, including default file name, format and template.




## Template

The radio button can be used to choose one of the following options:

- **Use default template**
  When this option is used, a default template is applied to generated documentation. By default, the templa

> **Tip:**
> When user clicks the link **Default template**, PackBench opens the folder where the report template is and highlights its name in Windows Explorer.

- **Use custom template**
  An existing `.docx` file can be used as a template, overriding the default on project basis.The file must follow certain rules, described in section *Customizing reporting templates*.

When a custom template is selected, additional three buttons are available:

- **Load...**
  Opens a file browser allowing to import custom `.docx` file as a default report template.

> **Tip:**
> It is not recommended to start from scratch, because PackBench expects certain placeholders to be found in the template. Instead, it is much easier to adjust the default

template that comes out-of-the-box with a default installation of PackBench. The default template is available in the installation folder in `Resources\Reports\DefaultTemplate.docx`.

- **Save current as...**
  Saves the current custom template as a `.docx` file. If no custom template has been specified, the content of a default template will be copied instead.

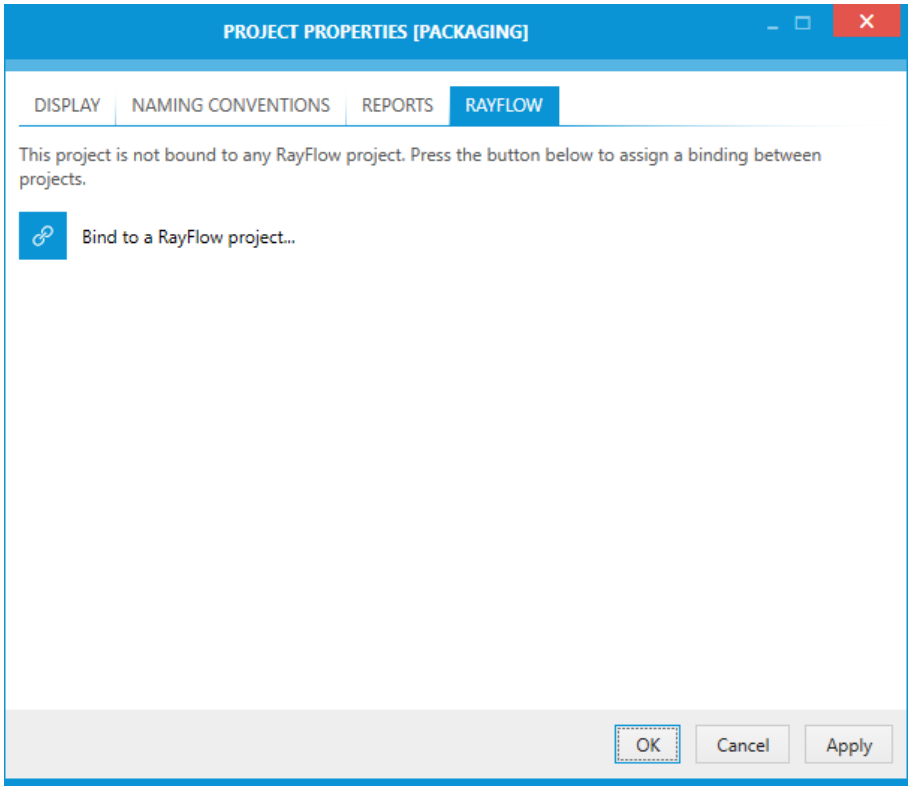
- **View/edit...**
  Downloads the current template as `.docx` file and opens it in default editor. Once the editor is closed, if the file has changed its content is uploaded again. Nothing will happen if you close the default editor without making any changes to the opened file.

## File Name and Format

This formatted text box contains a default name pattern used when a new report is generated. By pressing the ellipsis browse button [**...**] the **VALUE EDITOR** will open. For more on the **VALUE EDITOR** refer to the *Value Editor* chapter.

# RayFlow

The **RAYFLOW** tab is used to configure a binding between a RayFlow project and a PackBench project.




**Note:**
For this to work, a valid RayFlow instance must be running. Refer to section *Working with RayFlow* for more information.

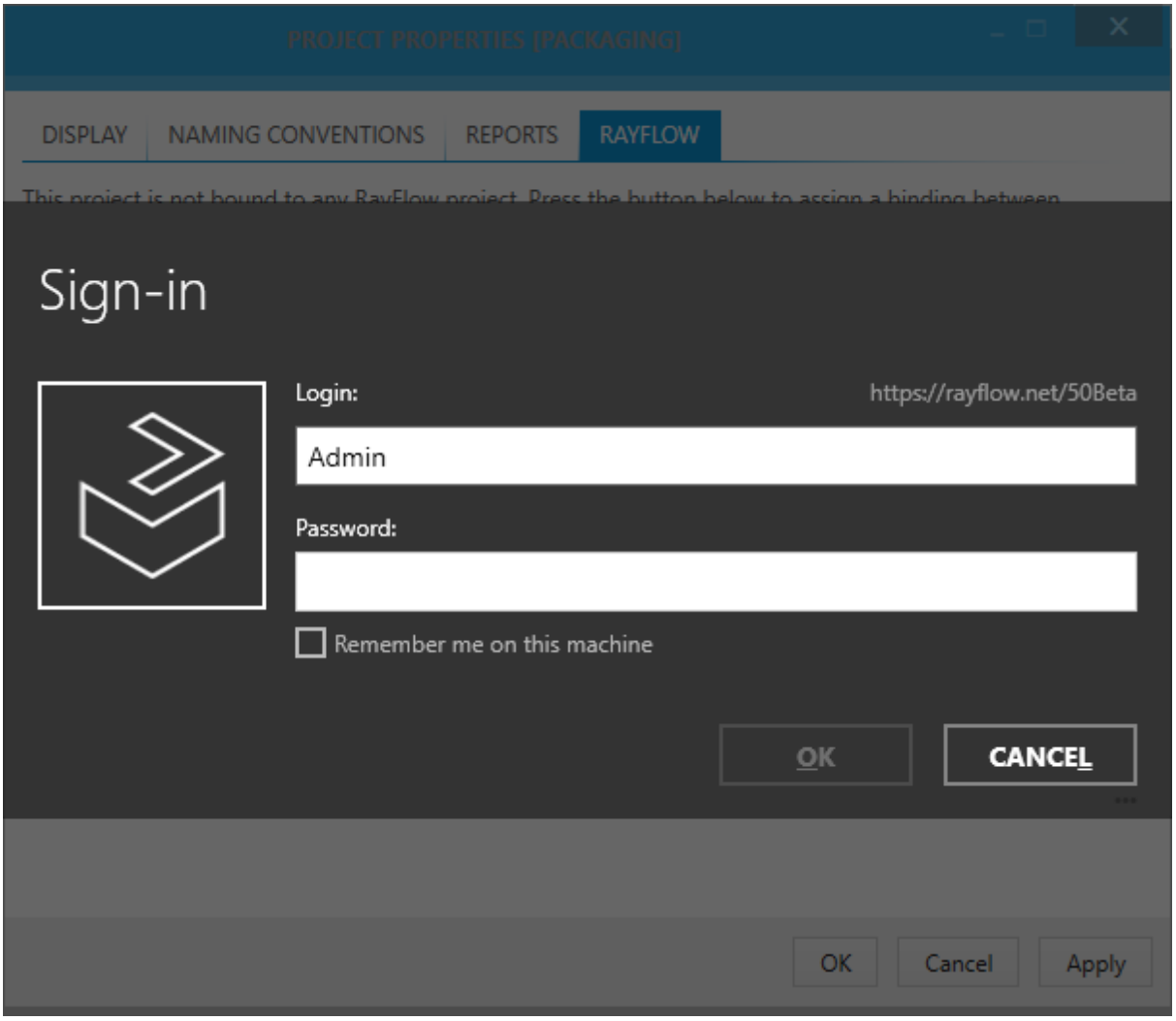
## In Order to Bind RayFlow Project...

1. Press **Bind to a RayFlow project...**

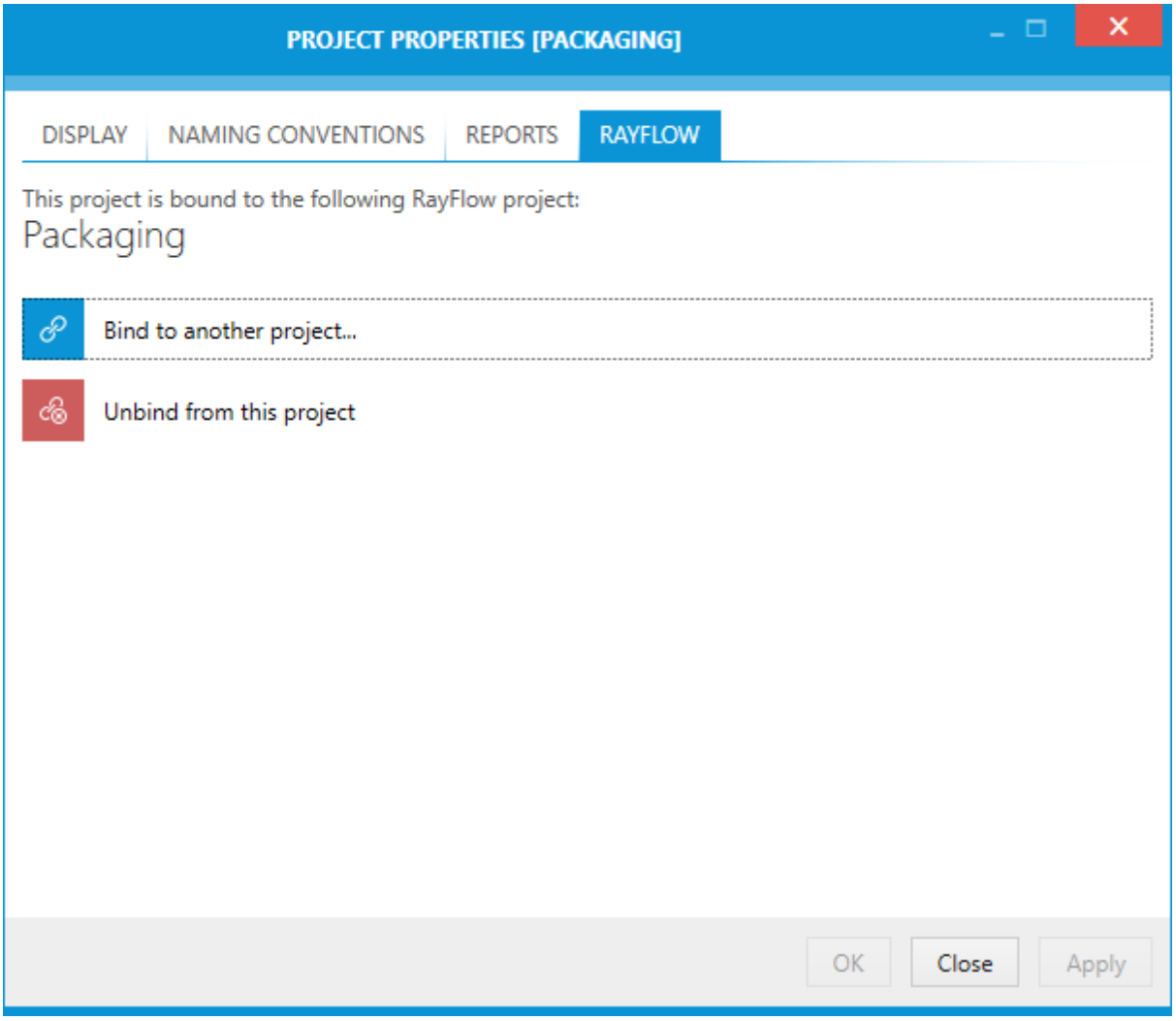
2. Enter the credentials for a running RayFlow instance and select the project

**Note:**
The login dialog will be shown even if already logged-in to RayFlow. This is to verify that the current user has permissions to see certain RayFlow projects.

3. Once authenticated, select the target RayFlow project and press **OK**.




4. The project name will be now shown in the properties dialog




## In Order to Change RayFlow Project...

To change the RayFlow project, press **Bind to another project** button. If not yet signed in, PackBench will ask for RayFlow credentials. Then, repeat the procedure covered in previous section.

## In Order to Remove Binding to a RayFlow Project...

To remove the binding to a RayFlow project, press **Unbind from this project** button. If not yet signed in, PackBench will ask for RayFlow credentials.

# Archiving Projects

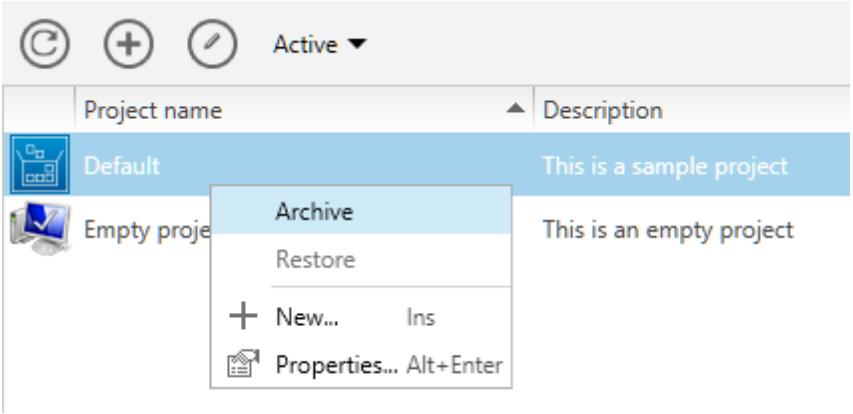
Projects cannot be removed once created, but they can be archived. Archived projects are by default not visible in the projects list, they are also never displayed in the project selector (once more than one project has been defined for the current instance).

# In Order to Archive a Project...

1. In the *project configurations view*, select the project to be archived.

2. Right click to open a context menu.

3. Click on **Archive**

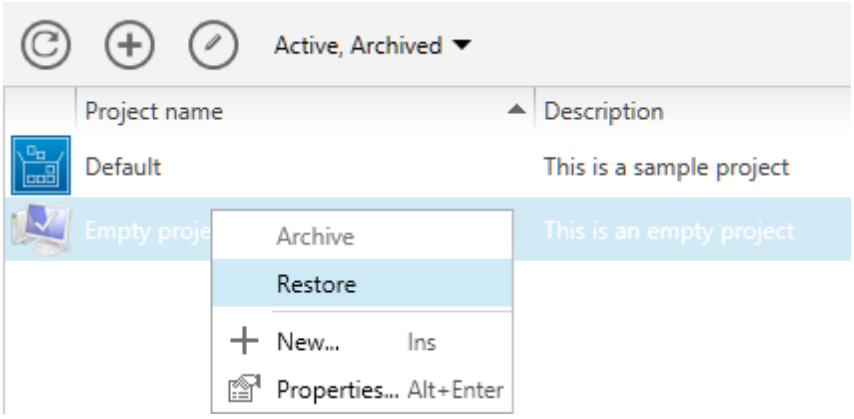



4. The project will be immediately archived, and if the filter is configured to hide archived projects, the item will disappear from the screen.

# In Order to Restore a Project...

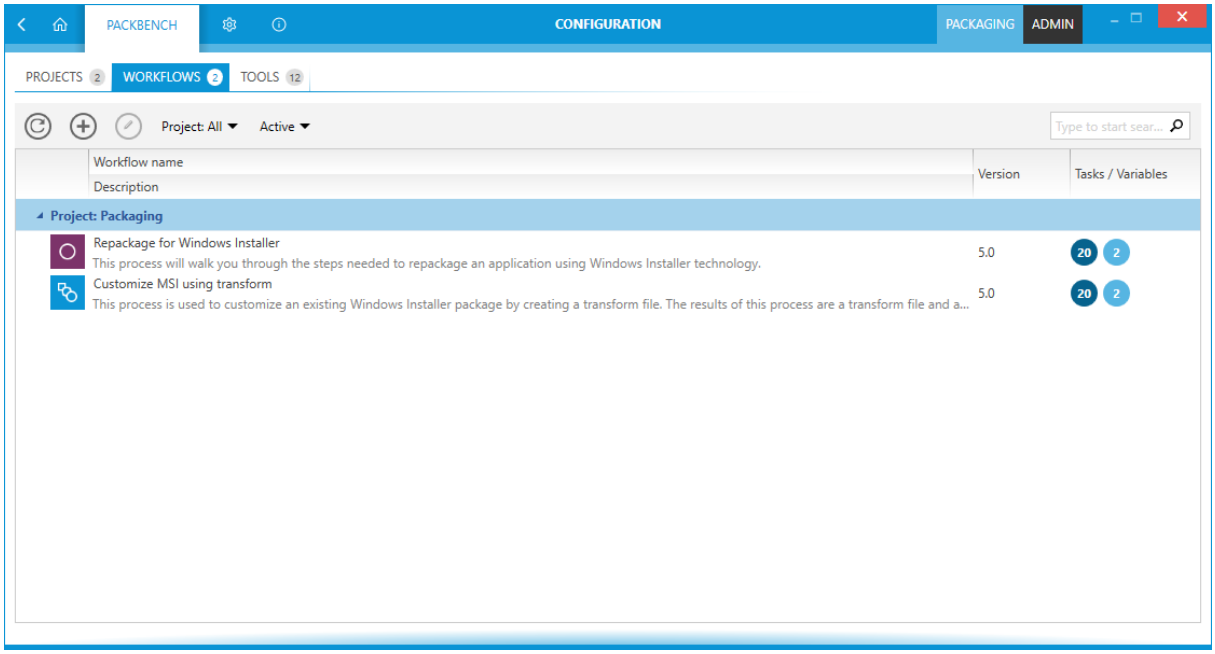
1. In the *project configurations view*, make sure the filter is set to **Archived** or **Active, Archived**.

2. Select the project to be restored.

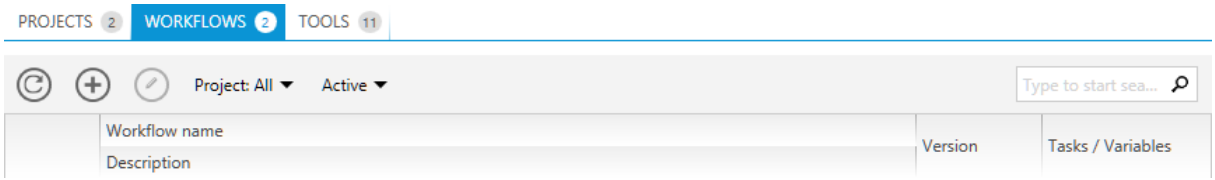


# Configuring Workflows

This chapter explains how to create workflows with PackBench and how to edit an already existing workflow.




To configure a workflow left-click on the **Configuration** tile in the **config** section of the Home Screen. The **CONFIGURATION** view is opened. Select the **WORKFLOWS** tab inside of the **CONFIGURATION** screen.




The **WORKFLOWS** tab shows an action bar, the header of the content area, and the content area where the projects are shown.

## Workflows Tab Header

The header of the tab is highlighted when the tab is selected. Furthermore, it shows a count of the workflows which are currently existing in PackBench. Archived workflows are still counted.

### Action Bar

There are six different abilities available in the action bar.

## Refresh this View Button

This button can be used to refresh the content area of the **WORKFLOWS** tab. If any changes have been made to any of the workflows, this button should be used to ensure that the changes are shown in the tab.

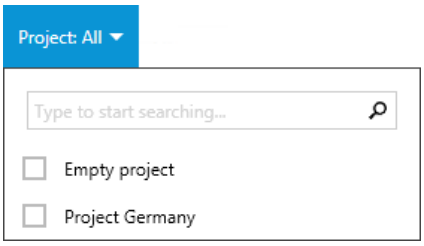
## Create New Workflow Button

Click this button to create a new workflow. For more information on this, refer to the *Creating Workflows* section of this chapter.

## Edit Tool Button

Click this button to edit currently selected workflow. For more information on this, refer to the *Editing Worflows* section of this chapter.

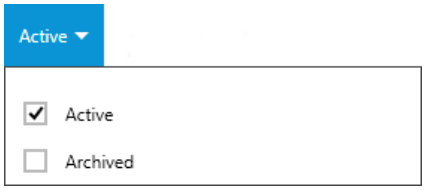
## Project




The Project button contains a dropdown menu with an integrated search function. A list of all projects of the workflows which are currently shown in the content area will appear. This list can be searched by using the keyword search field integrated into the dropdown menu to search for projects. This search field works like the keyword search field in the tab. To learn how to use the keyword search field continue with this *section*.

After selecting projects from the dropdown menu only workflows from the selected projects will be shown in the content area. Projects are chosen by enabling or disabling the checkbox related to the project. The title of the option will show the selected projects. If no project is selected, all projects will be shown. By default, no project is selected.

## Status




The **Status** button contains a dropdown menu which can be used to define which workflows will be shown in the **WORKFLOWS** tab. Available options in this menu are **Active** and **Archived**. Enable or disable an option by using the checkbox related to the option. If archived workflows are shown, they will be grayed out. The title of the option will also change to **Active, Archived** to clarify that all workflows are shown. By default, only active workflows are shown.

## Keyword Search Field

The keyword search field can be used to filter the workflows. Type the keyword into the search field. The workflows will be filtered for the keyword while typing.

For example, type "sample" into the search field an only workflows which contain the word "sample" in either the **Description** or the **Workflow name** will be shown.

### Sorting

It is possible to sort the content by column. Select the column and click on it to sort the workflows alphabetically. An arrow will appear next to the selected column header. To change the sort order simply click on it again. The arrow will indicate the order by either pointing upwards or downwards.

The columns themselves can be interchanged by drag and drop. Simply drag the column header and drop it at the target position. The columns can also be changed into rows by dropping the header of on of the columns underneath the other header.

### Content Area

In the content area the different workflows are shown. Next to the name of the workflow there is a number which is shown in white with a blue background.

These number represent respectively:

- The number of Tasks which are present in the respective workflow.
- The number of custom variables belonging to the workflow.

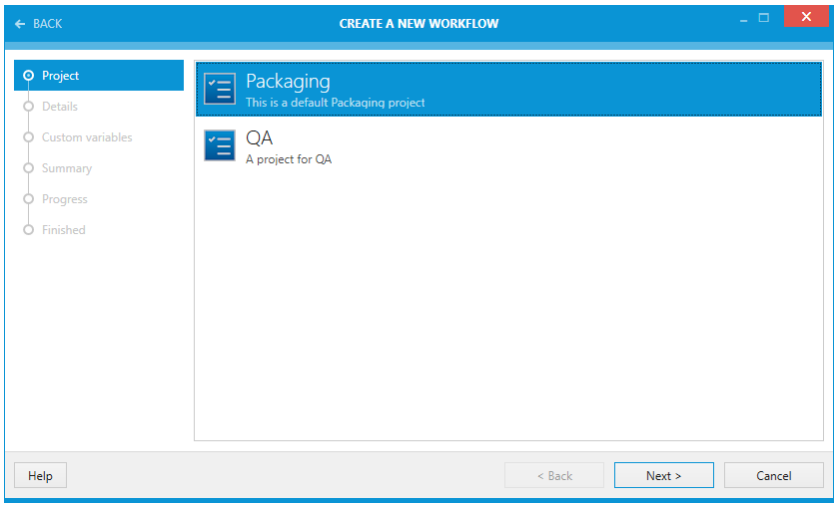
# Creating Workflows

## To Create a New Workflow...

1. To create a new workflow open the **WORKFLOWS** tab in the **CONFIGURATION** view and then either open the context menu and select **New...** or click on the **Create new workflow** button ( ⊕).

The **CREATE A NEW WORKFLOW** wizard is used to create new workflows in PackBench.

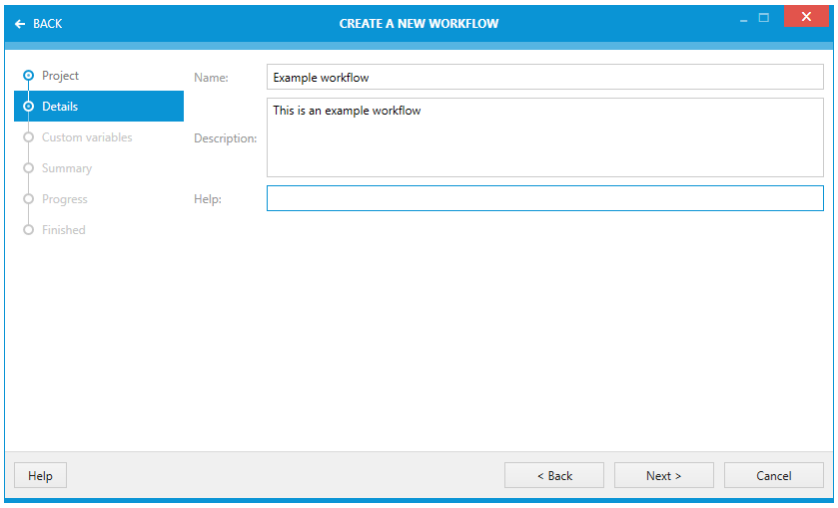
### Project Page




On the **Project** page, select the target project for the workflow. All currently active projects are shown here. Archived projects are not available for selection.

Click on the **Next >** button to continue to the **Details** page.

## Details Page




On the Details page there are three different fields available.

- **Name**: Enter a name for the workflow. This field is mandatory and may not be empty.
- **Description**: Enter a description of the workflow. The description is optional and can be left empty.
- **Help**: A short help text can be entered here. This field is optional and can be left empty.

Click on the **Next >** button to proceed to the **Custom variables** page.

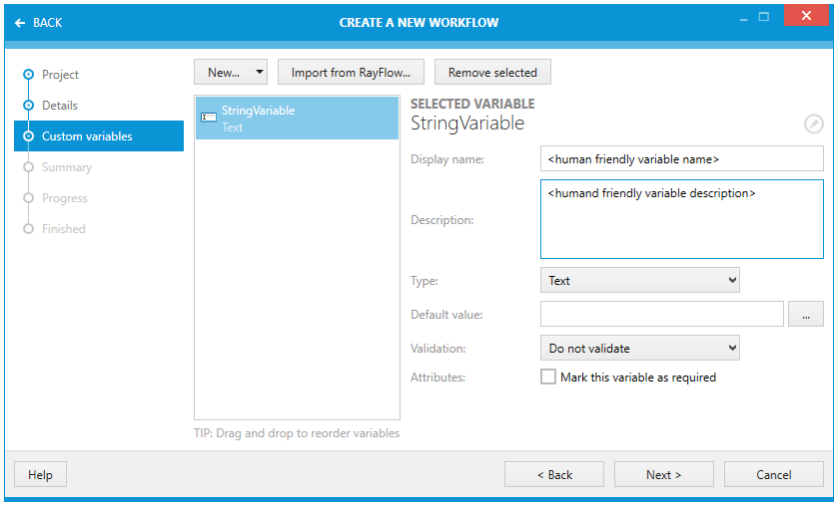
## Custom Variables Page

Custom variables extend the default set of standard properties. It is possible to define them here, or later once the workflow is configured and their tasks are set. For more information about variables, refer to information contained within the section *Custom Variables.*

The variables can be created manually, or imported from a RayFlow phase.

- To create a new value, press **Add custom variable** button. A new variable of the type string will be created and selected.
- To import variables from RayFlow, press **Import from RayFlow** phase. PackBench will ask to sign in to RayFlow and select the phase from which to import.

Once at least one variable is defined, the screen changes and shows a sidebar containing a list of all variables, and a right panel showing the details of the current selection.

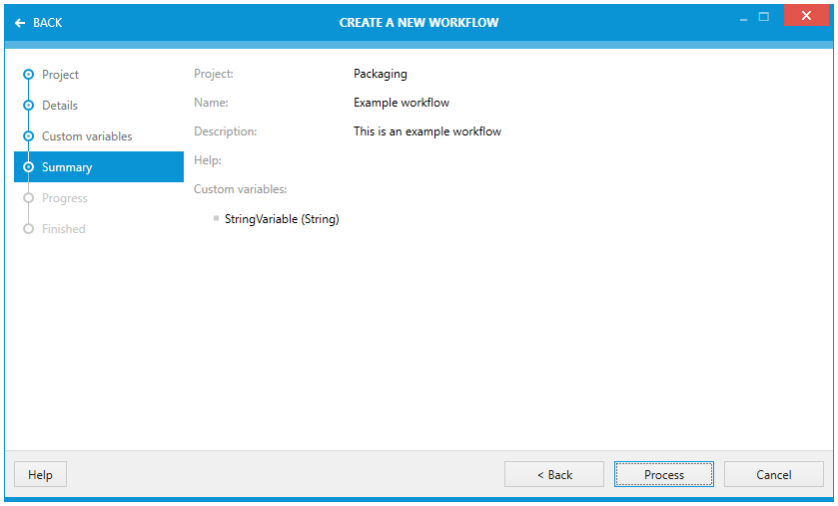



- To remove a variable, highlight it first and press the **Remove selected** button
- To reorder variables, use drag and drop technique using the left mouse button.

On the right site of the page the details of a selected variable are shown. The field varies, depending of the type of the variable selected. For detailed information on the custom variables, refer to the *Custom Variables* chapter.

Click on the **Next >** button to proceed to the **Summary** page.

## Summary Page




The **Summary** page shows a summary of all the settings that have been made on the previous pages. Verify the entries and click on the **Process** button to create the new workflow.

**Note:.**
The New Workflow wizard does not allow adding tasks to the workflow. This has to be done in post-processing, as described in chapter *Tasks*.

# Editing Workflows

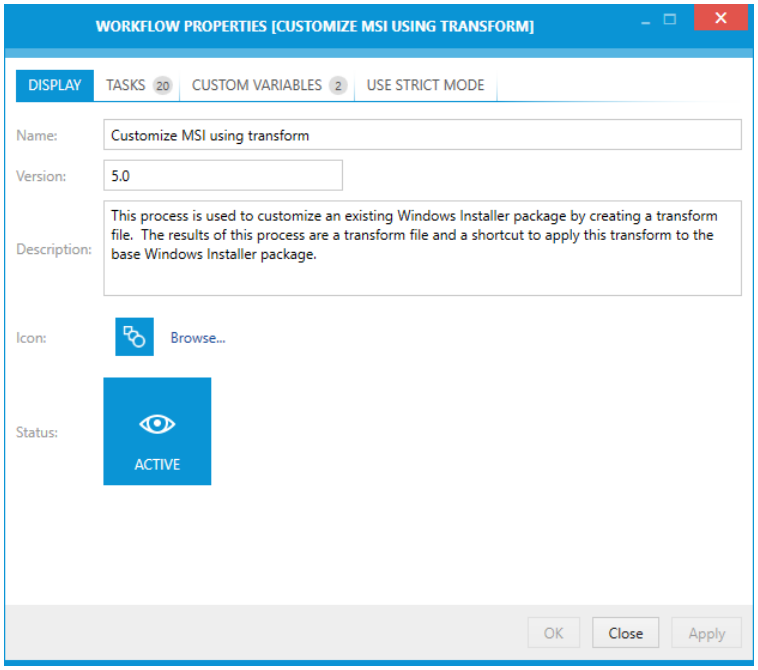
Workflows can be edited by opening the **WORKFLOW PROPERTIES** of the target workflow. They can be opened by clicking on the **Properties...** option in the context menu.

The dialog consist of four functional tab groups:

- DISPLAY
- TASKS
- CUSTOM VARIABLES
- ADVANCED

Click on the **OK** button to save the changes and close the window, click on the **Apply** button to save the changes without closing the window, or click on the **Cancel** button to dismiss the changes.

## Display




In the **DISPLAY** tab of the workflow configuration there are five different objects which can be configured.

### Name

Shows the name of the workflow and can be used to change the name. This field is mandatory for all workflows.

### Version

Shows the version of the workflow. This field is mandatory for all workflows and has to be a valid version string (for example 1.0, 2.3.4 or 3.45.67.890).

### Description

The description of the workflow can be entered or altered here.

### Icon

A small picture identifying the workflow. Press the **Browse...** button to select a supported icon/picture.

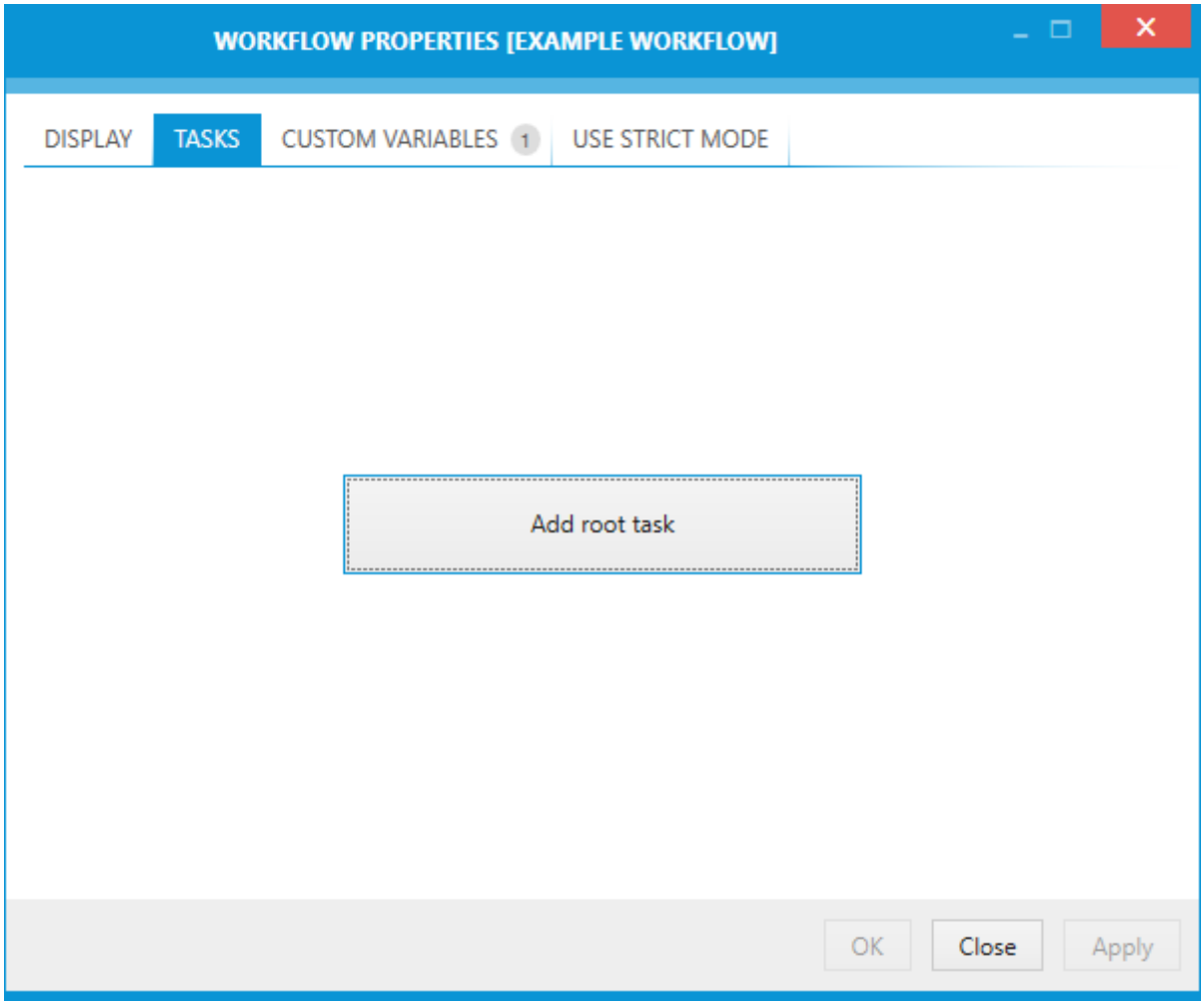
### Status

The status of the workflow can be defined by clicking on the button. It can be set to either **ACTIVE** or **ARCHIVED**. The current status will be shown on the button and the button will be highlighted if the current status is **ACTIVE**.

## Tasks

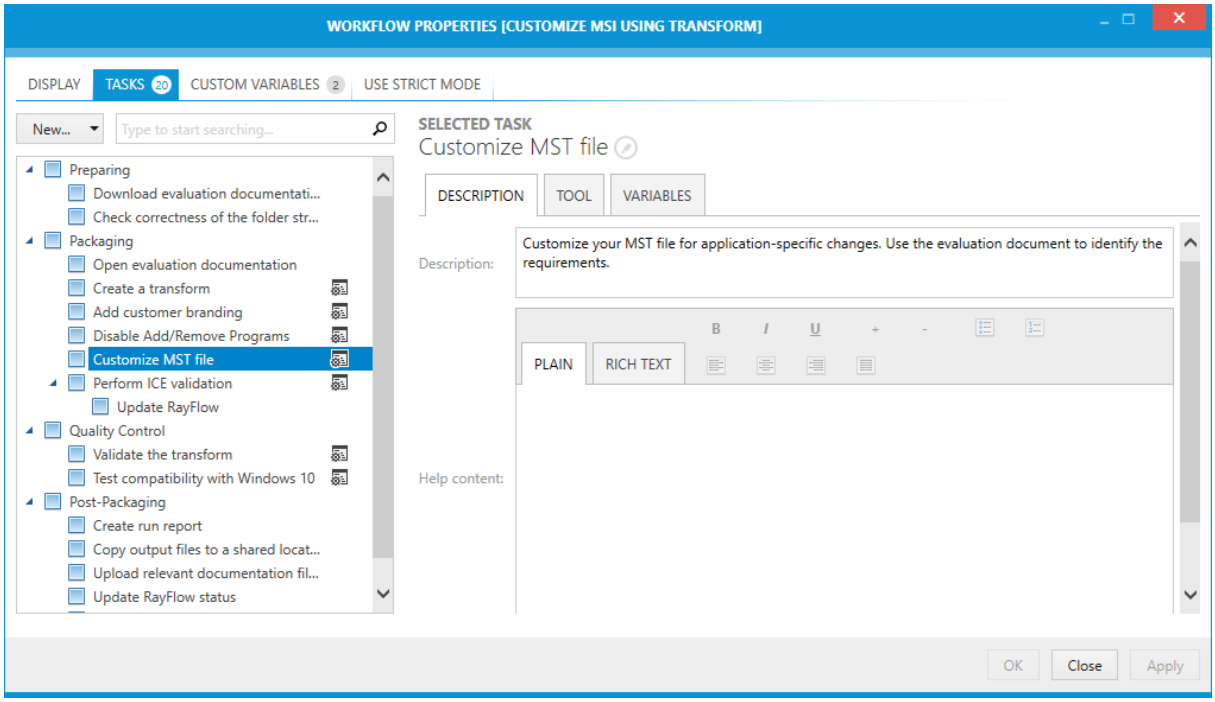
The header of the **TASKS** tab shows the total number of tasks that are currently contained in the workflow. Root tasks and subtasks are both counted as one task. If you have a workflow with seven root tasks and two of this root tasks each have two subtasks, the number shown in the header will be eleven. If no tasks exist in the current workflow, there is no number in the header and the tab is grayed out. The task can still be selected, but the task overview will be empty.

When no tasks are present, a simplified view of the tab is shown:




Click **Add root task** to create a first element in the task tree.

Once at least one task is already assigned, the view changes:



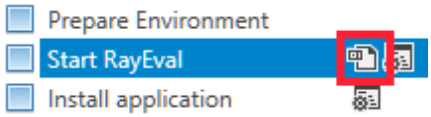

The area is divided in two different sections:

- The left panel shows all tasks assigned to the workflow, in a tree-style indicating parent and child relationships.
- The right panel shows the details of the current selection

# Task Tree List

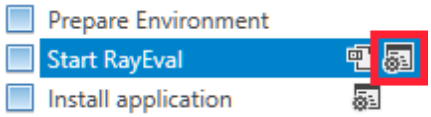
In the task overview, existing tasks are shown. The tree may contain additional icons, which have the following meaning:

- **This task displays variables**




Indicates that there are *variables assigned to this task.*

- **This task executes a tool**



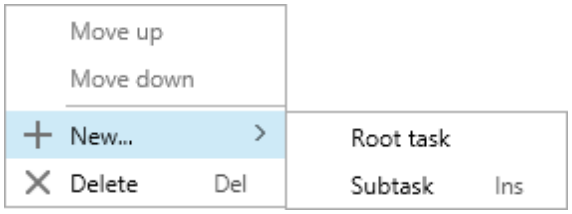

Indicates that there is *a tool assigned to this task.*

The task icon presents the state as follows:

Unchanged task
Edited task
Added task

## Creating New Tasks

New tasks are created by opening the context menu and hover over the **New...** option. The menu will expand and **Root task** and **Subtask** will become available for selection.




Select either **Root task** or **Subtask**. A task which is called task will be created at the selected position. Select the new task. It can now be configured in the task details on the right side of the tab.

**Tip:**
The hot-key button **Ins** can be used to create child tasks.

## Reordering Tasks

Tasks can be reordered by drag and drop. Simply drag the task and drop it into the new position. Through reordering root tasks can become subtasks and subtasks can become root tasks.

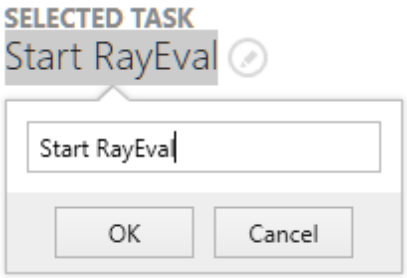
## Removing Tasks

In order to delete a task, open the context menu and press **Delete**. When removing tasks having sub-tasks, PackBench shows a confirmation dialog. Once a task is deleted, all its subtasks will be deleted as well. In order to preserve them, they have to be moved to another task (see *Reordering tasks*) prior to deletion.

**Tip:**
The hot-key button **Del** can be used to delete tasks.

# Task Name

The name of the currently selected task is displayed in the top right corner of the **TASKS** tab. The name of the task can be changed by clicking on it. Changes can be accepted by pressing **OK**, and discarded using **Cancel** button.




**Tip:**
Tasks can also be renamed by highlighting them in the tree list, and pressing **F2**.

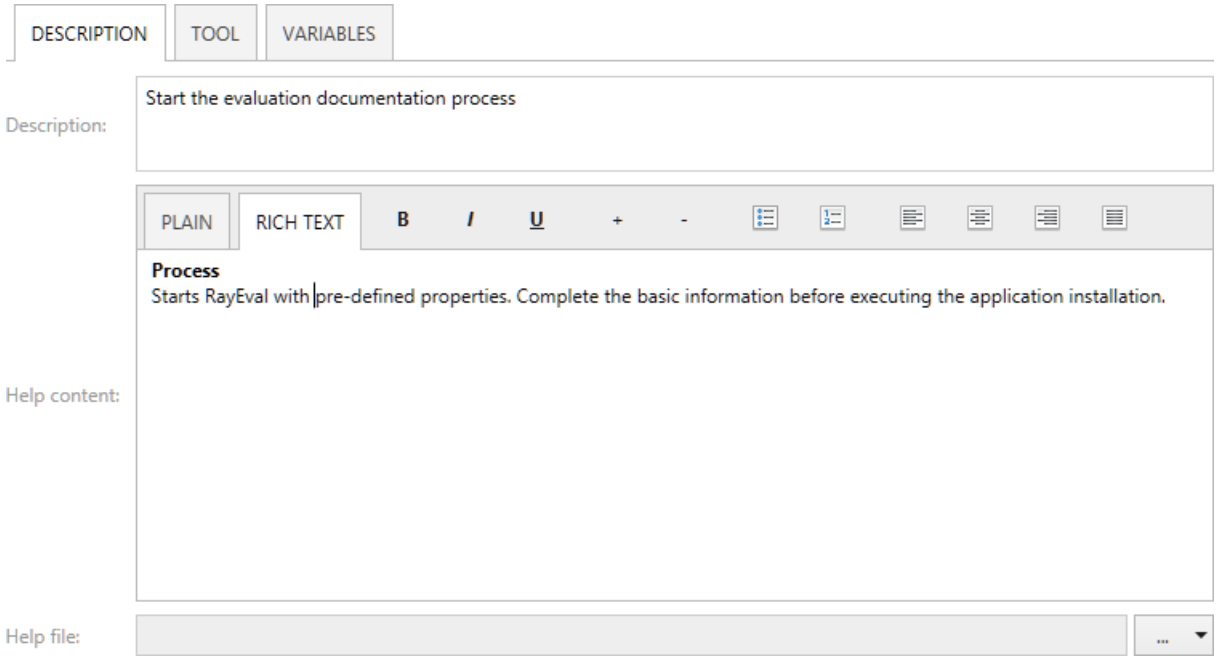
# Task Details

Task details section consists of three tabs:

- DESCRIPTION
  Defines description, extended notes and a help content for a given task.
- TOOL
  Defines tool assignment for a given task.
- VARIABLES
  Defines relations between a given task and custom variables.

## Changing Task Appearance

The **DESCRIPTION** tab defines the appearance of the task and optionally a help content to document the work..




## Taskname

The task name can be changed by clicking on the task name. A window will appear. Type the name into this window and click on the **OK** button.

## Description

The description of the tasks belongs here. Type in the description and press the **Apply** button to save the data.

## Help Content

Information that is helpful for the task belongs here. The text can be entered into two different tabs. The **PLAIN** and the **RICH TEXT** tab. The **PLAIN** tab can be used to enter a text without formatting it. The **RICH TEXT** tab can be used to enter text and format it. Type in the respective information and press the **Apply** button to save the data.

## Help File

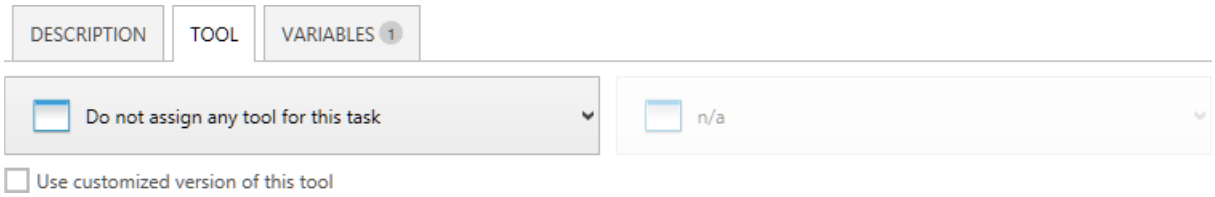
A help document can be attached to the task under this option. Click on the Browse button [**...**]. Two options will become available. Choose if the file should be attached by using the **From local folder...** option or by using the **Using variables...** option.

- **From local folder...**: Choosing this option will open the **Open** window. Browse to the location where the file can be found, select the file, and click on the **Open** button to add the file to the task.
- **Using variables...**: Choosing this option will open the **Value Editor**. For information on how to use the **Value Editor** refer to the *Value Editor* chapter.

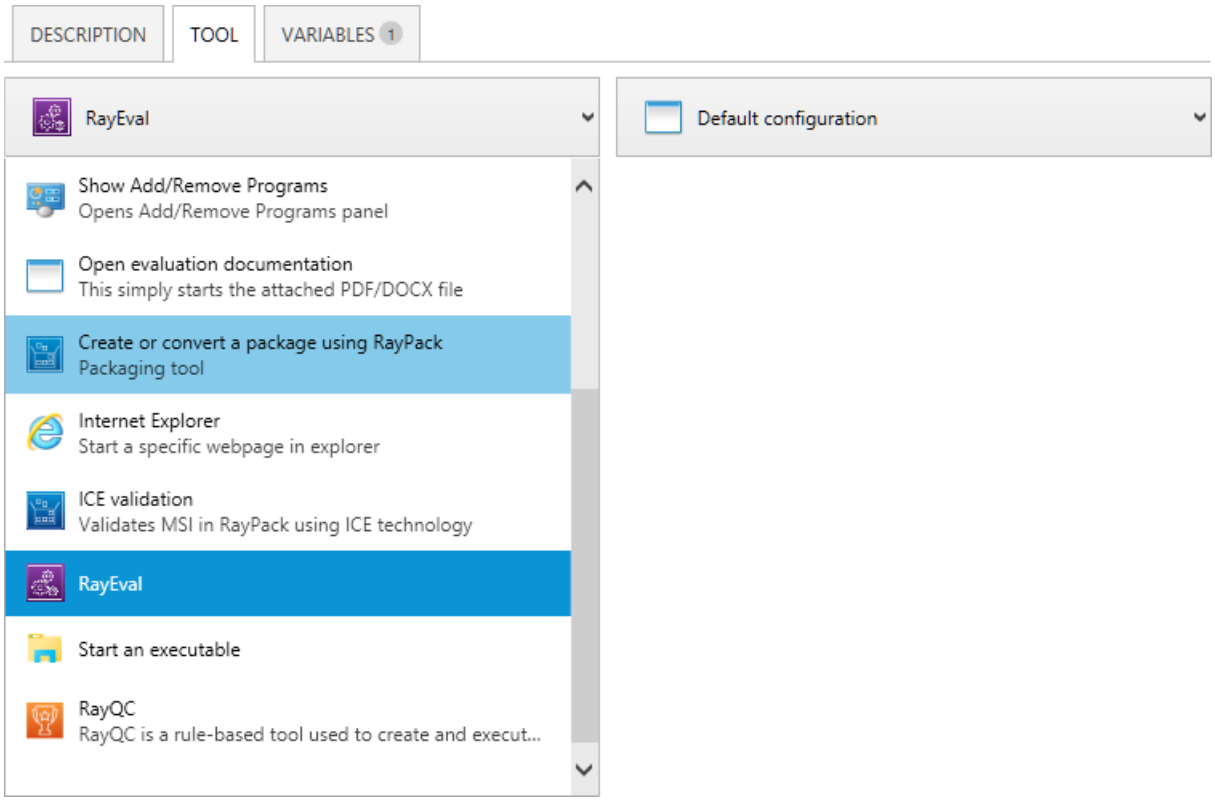
Select the file or the value and click on the **Apply** button to save the changes.

## Configuring Task Tools

The **TOOL** tab defines the link between tools already defined in this instance and a given task.

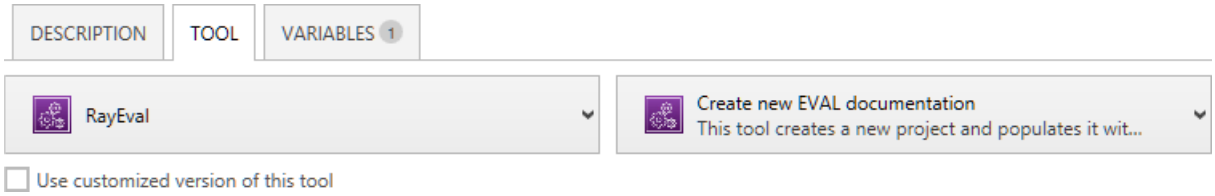



A tool can be assigned to a task. To assign a tool click on this field. By default, it is set to **Do not assign any tool for this task**. After clicking on this field a dropdown menu will open.




The dropdown menu contains a list of tools from which the tool that is to be assigned to this task can be chosen.

For some tools, additional configurations are available. The configurations are displayed in the second drop-down. If there are no configurations available for a tool, the second-drop is disabled. Configurations can be used to select a specific variant of a given tool (see chapter *Configurations* for more information). For example, the following choice:




means that the tool **RayEval** will be assigned to this task, and when started the command configured in configuration **Create new EVAL documentation** will be executed.

# Removing Tool Assignment

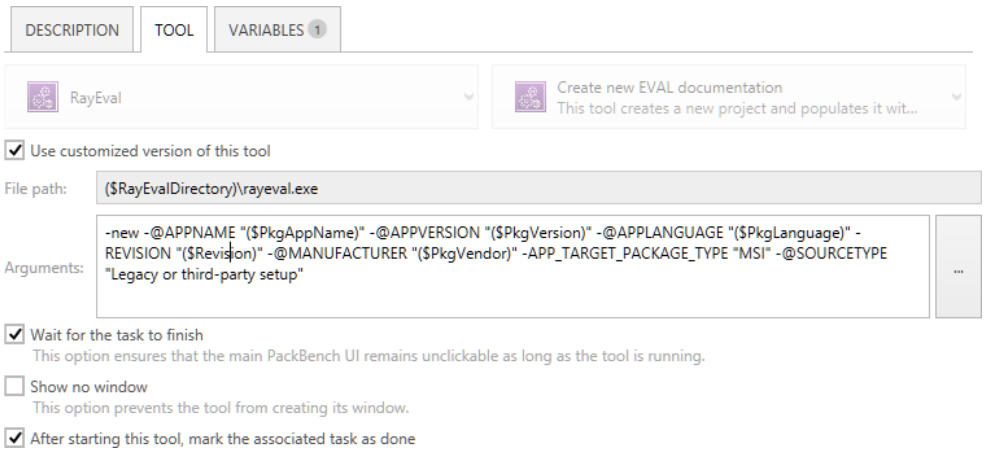
In order to remove the assignment, select option **Do not assign any tool for this task**.

# Overriding Tools

Frequently, default tool configurations are not 100% suitable to finish the specified task. This can be due to number of reasons, for example:

- The default configuration contains placeholders in the tool so that certain strings have to be replaced with workflow-specific variables
- Certain settings have to be adjusted on workflow basis, leaving the generic version of the tool intact.

In order to override certain tool or tool configuration settings, check the **Use customized version of this tool** checkbox. A new section will be revealed, containing relevant customization options:

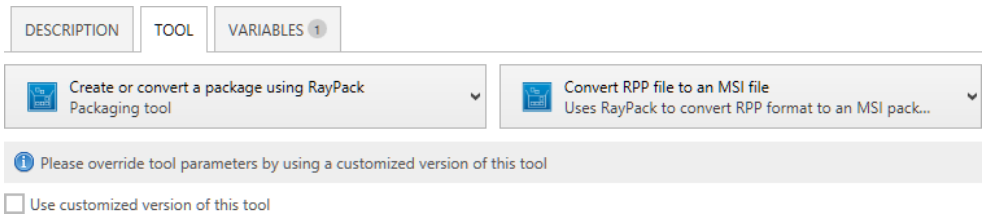



The options to override are similar to the configuration of a tool configuration (see chapters *Command* and *Configurations* for more details of how to use them).

**Note:**
Tools are global. By overriding the tool and simply reusing the existing command line and attributes you can create a "frozen state" of a tool which will be available in the workflow, even if the original tool changes in the future. By doing so, you make the task's tool independent.

# Recommendations from Workflow Creator

Workflow creator might have already suggested that a tool requires override or not, from the very same reasons as already outlined in the previous section. When a certain tool or its configurations have been configured in this way, when using the tool in the task view the following suggestion will be shown:

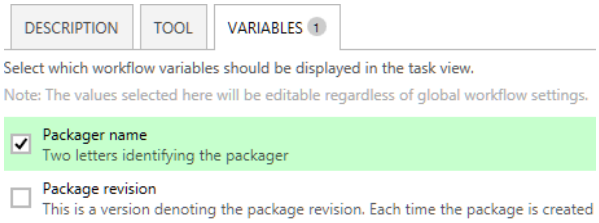



**Note:**
It is still possible to decide not to override the tool. However, ignoring the recommendation may lead to a tool that does not work, or works incorrectly.

## Configuring Task Variables

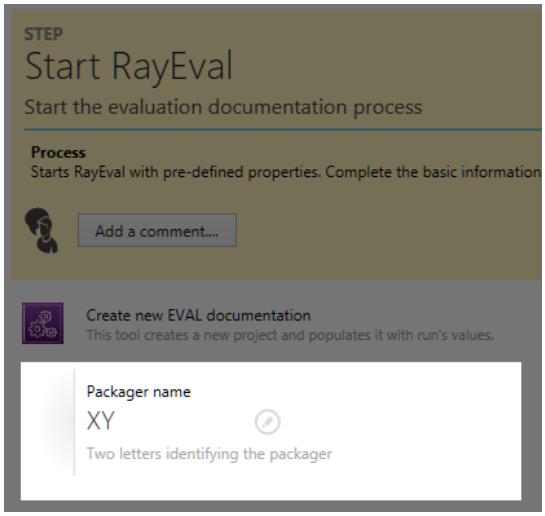
The **VARIABLES** tab defines relations between a given task and custom variables.

The tab shows a list of variables. defined in the *Custom Variables* tab. Any variable can be assigned to a task by ticking a checkbox next to its name:



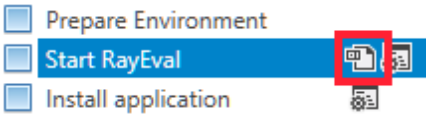

Variables assigned to a task are marked using a green background and a checkbox. Additionally, the number displayed in the tab header informs, how many variables are assigned to a task.

If a variable is assigned to a task, in the Run view the task view displays it under the tool section:




**Note:**
Variables assigned to a task can be always edited within the task view. The flag *Allow changing custom variables after a run is started* has no effect on variables displayed in the task view.

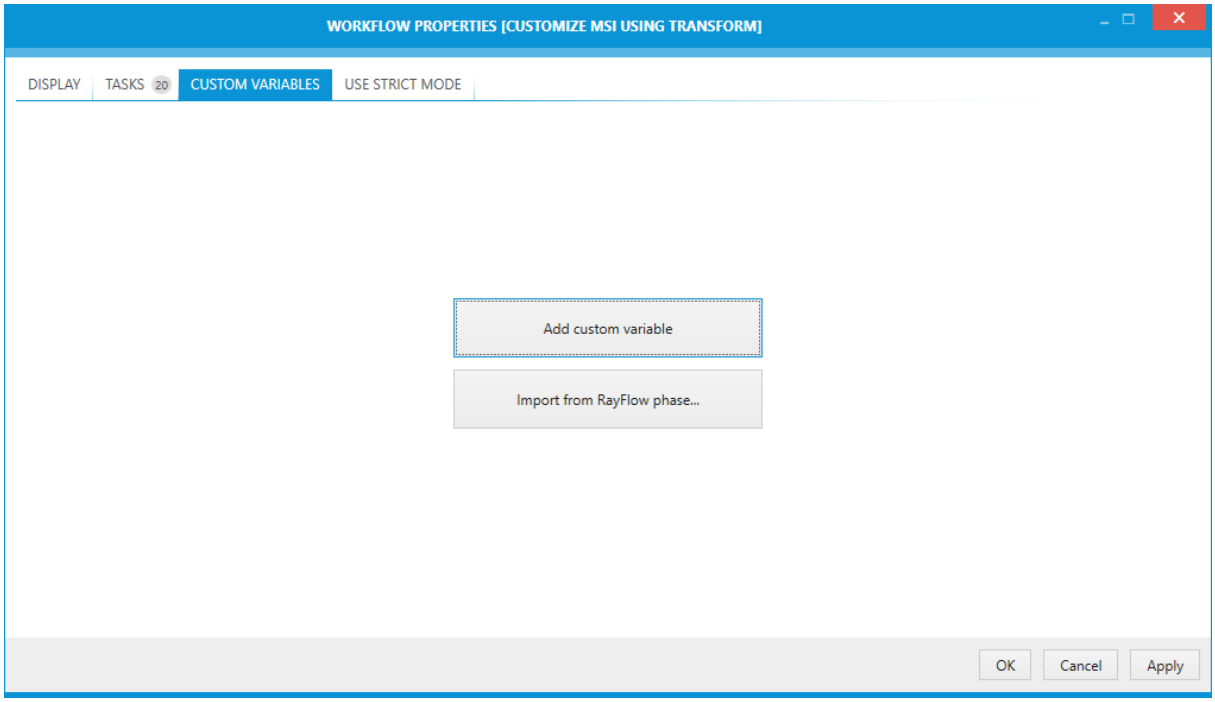
Once a task contains any variable assignments, an additional icon is shown in the sidebar:



# Custom Variables

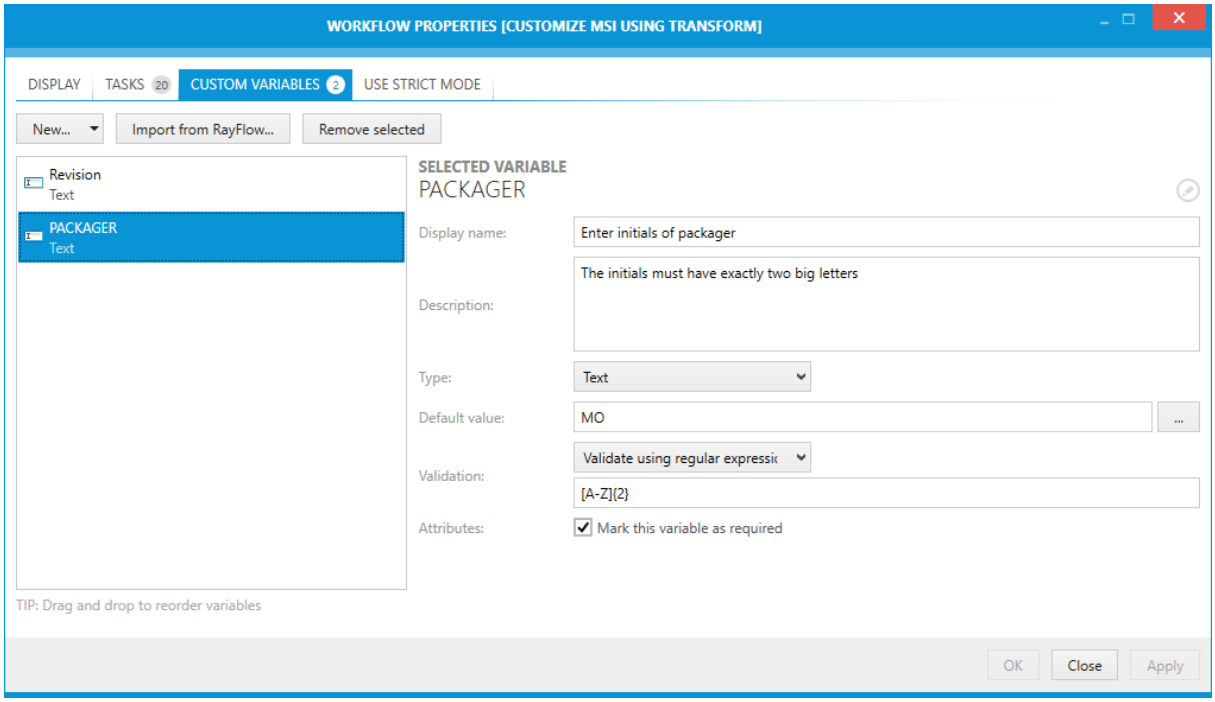
The header of the **CUSTOM VARIABLES** tab shows the total number of variables that are currently assigned to the workflow. If no variable exists in the current workflow, there is no number in the header and the tab is grayed out.

When no variables are present, a simplified view of the tab is shown:




- To create a new value, press **Add custom variable** button. A new variable of a type string will be created and selected.
- To import variables from RayFlow, press **Import from RayFlow** phase. PackBench will ask to sign in to RayFlow and select the phase to be imported.

Once at least one variable is already assigned, the view changes:




The area is divided in two different sections:

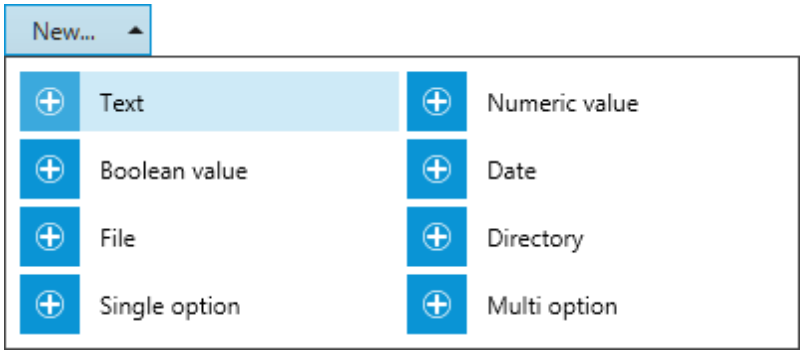
- The left panel shows all variables assigned to the workflow.
- The right panel shows the details of the current selection

# Variables List

In the variables overview, existing variables are shown. An icon representing the type of the variable is shown next to its name.

## Creating New Variables

New variables are created by pressing **New...** menu.The menu will expand and several options (for different variable types) will become available for selection.




Select the required type of a variable. A new variable will be created and selected. It can now

be configured in the variable details on the right side of the tab.

### Reordering Variables

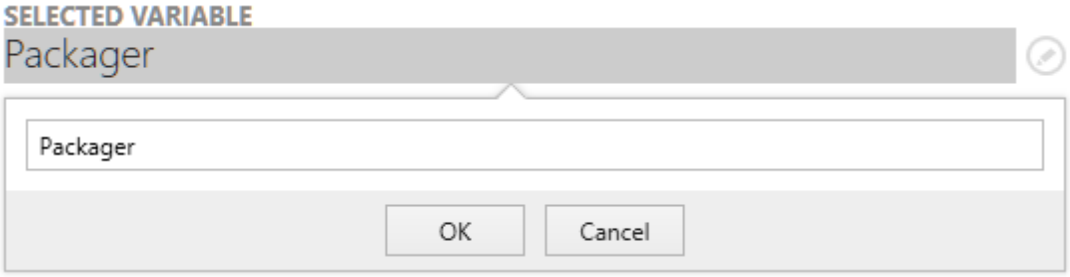
Variables can be reordered by drag and drop. Simply drag the variable and drop it into the new position.

### Removing Variables

In order to delete a variable, open the context menu and press **Delete**.

## Variable Name

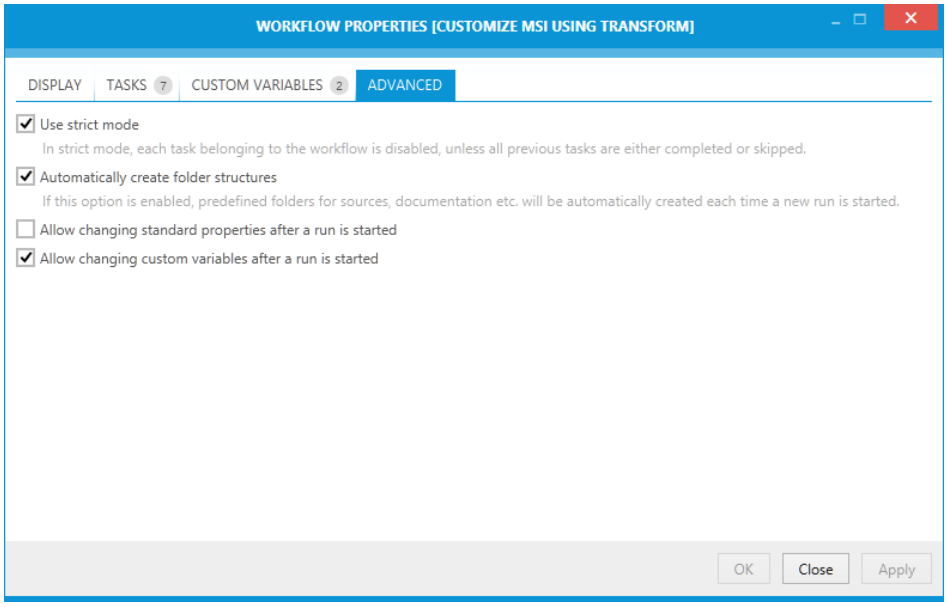
The name of the currently selected variable is displayed in the top right corner of the **CUSTOM VARIABLES** tab. The name of the variable can be changed by clicking on it. Changes can be accepted by pressing **OK**, and discarded using **Cancel** button.




## Variable Details

The content of the details panel depends on the type of the variable. Chapter *Custom Variables* contains a list of supported types and description of configuration options for each of them.

# Advanced




The **ADVANCED** tab contains advanced options which define which changes are possible after a run based on the current workflow has been started:

## Use Strict Mode

Using strict mode means that in order to proceed to any workflow task and mark it as done, all previous tasks have to be finished (or skipped) beforehand. Additional rules enforced by using the strict mode:

- The first workflow task is always enabled.
- If a task is marked as done or skipped, the next task on the same level is enabled
- If a parent task is enabled, its first sub-task is enabled as well..
- In order to proceed to the next task, all previous tasks including their sub-tasks have to be done or skipped.

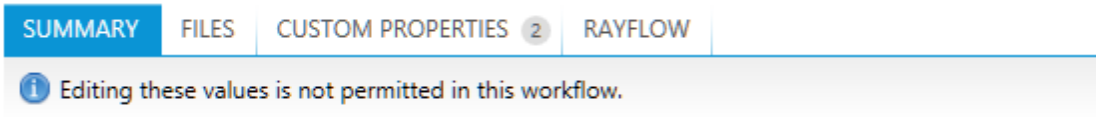
If strict mode is disabled, none of these rules apply and users are generally free to mark the completion as they want.

## Automatically Create Folder Structures

This settings control, whether the full set of default subfolders will be created in the package directory each time a new run is started. The names of the folders can be configured in *Naming conventions* in the Project Configuration settings.
By default, this option is enabled. If no subfolders shall be created in advance (but rather only as tools require them) then opt this option out by unselecting this checkbox.

## Allow Changing Standard Properties after a Run is Started

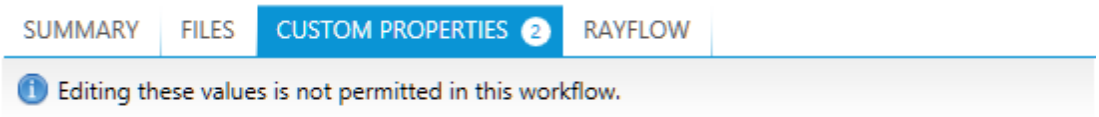
By default, standard properties (like name, version, manufacturer, package name etc.) are read-only once a run is started. This means they can be edited in the **New Run** wizard only and further changes are disabled due to the consistency rules. User will be presented with the following note in the Run view:




By enabling this option, standard properties are also editable. This option is disabled by default and should be used with caution.

### Allow Changing Custom Variables after a Run is Started

By default, custom variables are editable even once a run is started. By using this option, custom variables will not be editable anymore, and the following message will be shown in the Run view:




This option is disabled by default, which means that custom variables are editable.

**Note:.**
This option does not affect the editing capabilities of variables within the task view. Any variable associated with a task (as shown in chapter *Configuring task variables*) can be edited from its task view.

# Cloning Workflows

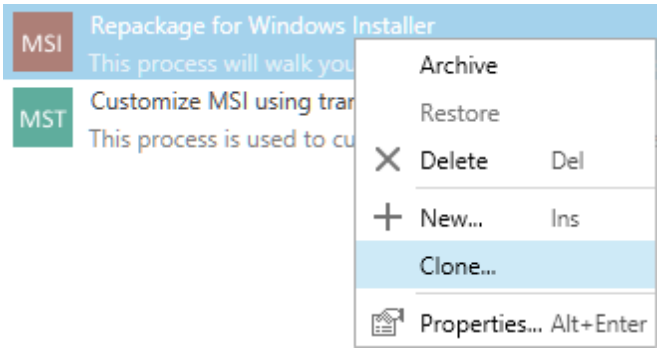
A workflow can belong to a single project only. Cloning a workflow may be useful if:

- The same workflows are to be duplicated and used it in different projects.
- A base workflow exists that should be used to create a new one
- To create a new version of the same workflow, just with increased version and minor changes

## In Order to Clone Workflow...

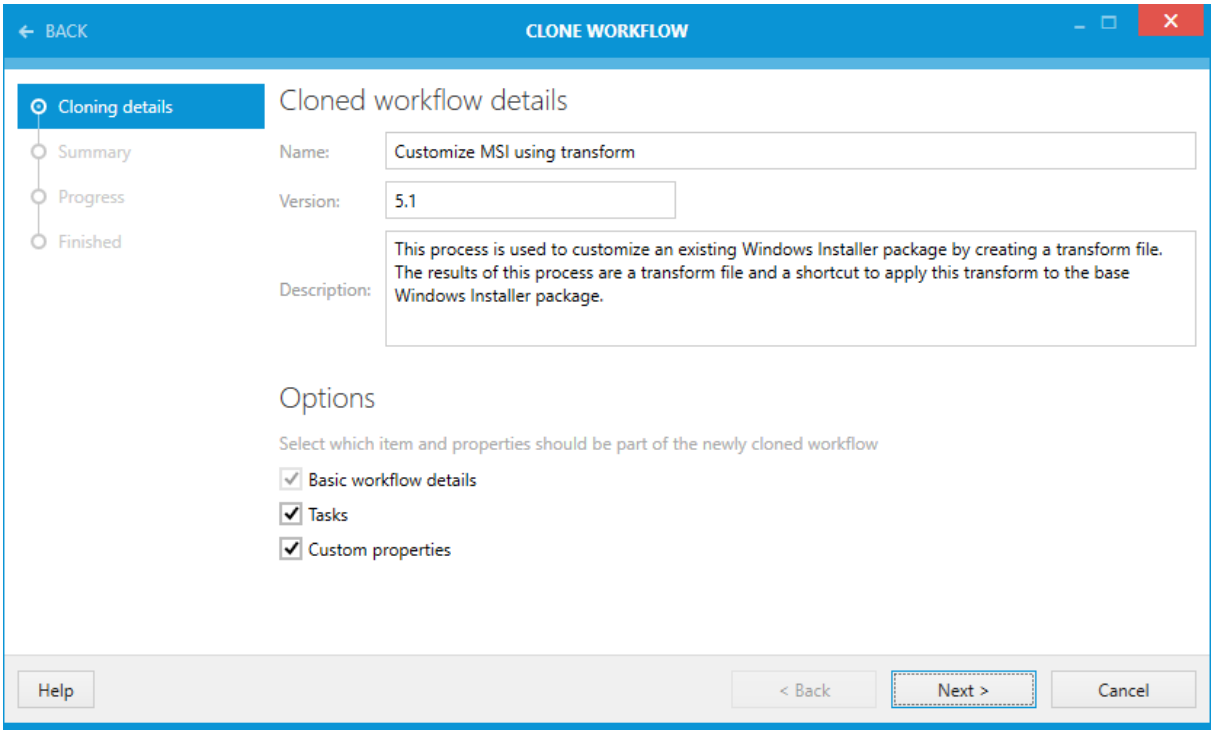
1. Select the workflow to be cloned in the *Workflows* view.

2. Right click to open a context menu.




3. Click on **Clone...**

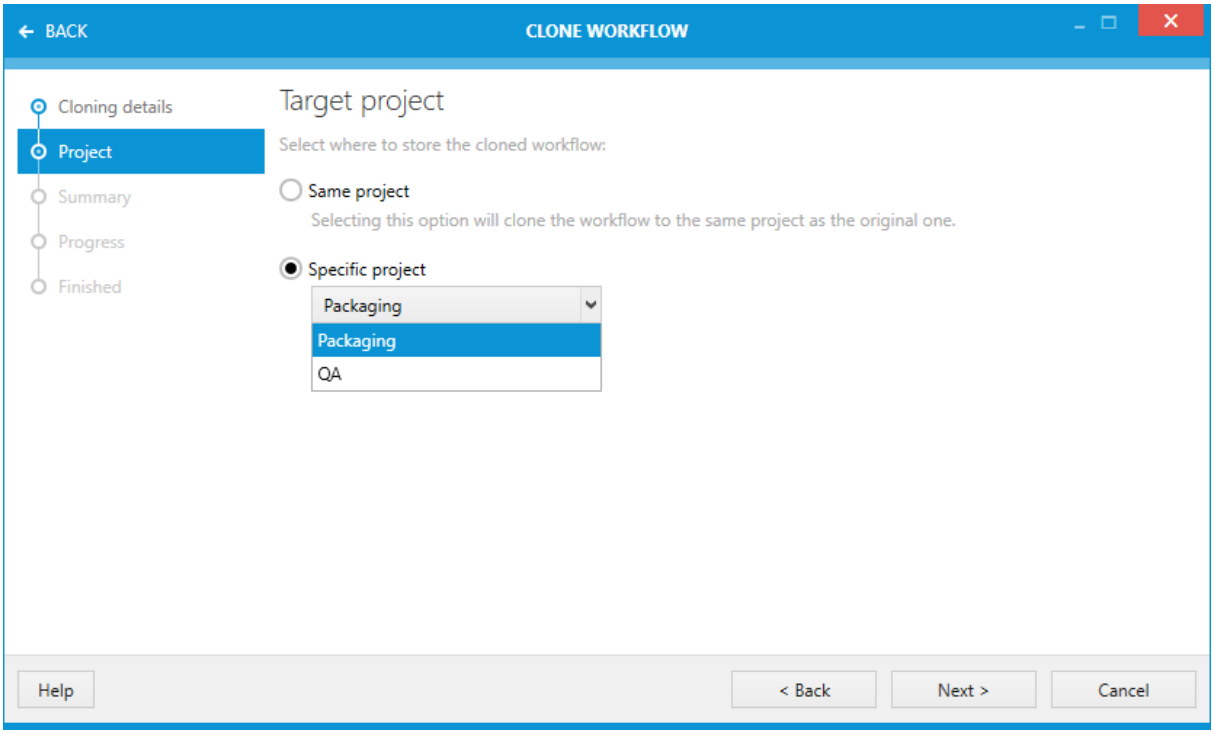
4. The **Clone Workflow Wizard** will be shown.




5. By default, values like name, version and description are based on the current workflow. The version is automatically incremented to indicate a new version, but may be reverted to the original or set to any arbitrary value.

6. Select which additional workflow properties are to be copied:
   a. Tasks - when this option is selected, all tasks will be copied to the new workflow
   b. Custom properties - when this option is selected, all custom variables will be copied to the new workflow

7. Press **Next >** to go the project selection page.




8. Select **Same** project to save the cloned workflow in the same project as the original one. Select **Specific project** to clone workflow to another project. The drop-down menu can be used to specify which project is to be used as a clone target.

9. Press **Next >** two times to start cloning. After the workflow is cloned, you can customize its properties, tasks and variables using the workflow editor as described in chapter *Editing Workflows*.

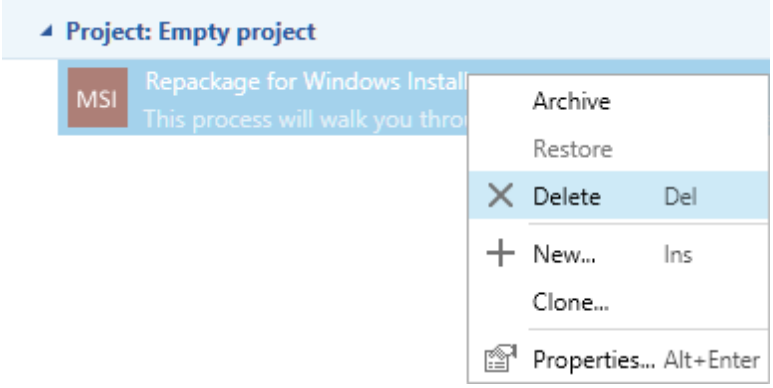
# Deleting Workflows

## In Order to Delete a Workflow...

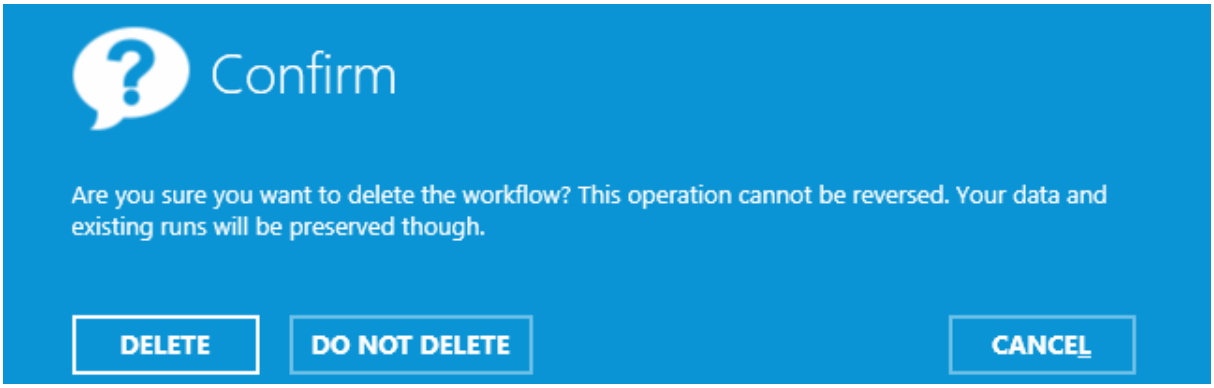
1. Select the workflow to be deleted in the *workflows view*.

2. Right click to open a context menu.

3. Click on **Delete**




4. Confirm deletion of the workflow.




5. The workflow will be deleted. All runs based on that workflow will be unaffected by this operation. If the runs should also be deleted, refer to chapter *Deleting runs*.

**Tip:**
The keyboard shortcut **Delete** can also be used to delete a workflow without going to context menu.

# Custom Variables

In PackBench it is possible to define custom variables of the following types:

- *Text*
- *Boolean value*
- *Numeric value*
- *Date*
- *Single option*
- *Multi option*
- *File*
- *Folder*

Depending on the type of the custom variable different options are available for their creation, but there is also a number of options which is identical for all types of custom variable regardless of their type.

## Variable Name

The name of the variable can be configured by clicking on the name. A dialog will open in which the name can be changed. The name of a variable must start with either a letter or an underscore and is not allowed to contain spaces or other characters than alphanumeric characters..

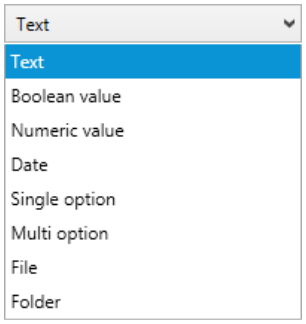
## Display Name

This is where the display name of a variable is defined. The display name is the name for the variable which will be displayed in a run. The display name may contain any type of character.

## Description

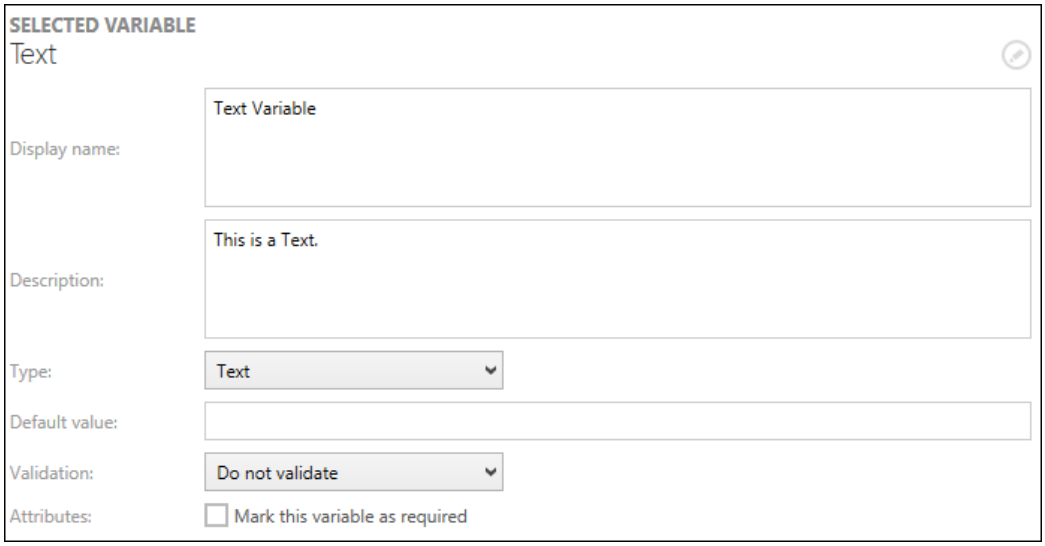
The description of the variable should be entered into this field.

## Type




The type of a variable is chosen by selecting the type from the dropdown menu shown above. The type of a variable defines any further options which are available for the configuration of a variable.

# Text




A variable of the type **Text** offers the following additional options for configuration:

## Default Value

This is where the default value of the variable is defined. The default value can be left empty or consist of any characters.

The default string value can contain run variables (see chapter *PackBench variables*). Standard values are resolved as they are changed. For example, to define that a variable has to contain product version and language by default, use the following default value:

```
($PkgLanguage)_($PkgVersion)
```

## Validation

There are two options available for this setting:

**Do not validate**: If this option is chosen, there will be no validation for this variable.
**Validate using regular expressions**: If this option is chosen, a new field will appear into which the expression used for validation can be entered. By default, the content of the field will be ^.*$. To learn more about the usage of regular expressions refer to the chapter *Regular expressions.*

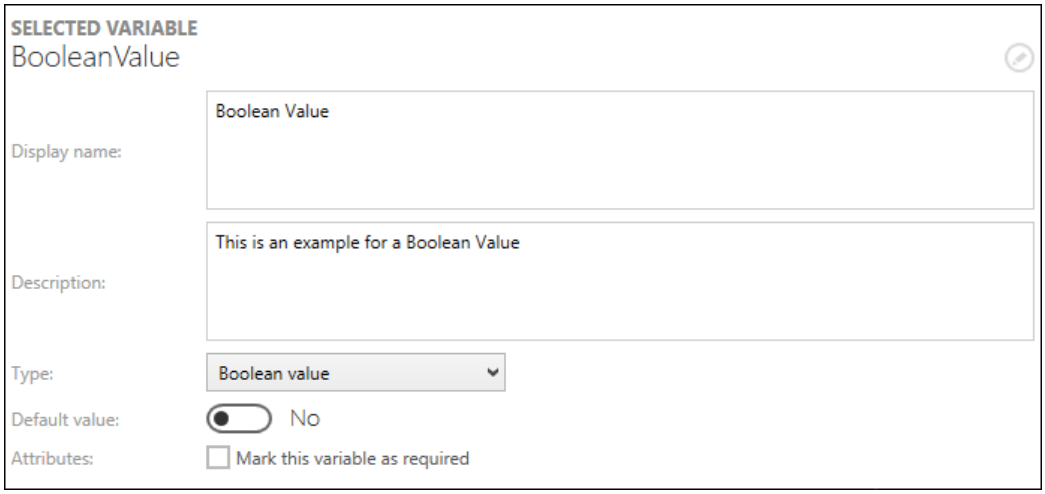
Below are a couple of frequently used expressions:

| | |
|---|---|
| `^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$` | A version in format #.#.#.# |
| `^(?i)[{(]?[0-9A-F]{8}[-]?([0-9A-F]{4}[-]?){3}[0-9A-F]{12}[)}]?$` | A GUID |
| `^[A-Z]{2}$` | Initials, using two capital letters |

## Attributes

By checking the **Mark this variable as required** checkbox a variable of the type **Text** can be defined as a mandatory variable. This means, that a valid value has to be entered into this field during a run.

# Boolean Value




A variable of the type **Boolean value** offers the following additional options for configuration:

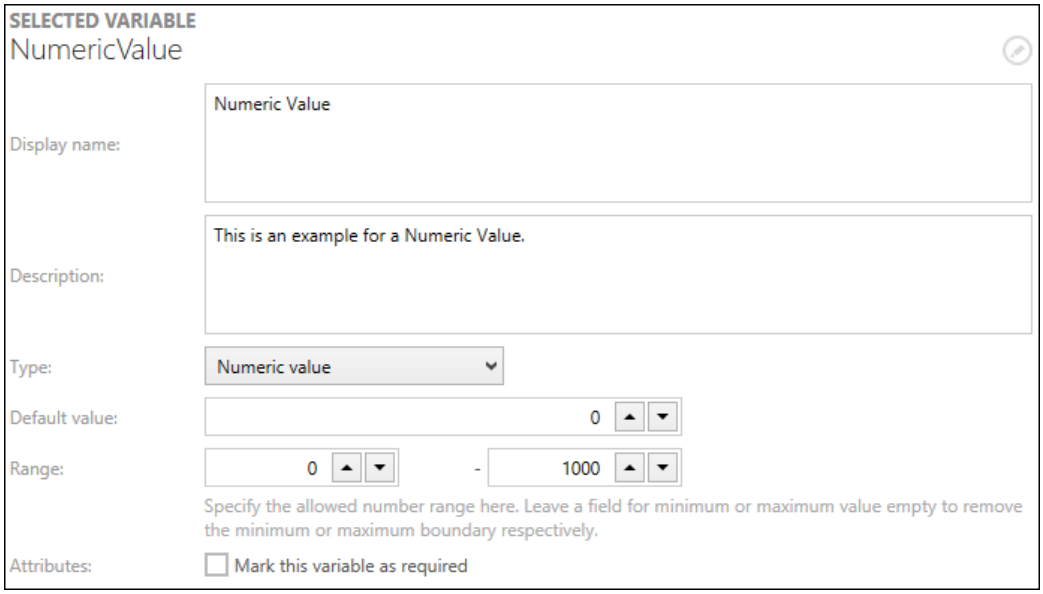
## Default Value

This is where the default value of the variable is defined. The default value of the variable can be set to either **No** or **Yes**. If the value is set to **Yes** the switch will be highlighted.

## Attributes

By checking the **Mark this variable as required** checkbox a variable of the type **Boolean value** can be defined as a mandatory variable. This means, that a valid value has to be entered during a run.

# Numeric Value




A variable of the type **Numeric value** offers the following additional options for configuration:

## Default Value

This is where the default value of the variable is defined. It can be configured by either typing a value or using the **count up** and **count down** buttons. The field must contain a numeric value which has to be inside of the range that is configured in the **Range** option.

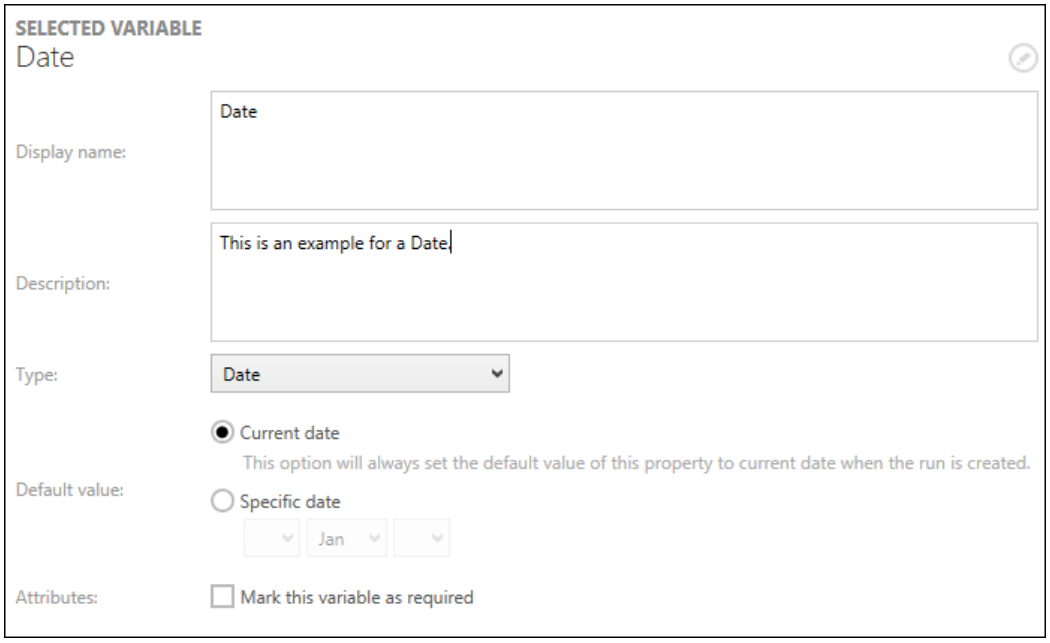
## Range

The range of the variable is configured in this two fields. It is configured from minimum to maximum. Both fields need to contain numeric values. The value of a field is configured by either using the **count up** and **count down** buttons or by typing a value inside of the field. The fields must contain numeric values. If the fields are left empty the respective boundaries will be removed. For example, if the minimum value is empty, there is no boundary for the minimum and if the maximum value is empty, there is no boundary for the maximum.

## Attributes

By checking the **Mark this variable as required** checkbox a variable of the type **Numeric value** can be defined as a mandatory variable. This means, that a valid value has to be entered during a run.

# Date




A variable of the type **Date** offers the following additional options for configuration:

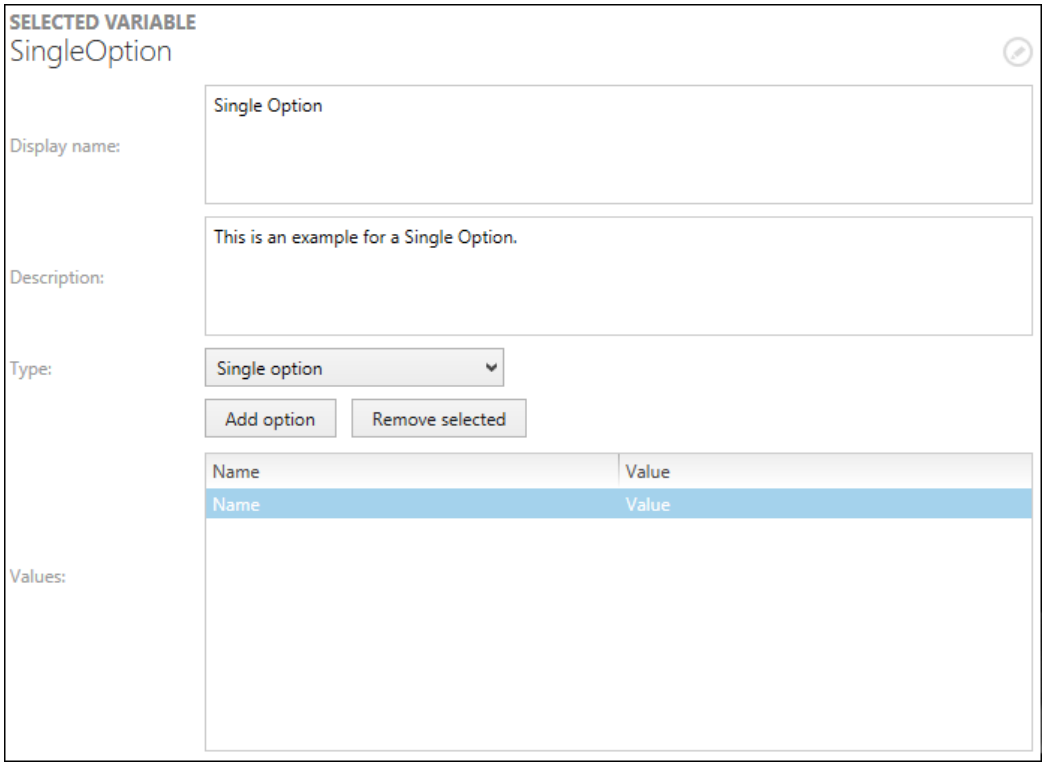
## Default Value

This is where the default value of the variable is defined. The default value for this option can be set to either **Current date**, which will display the current date as the default value of the variable, or **Specific date**, which will display the date defined in the value as the default value of the variable.

## Attributes

By checking the **Mark this variable as required** checkbox a variable of the type **Date** can be defined as a mandatory variable. This means, that a valid value has to be entered during a run.

# Single Option




A variable of the type **Single option** offers the following additional options for configuration:

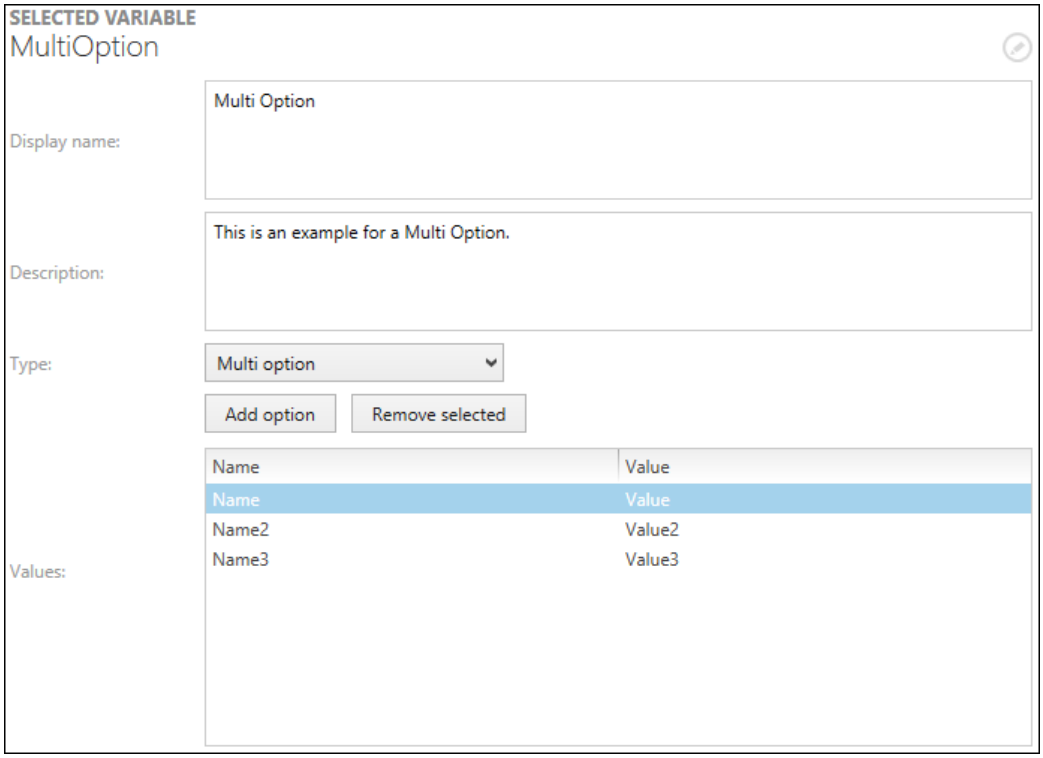
## Values

There are two buttons and a table available here. The first button **Add option** can be used to add an option to the table. The second one **Remove selected** can be used to remove a selected option from the table below.

The table itself consists of two columns: **Name** and **Value**. In the **Name** column, the name of the option is defined. In the **Value** column, the value of the option is entered.

During a run a **Single option** variable will allows users to choose one single option from a dropdown list.

# Multi Option




A variable of the type **Multi option** offers the following additional options for configuration:

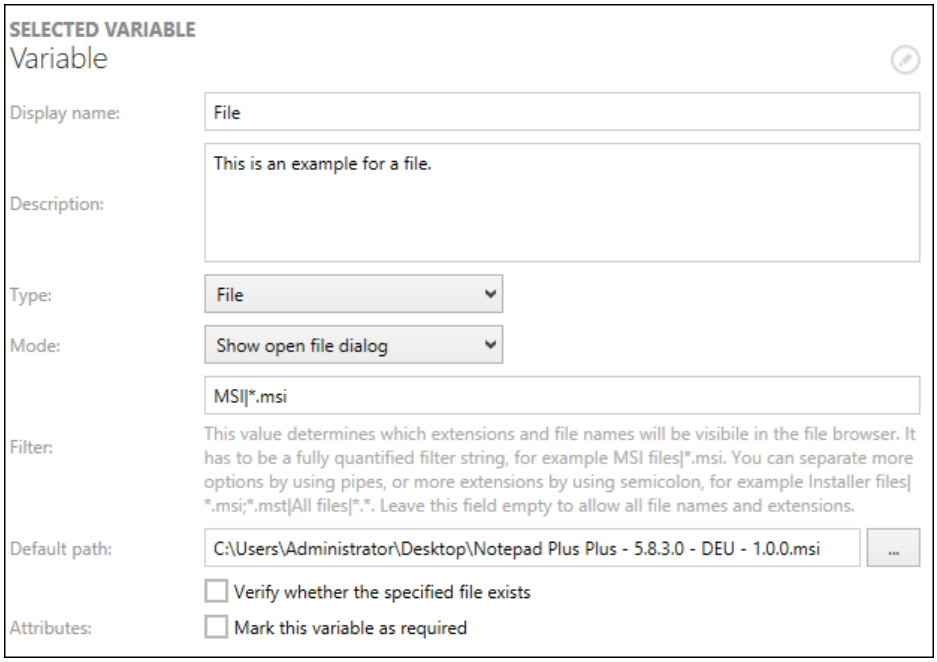
## Values

There are two buttons and a table available here. The first button **Add option** can be used to add an option to the table. The second one **Remove selected** can be used to remove a selected option from the table below.

The table itself consist of two columns: **Name** and **Value**. In the **Name** column, the name of the option is defined. In the **Value** column, the value of the option is entered.

During a run a **Multi option** variable will allow users to enable or disable different options by checking or unchecking the respective checkboxes.

# File




A variable of the type **File** offers the following additional options for configuration:

## Mode

The mode for the variable can be set here. There are two different modes available.

**Show open file dialog**: If this mode is selected the variable will open an open file dialog.
**Show save file dialog**: If this mode is selected the variable will open an save file dialog.

## Filter

This value determines which extensions and file names are visible in the file browser. It has to be a fully qualified filter string (example: `*.msi`). Two or more options can be separated by using pipes (example: `Installer files|*.msi|All files|*.*`). Two or more extensions can be separated by using semicolons (example: `*.msi;*.mst`). The field can be left empty to allow all files and extensions.

## Default Path

This is where the default value of the variable is defined. The value will contain the path to a file. It can either be entered by manually typing the path or by clicking on the Browse button [**...**]. The **Open** dialog will appear. Browse to the target file and click on the **Open** button.

If the **Verify whether the specified folder exists** option is checked, PackBench will check if the path given in the variable is valid and return an error message if it is not.

The default value can contain run variables (see chapter *PackBench variables*). Standard values are

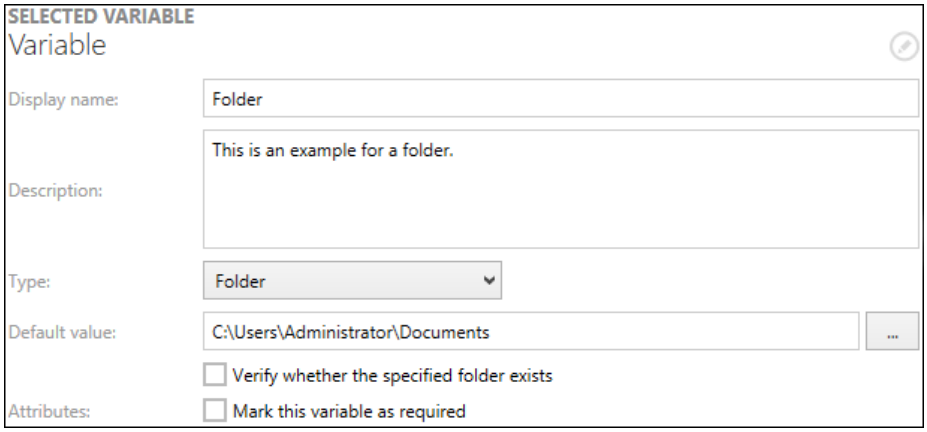
resolved as you change them. For example, to define that a variable by default has to contain full package directory and arbitrary file name, use the following default value:

```
($PkgDir)\TestFile.log
```

## Attributes

By checking the **Mark this variable as required** checkbox a variable of the type **File** can be defined as a mandatory variable. This means, that a valid value has to be entered during a run.

# Folder




A variable of the type text offers the following additional options for configuration:

## Default Value

This is where the default value of the variable is defined. The value will contain the path to a folder. It can either be entered by manually typing the path or by clicking on the Browse button [...]. The **Select Folder** dialog will open. Browse to the target folder and click on the **Select Folder** button.

If the **Verify whether the specified folder exists** option is checked, PackBench will check if the path given in the variable is valid and return an error message if it is not.

The default value can contain run variables (see chapter *PackBench variables*). Standard values are resolved as you change them. For example, to define that a variable by default has to contain full package directory and arbitrary subfolder name, use the following default value:

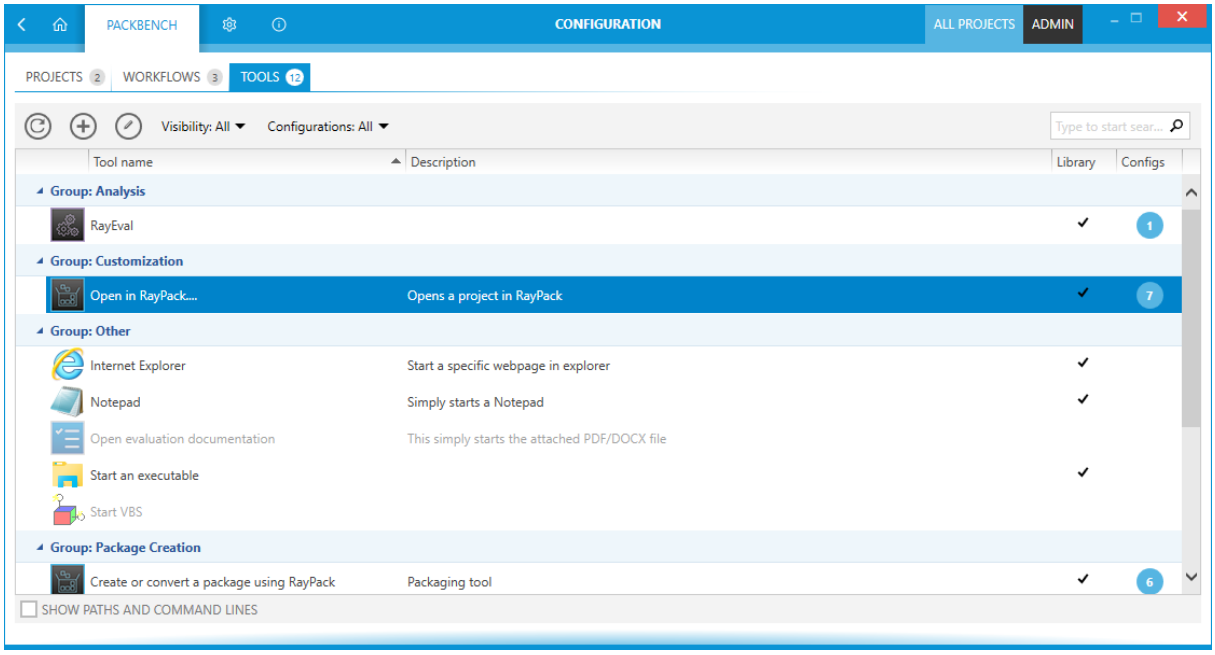
```
($PkgDir)\Subfolder
```

## Attributes

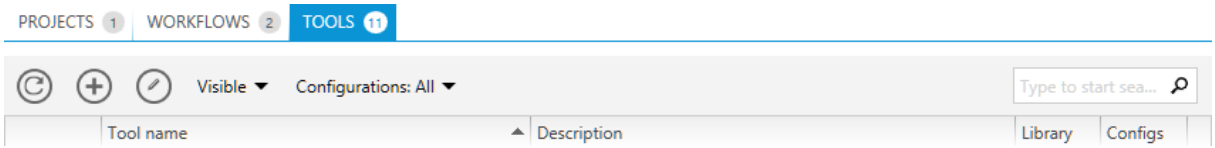
By checking the **Mark this variable as required** checkbox a variable of the type **Folder** can be defined as a mandatory variable. This means, that a valid value has to be entered during a run.

# Configuring Tools

This chapter explains how to create tools with PackBench and how to edit an already existing tool.




To configure a tool left-click on the **Configuration** tile in the **config** section of the Home Screen. The **CONFIGURATION** view is opened. Select the **TOOLS** tab inside of the **CONFIGURATION** screen.




The TOOLS tab shows an action bar, the header of the content area, and the content area where the tools are shown.

## Tools Tab Header

The header of the tab is highlighted when the tab is selected. Furthermore, it shows a count of the tools which are currently existing in PackBench. Hidden tools are still counted.

### Action Bar

There are four different abilities available in the action bar.

## Refresh this View Button

This button can be used to refresh the content area of the **TOOLS** tab. If any changes have been made to any of the tools, this button should be used to ensure that the changes are shown in the tab.

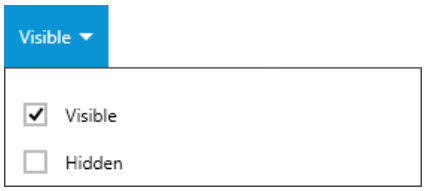
## Create New Tool Button

Click this button to create a new tool. For more information on this, refer to the *Creating Tools* section of this chapter.

## Edit Tool Button

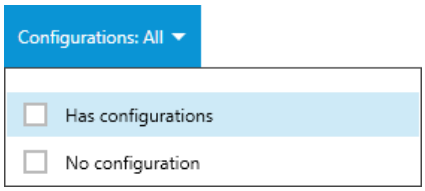
Click this button to edit currently selected tool. For more information on this, refer to the *Editing Tools* section of this chapter.

## Visibility




The **Visibility** button opens a dropdown menu which can be used to define which tools will be shown in the **TOOLS** tab. Available options in this menu are **Visible** and **Hidden**. Enable or disable an option by using the checkbox related to the option. If hidden tools are shown in the content area, they will be grayed out. The title of the button will also change to **Visible, Hidden** to clarify that all tools are shown. By default, only visible tools are shown.

## Configurations




The **Configurations** button opens a dropdown menu which can be used to define which tools will be shown in the **TOOLS** tab. Available options in this menu are **Has configurations** and **No configuration**. Enable or disable an option by using the checkbox related to the option. The

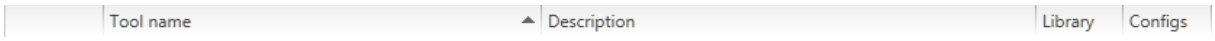
title of the button will also change to **Has configurations, No configuration** to clarify that all tools are shown. By default, all tools are shown.

## Keyword Search Field

The keyword search field can be used to filter the tools. Type the keyword into the search field. The tools will be filtered for the keyword while typing.

For example: type "sample" into the search field and only tools which contain the word "sample" in either the **Tool name**, the **Path**, the **Command line**, or the **Description** will be shown.

### Sorting and Columns




It is possible to sort the content by column. Select the column and click on it to sort the tools alphabetically. An arrow will appear next to the selected column header. To change the sort order simply click on it again. The arrow will indicate the order by either pointing upwards or downwards.

The columns can be interchanged by drag and drop. Simply drag the column header and drop it at the target position. The columns can also be put underneath each other by dropping the header of one column underneath the header of another column.

- **Icon**
  Displays the icon of the tool as configured in *the tool properties dialog*

- **Tool name**
  Displays the name of the tool as configured in *the tool properties dialog*

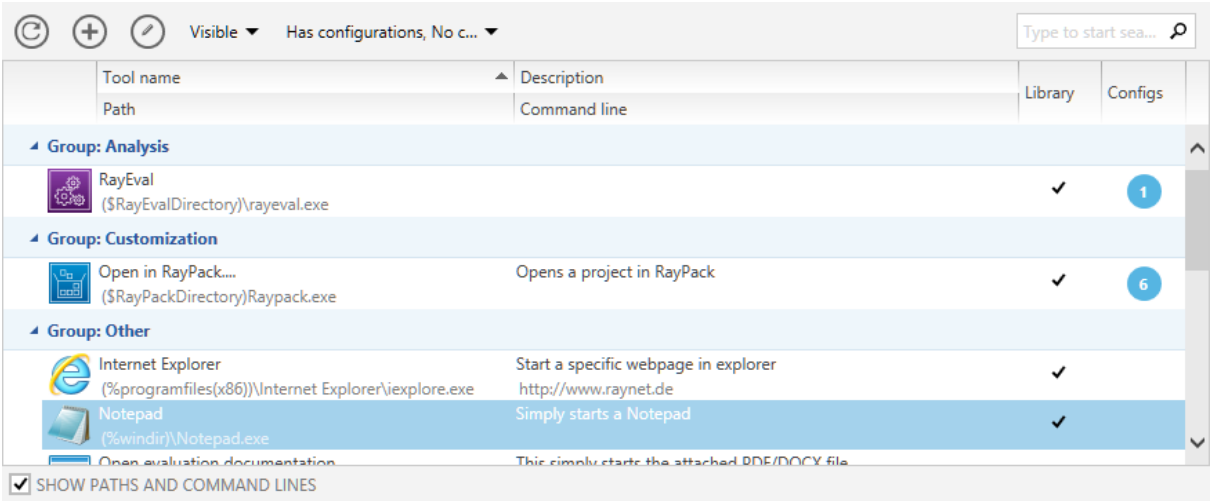
- **Description**
  Displays the description of the tool as configured in *the tool properties dialog*

- **Library**
  Displays a checkmark informing whether the tool is visible in the library as configured in *the tool properties dialog*

- **Configs**
  Displays a number informing how many configurations are defined for this tool. If there are no configurations, nothing is shown. The configurations can be edited in *Configurations* section.

### Showing Command Line Paths and Arguments

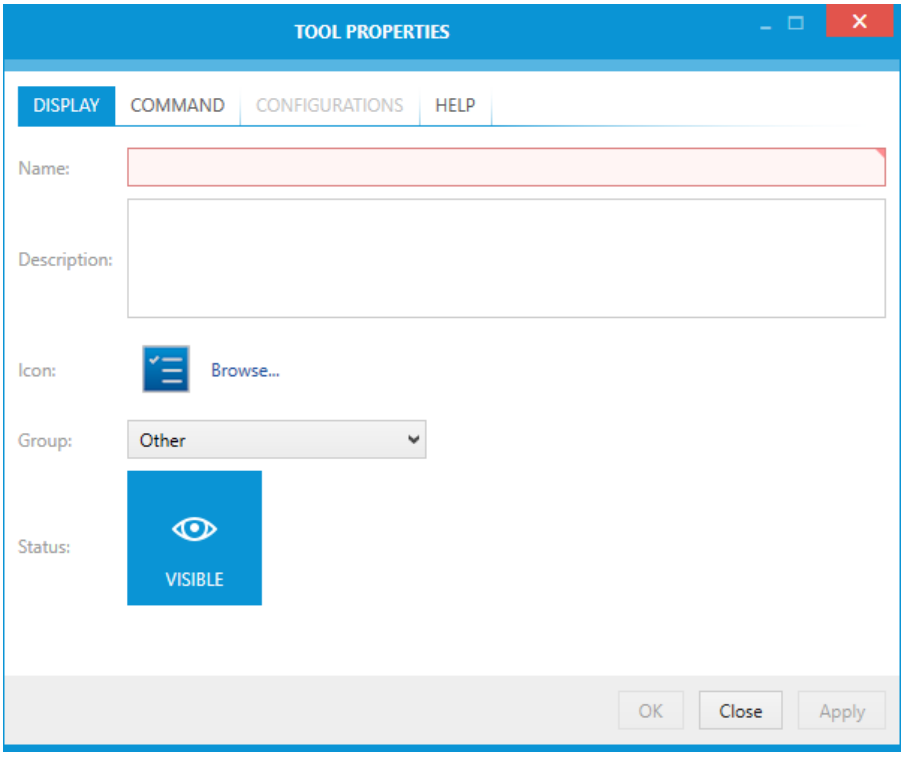
By default, paths and command line arguments are hidden in this view. To enable them, tick the checkbox under the table. New columns will be revealed automatically. Untick the checkbox to hide them again.




# Creating Tools

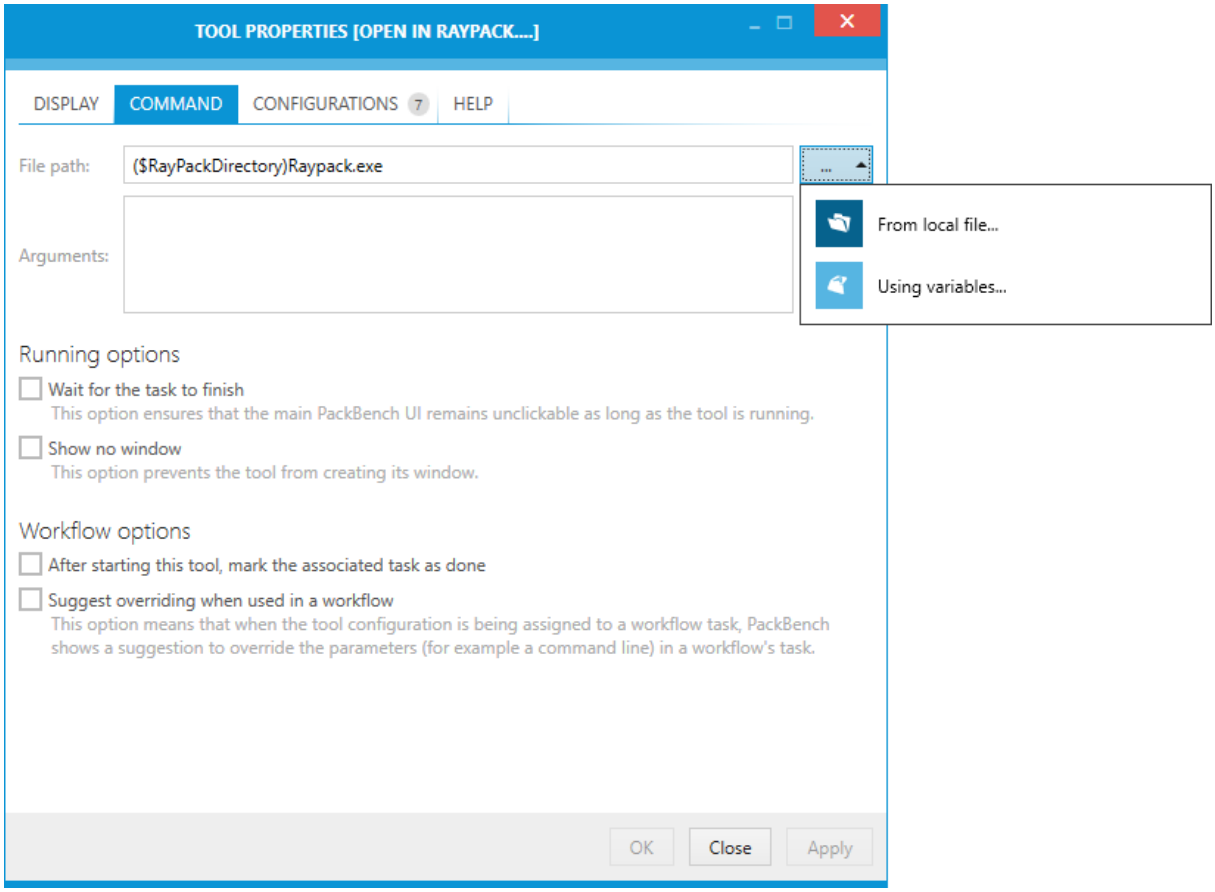
To create a new tool open the **TOOLS** tab in the **CONFIGURATION** view and then either open the context menu and select **New...** or click on the **Create new tool** button (⊕).

The **TOOL PROPERTIES** window will open.




Enter the name of the new tool into the Name field.

After this, switch to the **COMMAND** tab of the **TOOL PROPERTIES** and enter a File path.




The file path can be either typed in directly or added by clicking on the Browse button [...] and either choosing the **From local folder...** option or the **Using variables...** option.

- **From local folder...**
  Choosing this option will open the **Open** window. Browse to the location where the file can be found, select the file, and click on the **Open** button to add the file to the task.

- **Using variables....**
  Choosing this option will open the **Value Editor**. For information on how to use the **Value Editor** refer to the *Value Editor* chapter.

**Note:**
Both **Name** and **File path** are mandatory.

Click on the **OK** to save the tool and close the window or the **Apply** button to save the tool and configure it further. To learn how to further configure the tool refer to *the next chapter.*

# Editing Tools

Tools can be edited by opening the **TOOLS PROPERTIES** of the target tool. They can be opened by clicking on the **Properties...** option in the context menu.

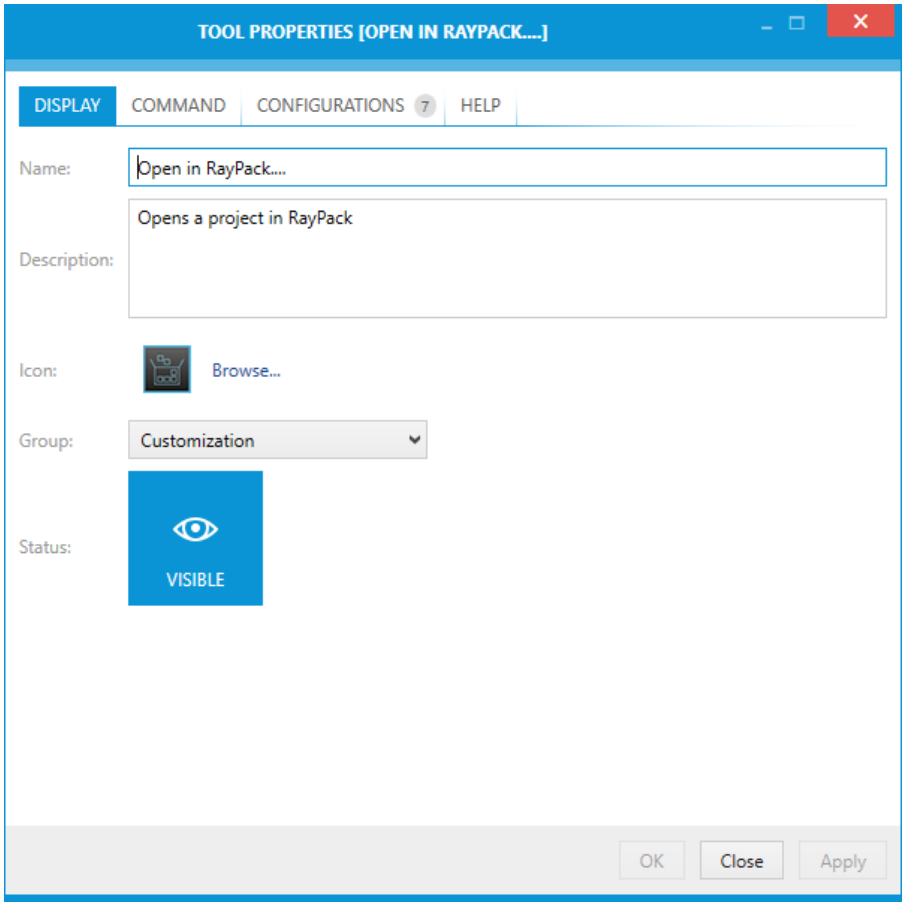
The dialog consist of four functional tab groups:

- DISPLAY
- COMMAND
- CONFIGURATIONS
- HELP

Click on the **OK** button to save the changes and close the window, click on the **Apply** button to save the changes without closing the window, or click on the **Cancel** button to dismiss the changes.

## Display

The **DISPLAY** tab contains various settings controlling the presentation and group belonging of a configured tool.



## Name

Shows the name of the tool and can be used to change the name. This field is mandatory for all tools.

**Tip:**
Name will be displayed in the tooltip in the dashboard library (once the tool has been configured to be visible there). Keep the name of the tool short and descriptive.

## Description

The description of the tool can be entered or altered here.

## Icon

This is the icon associated with the tool. The icon is shown in many places, including:

- Dashboard library (if the tool has been configured to be visible in the dashboard)
- Task view (if the tool has been associated with a task)
- Tool configurations are sharing the main tool icon.

In order to define a custom icon, press **Browse...** to browse for supported icon files.

**Tip:**
It is recommended to use `.ico` files. PackBench can also extract the icons from executables and use other graphic formats.

## Group

Tools are sorted in groups and the group of the tool is defined here. Select one of the groups from the dropdown menu. The following groups are available:

- Other
- Software management
- Application request
- Analysis
- Package creation
- Customization
- Conflict resolution
- Quality assurance
- Release management
- Validation

- Product administration

New groups can be configured in the `ToolGroup` table in the database (By default: `Servername\Databases\PackBench\Tables\dbo.ToolGroup`) using the SQL Server Management Studio.

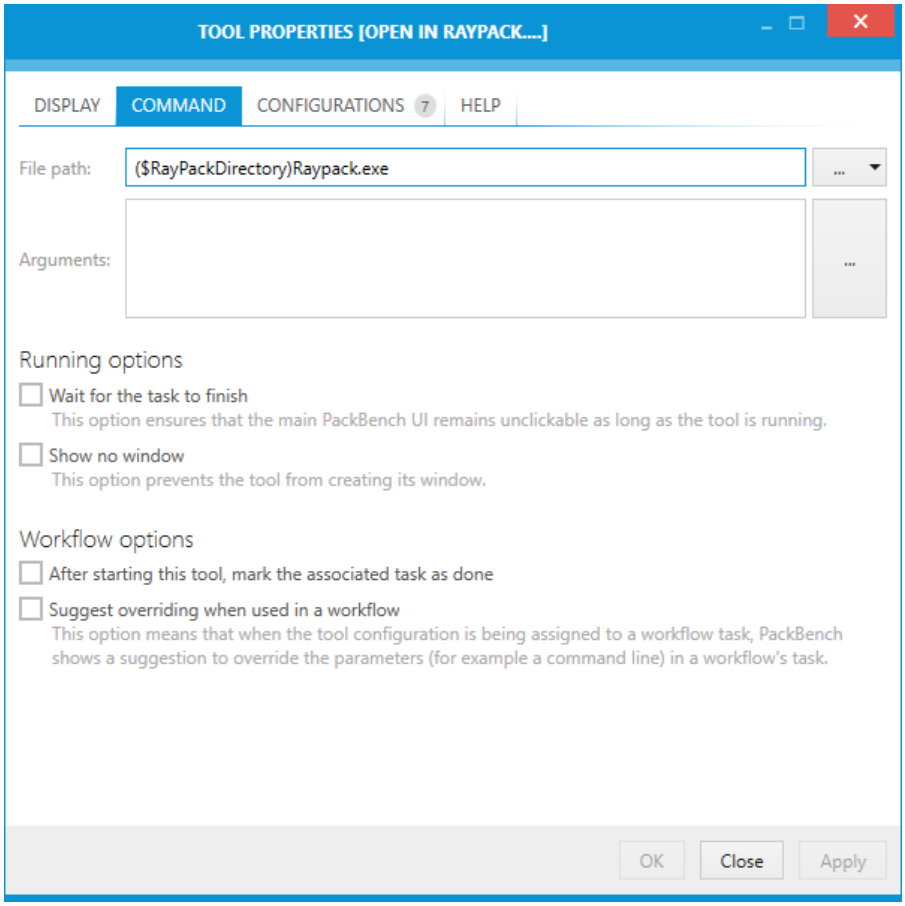
## Status

The status of the tool can be defined by clicking on the button. It can be set to either **VISIBLE** or **HIDDEN**. The current status will be shown on the button and the button will be highlighted if the current status is **VISIBLE**.

This setting affects whether the tool can be shown in the *library* or not. Hidden tools are not shown in the library, but they can be still accessed when working with *workflow editor*.

# Command

The **COMMAND** tab contains various settings specifying how the tool is executed and processed.




## File Path

The file path can be either typed in directly or added by clicking on the Browse button [...] and

either choosing the **From local folder...** option or the **Using variables...** option.

- **From local folder...**
  Choosing this option will open the **Open** window. Browse to the location where the file can be found, select the file, and click on the **Open** button to add the file to the task.

- **Using variables...**.
  Choosing this option will open the **Value Editor**. For information on how to use the **Value Editor** refer to the *Value Editor* chapter.

The **File path** is a mandatory field for all tools.

## Arguments

Arguments can be configured either directly or by clicking on the Browse button [**...**] and using the Value Editor. To learn more about arguments refer to the *Value Editor* chapter.

## Running Options

These options define how the tool is started and processed:

- **Wait for the task to finish**
  If this checkbox is checked, it is not possible to use the main PackBench UI while the tool is running. It can be used again, as soon as the tool is closed.

- **Show no window**
  If this checkbox is checked, the tool will not create a window.

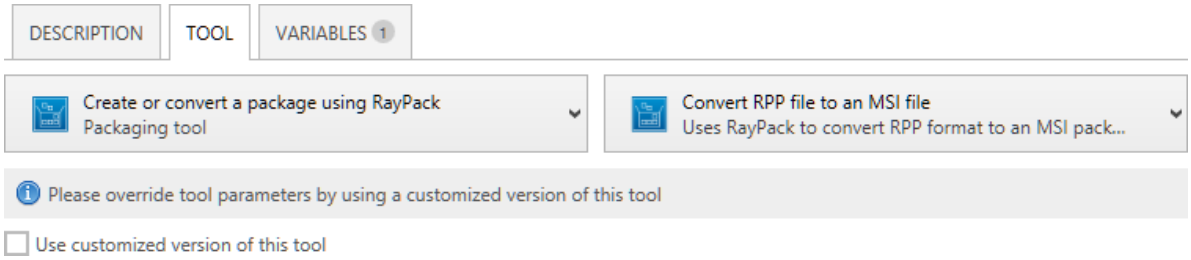
## Workflow Options

These options define the behavior of a tool when used/configured in a workflow:

- **After starting this tool, mark the associated task as done**
  If this checkbox is checked, the associated task in the workflow will automatically be marked as done if the tool is started with PackBench. If the checkbox is not checked, the task has to be marked as done manually.

- **Suggest overriding when used in a workflow**
  All tools can be overriden on a workflow basis (see *workflow configuration section for more information*). By enabling this option, when authoring a workflow a message will be displayed each time the tool is connected to a workflow without overriding its parameters. This option can be used, for example:
  - To use placeholders in the tool and signal to the workflow creator that certain strings have to be replaced with workflow-specific variables
  - To keep the tool generic even though it is expected that the workflow has to always configure it to make it work

Once this flag is open, every time the tool is accessed in the task editor a following message will be shown:




More information about consuming tools and their configurations can be found in chapter *Configuring task tools*.

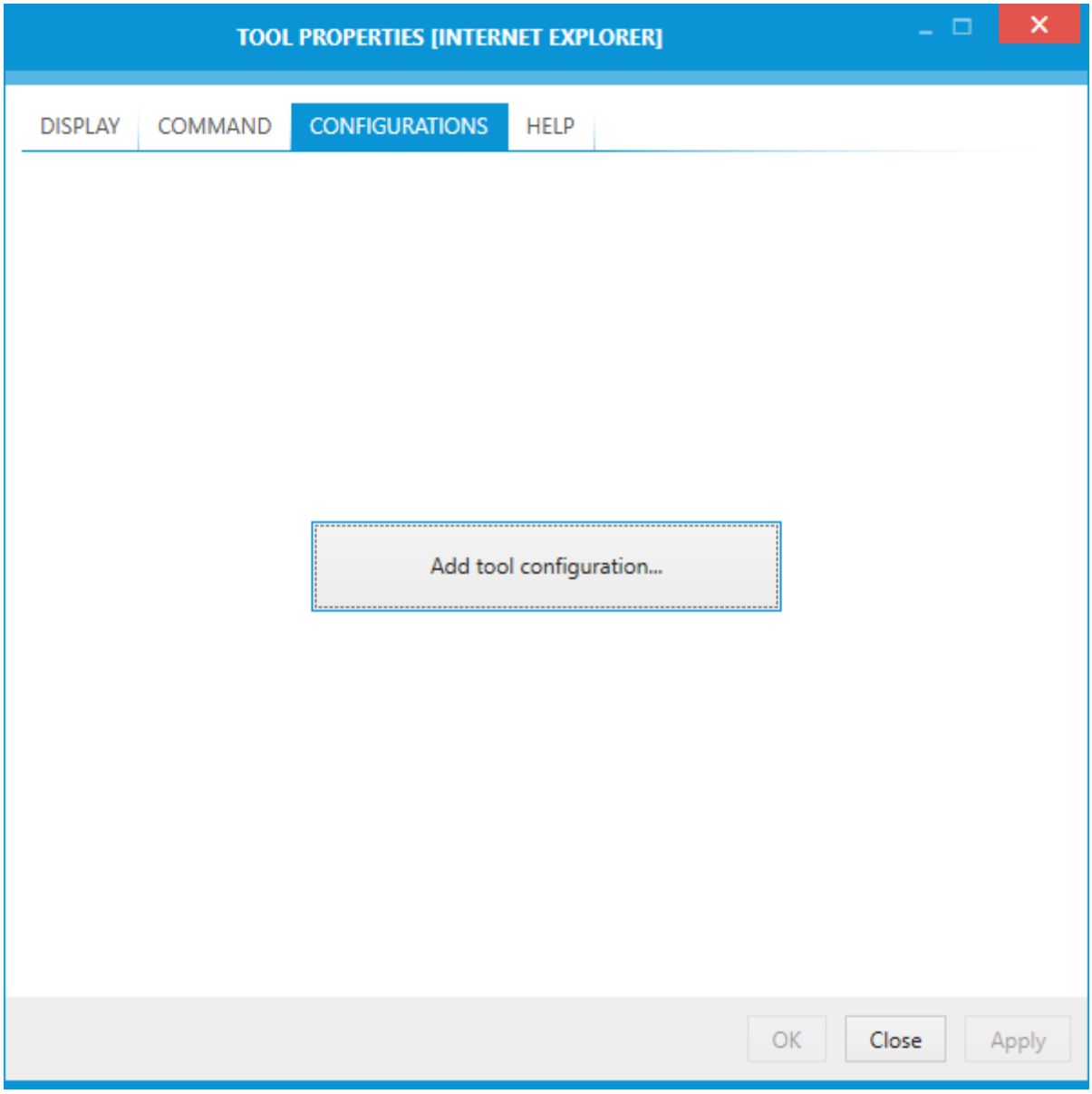
# Configurations

The **CONFIGURATIONS** tab contains an editor for tool configurations.
Each tool can define several tool configurations. A configuration can be used to redefine the entry point file, arguments, or behavior of a tool.

Configurations should be used in the following scenarios:

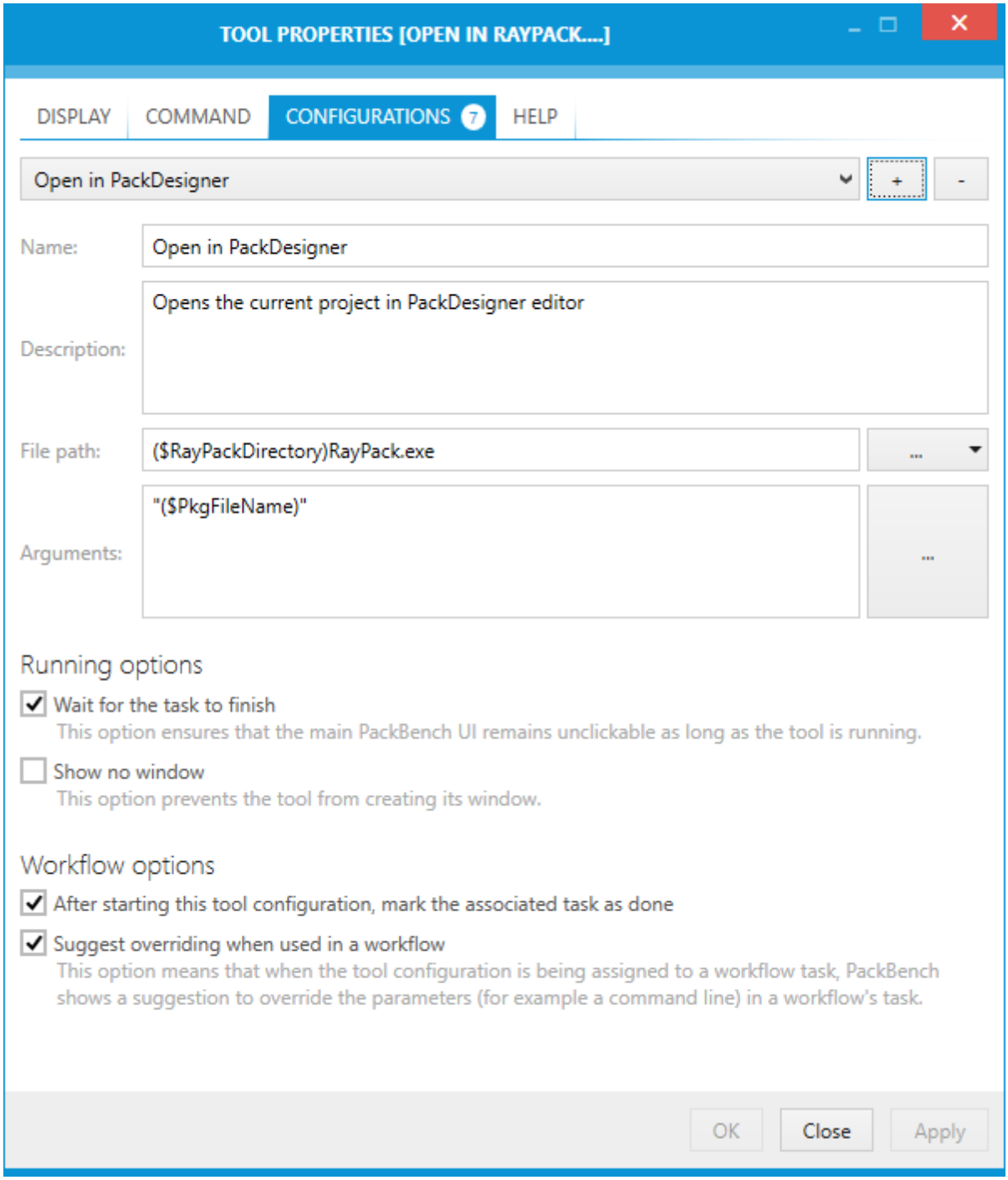
- To group similar functionality in one tool
- To group functionality by entry point executable
- To provide different configurations for slightly different tasks (for example: a configuration to convert RCP to RPP and another one to convert RPP to MSI format)

The number in the tab shows the number of configurations associated with this tool. When a new tool is created, by default, it has no configurations:




v

Once at least one configuration is available, the configurations are displayed in the following way:




## Configuration Selector

The drop-down menu contains all configurations associated with a given tool. All following input controls are always showing the details of the currently selected configuration.

- **To add a new configuration**, press the plus (**+**) button next to the drop-down menu.
- **To remove currently selected configuration**, press the minus (**-**) button next to the drop-

down menu.

### Name

Shows the name of the tool and can be used to change the name. This field is mandatory for all tools.

## Description

The description of the tool can be entered or altered here.

## File Path

The file path can be either typed in directly or added by clicking on the Browse button [**...**] and either choosing the **From local folder...** option or the **Using variables...** option.

- **From local folder...**
  Choosing this option will open the **Open** window. Browse to the location where the file can be found, select the file, and click on the **Open** button to add the file to the task.

- **Using variables...**.
  Choosing this option will open the **Value Editor**. For information on how to use the **Value Editor** refer to the *Value Editor* chapter.

The **File path** is a mandatory field for all tools.

## Arguments

Arguments can be configured either directly or by clicking on the Browse button [**...**] and using the Value Editor. To learn more about arguments refer to the *Value Editor* chapter.

## Running Options

These options define how the tool is started and processed:

- **Wait for the task to finish**
  If this checkbox is checked, it is not possible to use the main PackBench UI while the tool is running. It can be used again, as soon as the tool is closed.

- **Show no window**
  If this checkbox is checked, the tool will not create a window.

## Workflow Options

These options define the behavior of a tool when used/configured in a workflow:

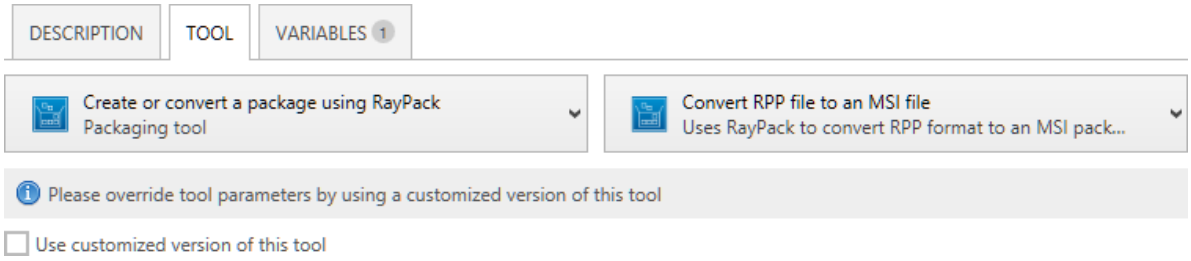
- **After starting this tool, mark the associated task as done**
  If this checkbox is checked, the associated task in the workflow will automatically be marked as done if the tool is started with PackBench. If the checkbox is not checked, the task has to

be marked as done manually.

- **Suggest overriding when used in a workflow**
  All tools can be overriden on a workflow basis (see *workflow configuration section for more information*). By enabling this option, when authoring a workflow a message will be displayed each time the tool is connected to a workflow without overriding its parameters. These option can be used, for example, to:

  - Use placeholders in the tool and signal to the workflow creator that certain strings have to be replaced with workflow-specific variables
  - Keep the tool generic although it is expected that the workflow author has to always adjust it to make it work

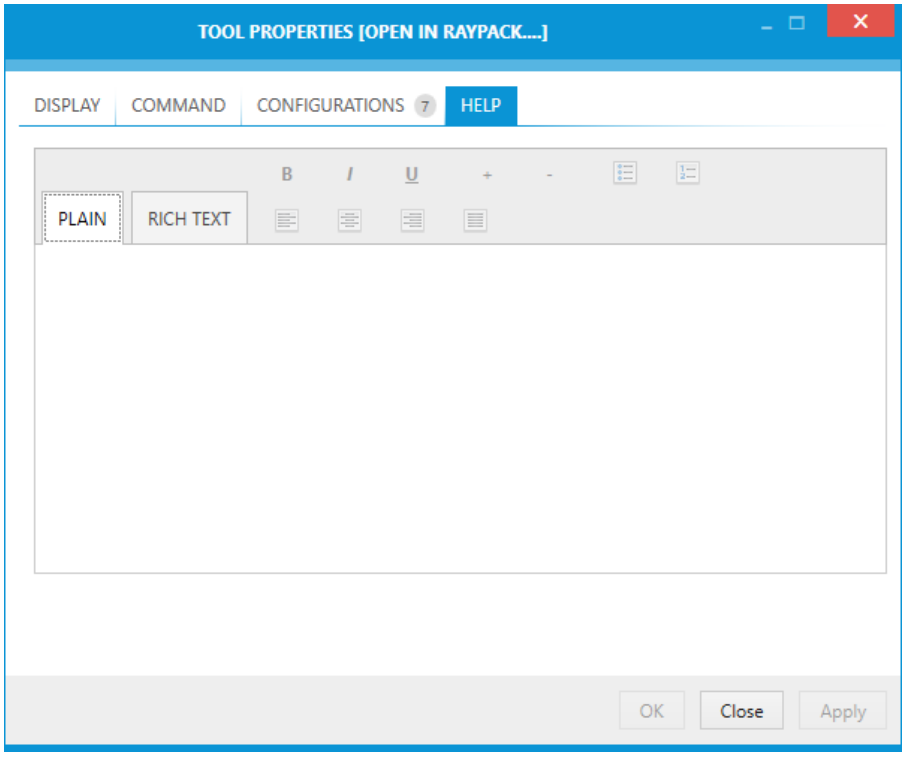
Once this flag is open, every time the tool is accessed in the task editor a following message will be shown:




More information about consuming tools and their configurations can be found in chapter *Configuring task tools*.

# Help

The **HELP** tab defines the help content for a tool.




The text can be entered into two different tabs. The **PLAIN** and the **RICH TEXT** tab. The **PLAIN** tab can be used to enter a text without formatting it. The **RICH TEXT** tab can be used to enter text and format it.

Help content is only displayed in this screen. It can be used to document certain command line settings, attach more information, hyperlinks and/or extended description.

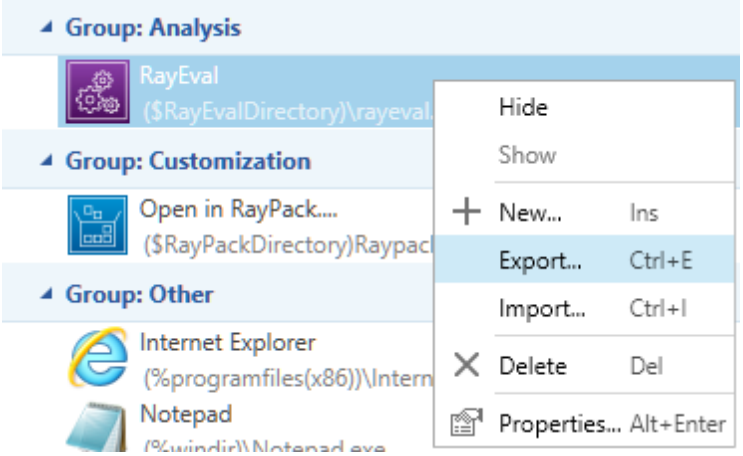
# Deleting Tools

## In Order to Delete a Tool...

1. Select the tool in the *tools view.*

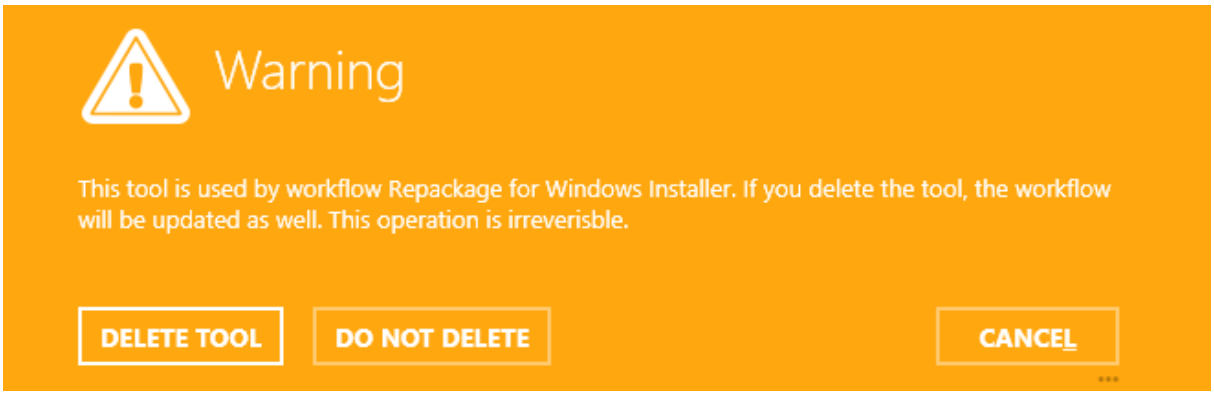
2. Right click to open a context menu.

3. Click on **Delete**




4. Confirm deletion of the tool.

5. If the tool is already used by any workflow, the following warning will be shown:




6. Removing a tool may affect existing runs and workflows using the tool.

**Tip:**
The keyboard shortcut **Delete** can be used to delete a tool without going to context menu.

# Exporting Tools

Tools can be exchanged between different instances of the product by using import/export functionality. PackBench supports exporting and importing from an XML-based format.

When exporting a tool, the following data are saved in XML file:

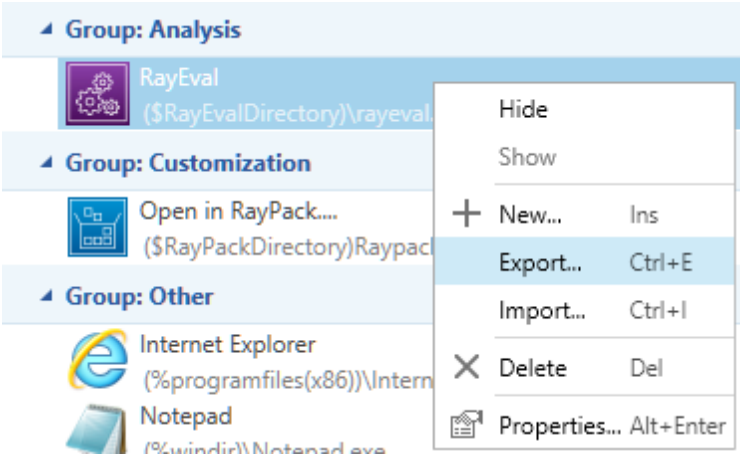
- Tool details
- Tool configurations
- Name of the group to which the tool belongs to

## In Order to Export a Tool to XML Format...

1. Select the tool in the *tools view*.

2. Right click to open a context menu.

3. Click on **Export...**




4. Navigate to the place where to save the exported tool and provide a file name for it.

5. The tool will be exported after pressing **Save** button.

**Tip:**
The keyboard shortcut **CTRL+E** can be used to export a tool without going to context menu.

# Importing Tools

Tools can be imported from XML files that were *previously exported* from PackBench.

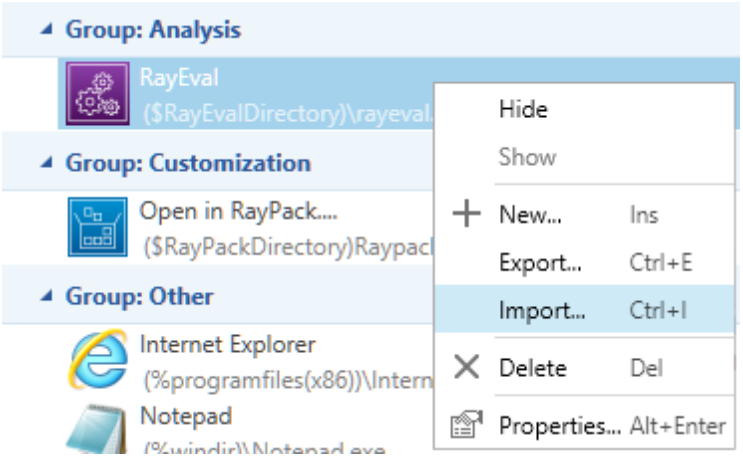
When importing a tool, the following data are read from XML file:

- Tool details
- Tool configurations
- Name of the group to which the tool belongs to
  - If no group with the given name exists in the current instance, a new group will be created automatically.

# In Order to Import a Tool to XML Format...

1. Right click anywhere in the *tools view* to open a context menu.

2. Click on **Import...**




3. Navigate to the place where the imported `.xml` file is located.

4. The tool will be imported after pressing the **Open** button.

**Tip:**
The keyboard shortcut **`CTRL+I`** can be used to import a tool without going to context menu.

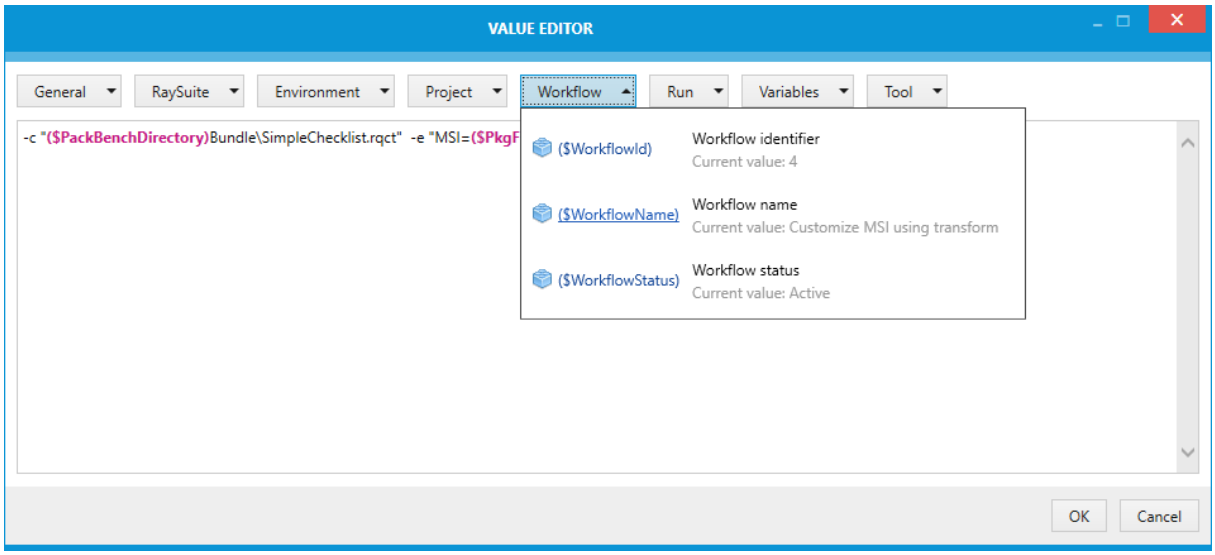
# Common Dialogs

Some dialogs of the PackBench interface cover generic functionality, such as editing formatted syntax using custom variables.

The following common dialogs for standard procedures are covered within this section:

- *Variable Editor*

## Variable Editor




The **VALUE EDITOR** can be used to create a custom value for the field from which the editor has been opened. It consist of several drop-down menus:

- *General*
- *RaySuite*
- *Environment*
- *Project*
- *Workflow*
- *Run*
- *Variables*
- *Tool*

Each of them contains predefined arguments which can be used when creating a new value. The number of menus varies depending on the location from where the editor has been opened.

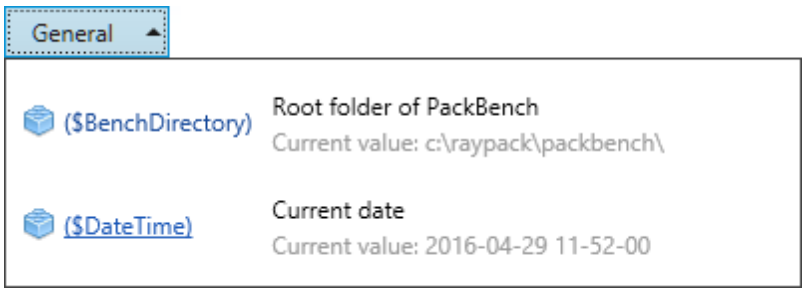
### Syntax Highlighting

The syntax is automatically highlighted. Recognized properties are displayed using a purple foreground, and properties that are not recognized but still using a valid syntax are displayed using the blue color. All other characters are displayed using standard black font.

### Accepting and Dismissing Changes

To enter the value into the field from which the editor has been opened, click on the **OK** button. To dismiss the value click on the **Cancel** button.

## General

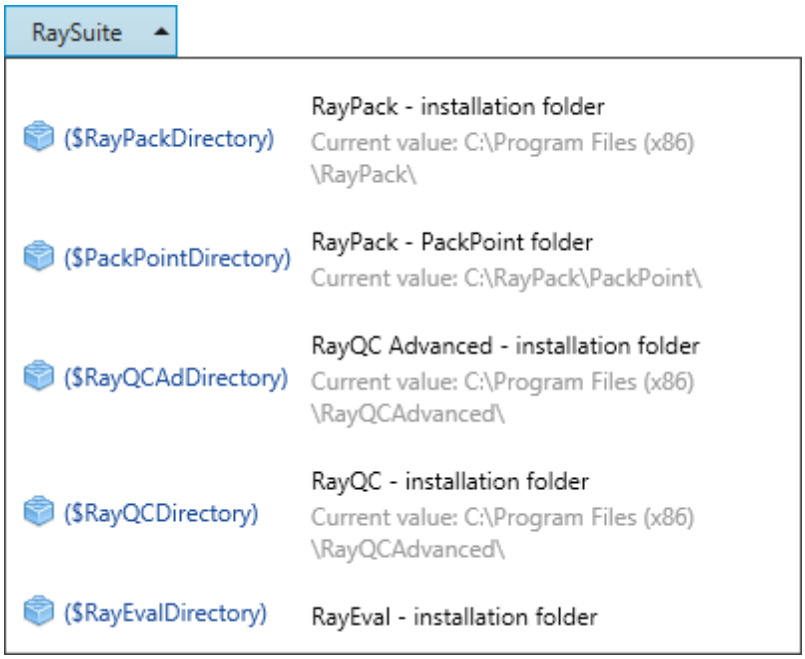



This dropdown menu contains general variables.

`(§BenchDirectory)`
This argument contains the path to the root folder of PackBench and can be used to navigate to the root folder.

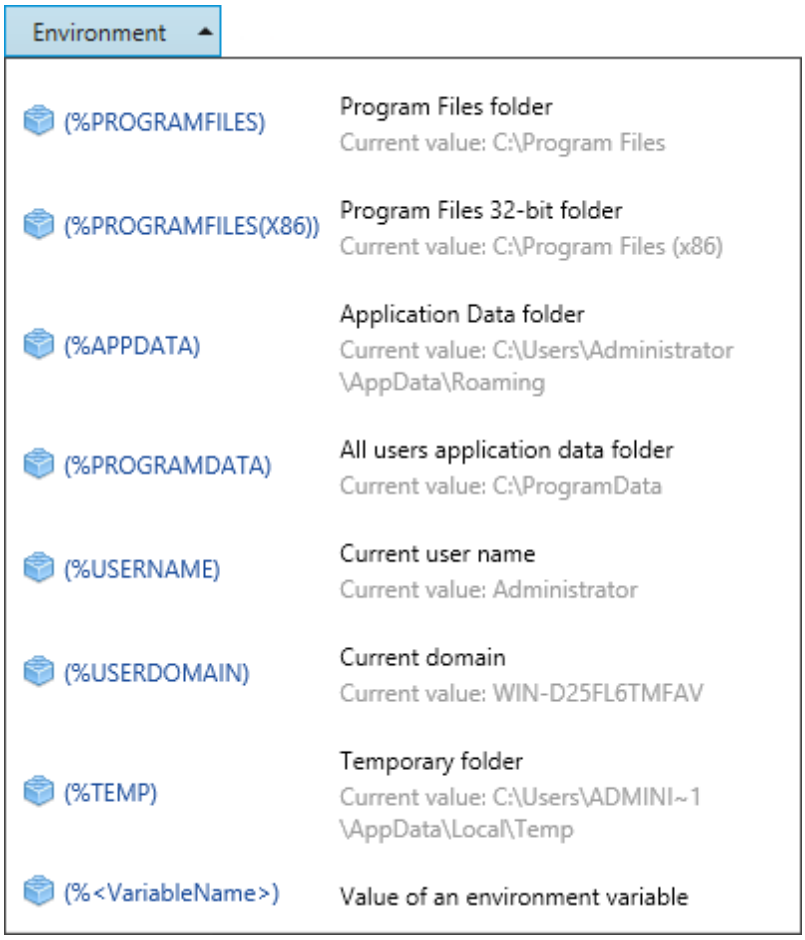
`($DateTime)`
This argument contains the current date and time, in format `YYYY-MM-DD MM-HH-SS`.

# RaySuite




This dropdown menu contains arguments which contain the installation folders of the RaySuite products which are currently installed on the local machine. The information is taken from the registry entries of the installed RaySuite product.
The list of supported variables is available in chapter *PackBench variables*.

# Environment




- `(%PROGRAMFILES)`
  This argument contains the path to the Program Files folder. By default this is `C:\Program Files`.

- `(%PROGRAMFILES(X86))`
  This argument contains the path to the 32-bit Program Files folder. By default this is `C:\Program Files (x86)`.

- `(%APPDATA)`
  This argument contains the path to the application data folder of the current user.

- `(%PROGRAMDATA)`
  This argument contains the path to the application data folder which is shared by all users of the device. By default this is `C:\ProgramData`.

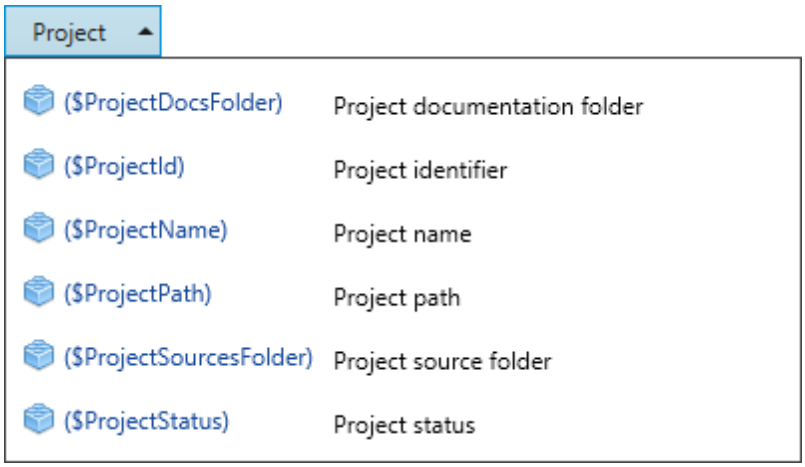
- `(%USERNAME)`
  This argument contains the user name of the current user.

- `(%USERDOMAIN)`
  This argument contains of the currently active domain.

- `(%TEMP`
  This argument contains the temporary folder of the currently active user.

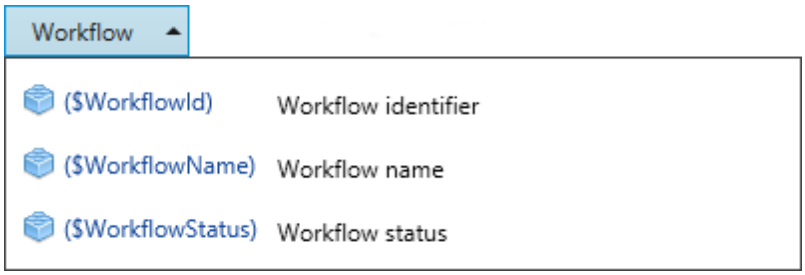
- `(%<VariableName>)`
  This argument contains the value of an environment variable by a given name, for example `(%PATH)`.

# Project



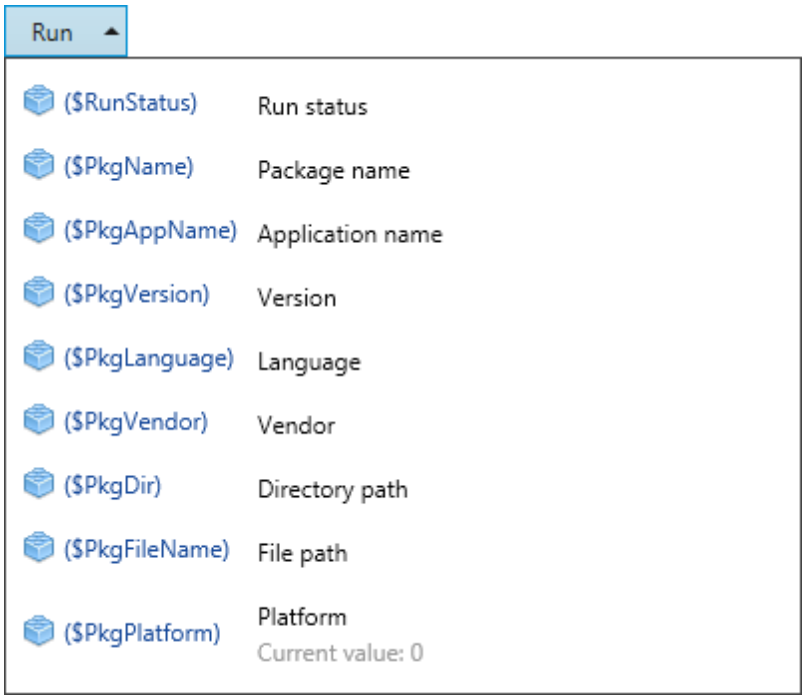

This dropdown menu contains variables describing the state of the current project.
The list of supported variables is available in chapter *PackBench variables*.

# Workflow



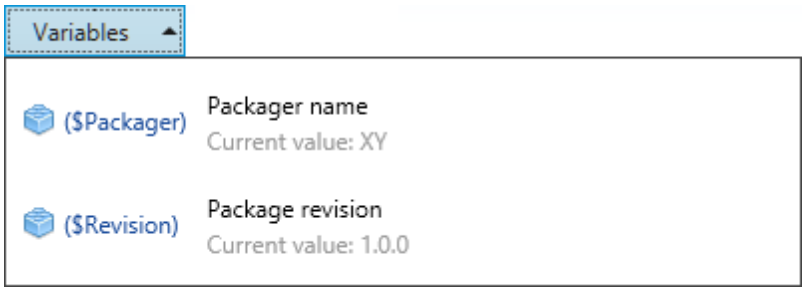

This dropdown menu contains variables describing the state of the current workflow. This dropdown is only available when working with command line arguments of a tool.
The list of supported variables is available in chapter *PackBench variables*.

# Run



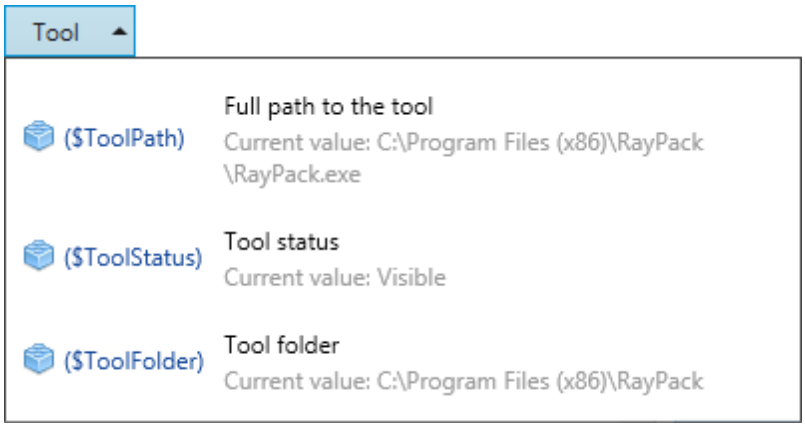

This dropdown menu contains variables describing the state of the current run. This dropdown is not available when editing project values.
The list of supported variables is available in chapter *PackBench variables*.

# Variables




This dropdown menu contains variables describing the state of the current run. This dropdown is only available when working with *overridden* command line arguments of a tool.
The exact content of the dropdown depends on workflow configuration.

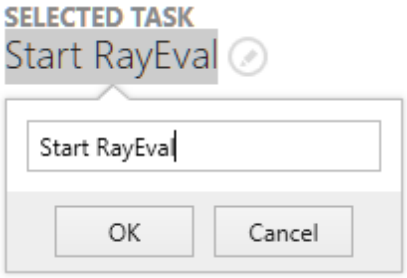
# Tool




This dropdown menu contains variables describing the details of the current tool. This dropdown is only available when working with command line arguments of a tool.
The list of supported variables is available in chapter *PackBench variables*.

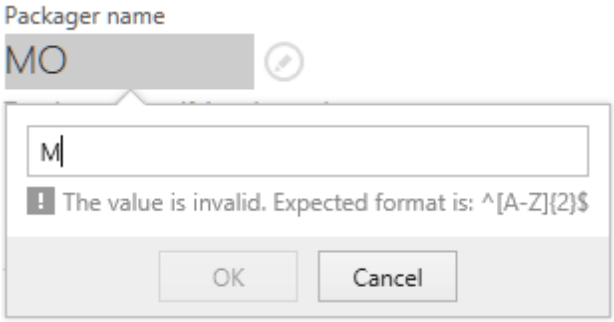
# Direct Value Editor

The Direct Value Editor is available wherever an edit icon is displayed next to a property or setting value. This handy data manipulation method allows property management directly within the original context, which is quicker and clearer than working with traditional interface types such as modal dialogs or edit forms.




If the currently edited value is restrained by rules regarding its required formatting or domain, the Direct Value Editor supports users to comply with these rules by automatic value correction, or giving textual and visual feedback – exactly where the users attention is focused and without delay.

If the entered value is invalid, a message is shown:




The **OK** button will be disabled as long as the value does not match the validation criteria.

## How to Use the Direct Value Editor Interface

- Clicking on either the property value or its **edit icon** on the right, allows to directly edit the current property value.
- To **save** the changes, hitting the enter key on the keyboard or clicking on the **OK** button below the input field is due.
- **Discarding** changes is done by hitting the escape key on the keyboard or clicking on the **Cancel** button below the input field.
- Moving the focus between the dialog controls is achieved by hitting tab on the keyboard.

# Command Line Tools

The set of command line tools for PackBench consists of a direct command line access to a dedicated executable. The following sections provide an overview of available options, required settings, and typical usage examples.

**Note:**
The command line interface is case insensitive. Therefore, it does not matter if the switch newRun is noted as "NewRun", "newrun", or "NEWRUN". Please use the diction that is recommended within your enterprises system integration style guidelines.

### Error Codes

Whenever a command is sent to the PackBench Command Line Interface Tools, they respond with a specific error code:

- If **0** is returned, the command has been executed **successfully**.
- Any other return code implies that issues have occurred during the execution.
  Details regarding the reasons for errors have to be retrieved from the log files.

# Silent Command Line Switches

The silent command line tool `PbCmd.exe` is installed automatically along with the core application and placed directly within the applications program directory, which is typically something like `C:\Program Files (x86)\RayPack\PackBench\`.

- NewRun
  Creates a new run from a given file or folder

**Be aware:**
All PackBench command line switches and parameters are case insensitive. This means that parameter `NewRun` is interpreted exactly the same as `newrun`, and a switch `-SET` is equal to `-set`

## NewRun

### Description

This switch instructs PackBench to create a new run from a local file or folder.

# Parameters

| Parameter | Required | Description |
|---|---|---|
| `-PkgName <name>` | YES | Package name |
| `-PkgVendor <vendor>` | YES | Application vendor |
| `-PkgAppName <name>` | YES | Application name |
| `-Description <description>` | YES | Description of the package |
| `-PkgDir <path>` | YES | Path to package directory |
| `-Bitness <platform>` | NO | Package platform. Neutral by default. The following values are supported: `x64, x84, Neutral` |
| `-PkgVersion <version>` | YES | Application version |
| `-PkgLanguage <language>` | YES | Application language |
| `-Project <name>` | YES | Name of the project |
| `-Workflow <name>` | YES | Name of the workflow |
| `-PkgFileName <path>` | NO | Path to sources. This can be either a full path to a source folder, or a full path to a source file. If omitted, the sources will not be assigned to the run. |
| `-SourcesEntryPoint <path>` | NO | Sources entry point. This parameter can be used only if the path to the sources points to a directory. This has to be a full path to a file present in the source folder. |

| Parameter | Required | Description |
|---|---|---|
| -CopySources | NO | Determines whether the sources should be copied to run folder. This parameter can be used only if the path to the sources has been specified. |
| -Set <name>=<value> | NO | Sets a custom variable to a given value. This parameter can be specified many times. |
| -RayFlowTaskId | NO | |
| -RayFlowUser | NO* | The account name used to make a connection with RayFlow.<br>This parameter is required if -RayFlowTaskId is specified. |
| -RayFlowPassword | NO* | The matching password (open-text) for an account used to make a connection with RayFlow.<br>This parameter is required if -RayFlowTaskId is specified. |
| -RayFlowFileName | NO | Optional name of the file uploaded to RayFlow to enable source copying. |

# Example

```
PbCmd.exe NewRun
-PkgName "0001_Package_1.0.0_ENU"
-PkgVendor Raynet
-PkgAppName MyPackage
-Description "This is a sample package"
-PkgDir \\raynet.local\sources\package_0001
-Bitness x64
-PkgVersion 1.0.0
-PkgLanguage English
-Project "Default project"
-Workflow "Repackage for Windows Installer"
-PkgFileName \\raynet.local\sources\package_0001\setup.exe
-CopySources
-Set CustomRevision=1.2.0
-Set PackagerName=MO
```

# Advanced Topics

## Customizing Reporting Templates

### Basics

Report files created by PackBench are customizable. Depending on the configuration, there is either a physical or a binary content stored in the database which represents the template file for each report template *configured per project*. The chapter *Reports* describes how to edit existing templates and how to save and update them. This chapter contains information about how to actually customize or even create documents from scratch, so that they can be used as report templates.

### Prerequisites

The templates are stored as Microsoft Word® `.docx` format. In order to edit or create them, Microsoft Word® or a third-party software capable of opening and saving `.docx` files is required.

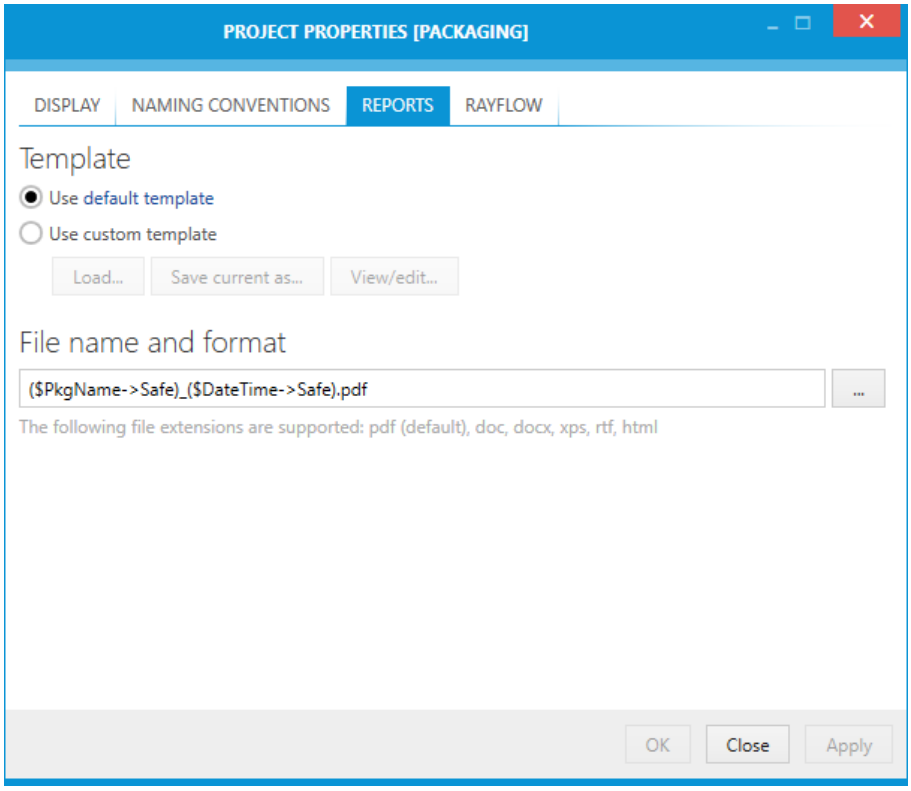
**Tip:**
It is recommended to start by making a copy of a default template and not to start from scratch.

### Default Template

The default template is located in the following location:

```
<INSTALLDIR>\Resources\Reports\DefaultTemplate.docx
```

Unless there are no other settings for a project (see the picture below), this file is used as a report template.




For example, a simple table containing base run properties can be defined using the following placeholders:

# Basic details

Executed in (#Progress)%

| | |
|---|---|
| Package name: | (#PkgName) |
| Package folder: | (#PkgDir) |
| Created: | (#PkgCreated) |
| Application name: | (#AppName) |
| Application vendor: | (#PkgVendor) |
| Application version: | (#PkgVersion) |
| Application language: | (#PkgLanguage) |

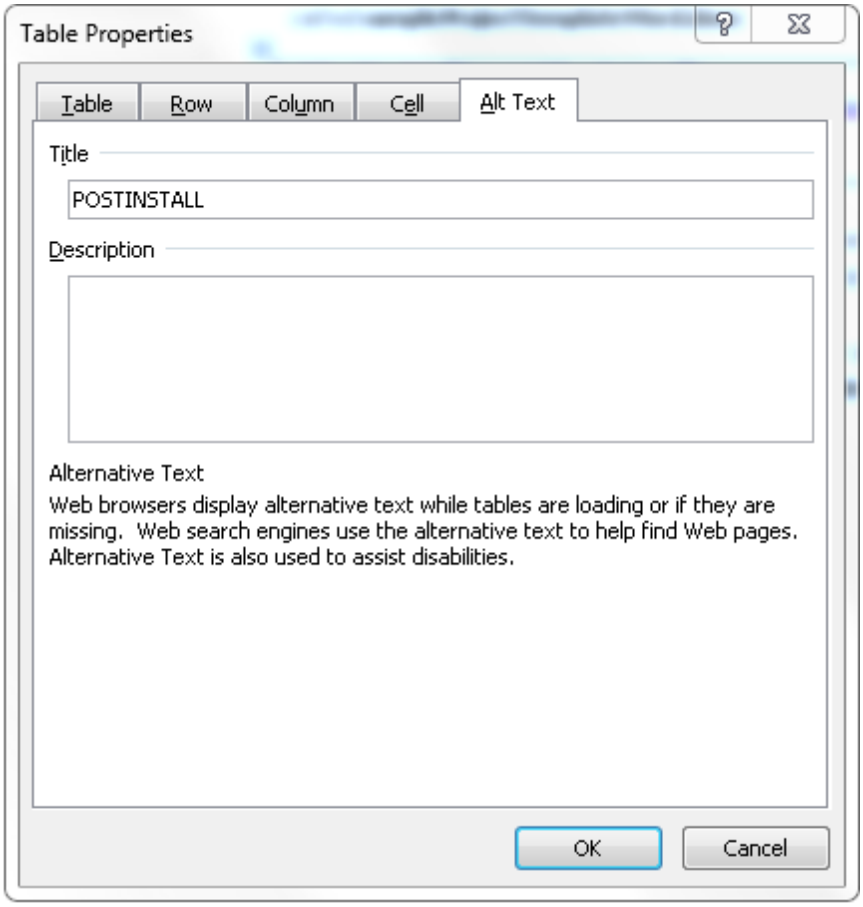
# Using Table Templates

PackBench scans the content of the template and looks for the presence of tables which have one of the following alternative texts:

- `(#Table.Tasks)`
- `(#Table.Variables)`

They both have a special meaning. Instead of printing the table as it is, the content of the table is repeated for each element - task and variable respectively - available within the current run.

In order to see or change the alternative text, select a table and from its context menu options select **Properties**. The alternative text can be changed by adjusting the **Title** field in **Alt Text** tab.




## Tasks Table

A table which contains alternative text `(#Table.Tasks)` is considered to be a tasks table. Its rows are repeated for each task present in the run. If there are no tasks, the table is not shown.

Within the table, the following placeholders can be used:

- `(#Subtasks)`
  When used inside the tasks table, it inserts a list of subtasks in the place where the placeholder occurs. It is r
- `(#Name)`
  When used inside the tasks table, it represents the name of the task.
- `(#Note)`
  When used inside the tasks table, it represents the description of the task.
- `(#HelpText)`
  When used inside the tasks table, it represents an additional help for the task.
- `(#ToolName)`
  When used inside the tasks table, it represents the name of the tool assigned to the task.
- `(#Status)`
  When used inside the tasks table, it represents a human-readable status of the task. For a colored version, us
- `(#ColoredStatus)`
  When used inside the tasks table, it represents a human-readable status of the task. The text has a foregroun
- `(#Comment)`
  When used inside the tasks table, it represents comments (if present) added to the task.

## Custom Variables Table

A table which contains alternative text `(#Table.Variables)` is considered to be a custom variable table. Its rows are repeated for each custom variable present in the run. If there are no variables, the table is not shown.
Within the table, special placeholders can be used:

- `(#Name)`
  When used inside a custom variable table, it represents the name of a custom variable.
- `(#Note)`
  When used inside a custom variable table, it represents the displayed name of a custom variable.
- `(#Description)`
  When used inside a custom variable table, it represents the description of a custom variable.
- `(#Value)`
  When used inside a custom variable table, it represent the value of a custom variable in a human-readable f

# Using Image Placeholders

PackBench scans the content of the template and looks for the presence of images which have one of the following alternative texts:

- `(#Icon)`
- `(#ProgressChart)`
- `(#Timeline)`

They all have a special meaning. Instead of printing the image as it is, the image is replaced with an actual content.

The following alternative texts can be used to indicate that the picture has to be changed dynamically:

- `(#Icon)`
  Represents the icon associated with the workflow used to create the run.

- `(#ProgressChart)`
  Represents a horizontal progress bar representing the current run progress.

- `(#Timeline)`
  Represents a chart showing the progress over time.

# Using String Placeholders

Every standard and custom property can be inserted to any place in a report, by accessing it using the following syntax:

```
(#VariableName)
```

For example, in order to insert the name of the package, write the following text in a `.docx` template:

```
(#PkgName)
```

**Be aware:**
Due to technical reasons. use "#" (hash) as a prefix inside the project template variables. This is different from how they would be used when working with tools, which use "$" (dollar sign) as a prefix character instead.

The following variables are available. For more information, refer to the chapter *PackBench variables*. Some values listed below are only allowed when used within a report template:

- `(#Note)`
  Represents the description of the run package.

- `(#Status)`
  Represents a human-readable status of the run package.

- `(#AppName)`
  Represents the application name of the run package.

- `(#FileName)`

Represents the full path to the entry file name of the run package.

- `(#PkgLanguage)`
  Represents the language of the run package.

- `(#PkgDir)`
  Represents the full path to the package directory.

- `(#PkgName)`
  Represents the name of the run package.

- `(#RayFlowGuid)`
  Represents the identifier of a RayFlow task assigned to the run..

- `(#PkgVendor)`
  Represents the name of the vendor of the package run.

- `(#PkgVersion)`
  Represents the version of the package run.

- `(#PkgCreated)`
  Represents the date of creation of the run.

- `(#Progress)`
  Represents an integer value in range from 0 to 100, informing about the current progress.

- `(#RunId)`
  Represents the numeric identifier of the run.

- `(#RunStatus)`
  Represents a human-readable status of the run.

- `(#Help)`
  Inserts an integer value in range from 0 to 100, informing about the current progress.

- `(#PkgBitness)`
  Represents the human-readable platform of the run package.

- `(#Variables)`
  Represents a list of custom variables (each variable in new line). It is not recommended to use this variable, a much more flexible and better looking option is to use the custom variables table.

- `(#Name)`
  Represents the name of the workflow used to create the run package.

- `(#WorkflowVersion)`
  Represents a full version of the workflow used to created the run.

# Custom Run Status

In order to create a custom run status, a direct access to SQL database is required.

Statuses are stored in `RunStatus` table.

Results | Messages

| | RunStatusId | Name |
|---|---|---|
| 1 | -200 | Complete |
| 2 | -100 | Closed |
| 3 | 100 | Open |
| 4 | 200 | Analyzing |
| 5 | 300 | Packaging |
| 6 | 400 | Testing |
| 7 | 500 | Deconflicting |
| 8 | 600 | On hold |
| 9 | 700 | Validating |

The default list contains several predefined statuses. The value in column `RunStatusId` has to be unique. There is a special behavior for negative values. Statuses having negative identifiers are by default considered as finished, which affects which items are by default visible in the Runs view (by default, workflows considered as not closed are shown).

# Custom Tool Groups

In order to create custom groups, a direct access to a SQL database is required.

Group names are stored in `ToolGroup` table.

Results | Messages

| | ToolGroupId | Name | Sequence | Notes | Status |
|---|---|---|---|---|---|
| 1 | 1 | Other | 11 | Other tools | 0 |
| 2 | 2 | Software management | 1 | NULL | 1 |
| 3 | 3 | Application request | 2 | NULL | 0 |
| 4 | 4 | Analysis | 3 | NULL | 1 |
| 5 | 5 | Package creation | 4 | NULL | 0 |
| 6 | 6 | Customization | 5 | Customization tools | 1 |
| 7 | 7 | Conflict resolution | 7 | NULL | 0 |
| 8 | 8 | Quality assurance | 8 | NULL | 1 |
| 9 | 9 | Release management | 9 | NULL | 0 |
| 10 | 10 | Validation | 6 | NULL | 1 |
| 11 | 11 | Product administration | 10 | NULL | 0 |

The default list contains several predefined groups. The value in the column Sequence defines

the order of the groups (as shown in the *tools configuration* screen). The Column Status is reserved for further use.

# Changing RayFlow Server

In order to create custom groups, a direct access to SQL database is required.

The RayFlow server URL is stored in `BenchConfig` table.

Results | Messages

| | Name | Value |
|---|---|---|
| 1 | Directory | c:\raypack\packbench\ |
| 2 | RayFlowServer | http://win-d25fl6tmfav/RayFlow/ |
| 3 | Version | 4.0.3 |

Simply update the value of a row with the name equal to `RayFlowServer` to a desired server URL. The clients have to be restarted in order to reflect the change.

# Advanced Logging Options

Besides the product configuration options that are available from the Settings section of the application UI, there are some additional configuration settings users may adjust to tailor PackBench towards their individual requirements.

## Logging PackBench Activity

The program data directory (`%AppData%\Roaming\RayBench\Logs`) is used by default to store the application activity log file (`{yyyy-MM-dd HH-mm-ss}.log`). If the default settings remain unchanged, PackBench adds a new line to this log file for every system DEBUG level message that is generated during application use.

By default, the log is to be used as in a rolling appender manner, which means that one global log file is permanently extended until a certain file size is reached. As soon as it is reached, the oldest lines are automatically transferred into an archive of log files, which is by default limited to 10 files. When this second limit is reached, the oldest archive is removed to free a slot for the newest archive file.

In order to change the default settings for the log file behavior, users have to manually edit the `log.config` file, which resides in the root of the installation folder of PackBench (usually something like `C:\Program Files (x86)\RayPack\PackBench`). The settings which are most likely to be of interest for adjustments are:

### Log File Storage Location

The log file is by default "`%appdata%\Raynet\RayBench\Logs\{yyyy-MM-dd HH-mm-ss}.log`". However, it is possible to define any other absolute local paths as well as shared network locations for logging resource storage.

```
<file type="log4net.Util.PatternString"
        value="%env{AppData}\\PackBench\\Logs\\%date{yyyy-MM-dd HH-mm-ss}.log" />
```

**Be aware:**
The user that runs PackBench must have write permissions in the log file location in order to initiate and maintain the message flow to the log file. If the user does not have sufficient access rights, there will be no error message, or actual product usage cutback, but simply a loss of system activity documentation. Please refer to the Troubleshooting section for additional instructions in case of missing logs.

### Max. Log File Size

The max. log file size may be defined as "`KB`", "`MB`" or "`GB`". The default setting for newly installed PackBench instances is "`2048KB`"

```
<maximumFileSize value="2048KB" />
```

## Log Level

The most frequently used log level settings are `DEBUG`, `INFO`, `WARN`, `ERROR`, `FATAL`, `OFF`, whilst `OFF` prevents logging at all, `FATAL` is the most restrictive but still writing setting, and `DEBUG` the most talkative option.
The recommendation is to use the `DEBUG` level for newly setup systems, since a lot of the information logged in this mode may help to adjust settings regarding access rights, and the like. As soon as the application and system are up and running productively, setting the log level to `WARNING` should be sufficient for permanent maintenance.

```
<level value="DEBUG" />
```

## Default Logging Configuration

The default configuration file is given below as a review and backup support:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>

  <configSections>
    <section name="log4net"
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
  </configSections>

  <log4net>
    <appender name="ConsoleAppender" type="log4net.Appender.ConsoleAppender">
      <layout type="log4net.Layout.PatternLayout">
        <conversionPattern value="[%-5p] - %m%n" />
      </layout>
      <threshold value="Info" />
    </appender>

    <appender name="RayBenchFileAppender"
type="log4net.Appender.RollingFileAppender">
      <immediateFlush value="true" />
      <file type="log4net.Util.PatternString" value="%property{LogFilePath}\%
date{dd-MM-yyyy-HH-mm-ss}.log" />
      <param name="StaticLogFileName" value="false" />
      <maximumFileSize value="10240KB" />
      <rollingStyle value="Once" />
      <maxSizeRollBackups value="2" />
      <layout type="log4net.Layout.PatternLayout,log4net">
        <param name="header" value="/***************************** LOG HEADER
******************************/&#xD;&#xA;" />
        <param name="footer" value="/***************************** LOG FOOTER
******************************/&#xD;&#xA;" />
        <param name="ConversionPattern" value="%d [%t] %-5p %c - %m%n" />
      </layout>
      <threshold value="INFO" />
    </appender>
```

```
    <root>
      <level value="INFO" />
      <appender-ref ref="ConsoleAppender" />
      <appender-ref ref="RayBenchFileAppender" />
    </root>
  </log4net>
</configuration>
```

### Further Information

PackBench uses an external library to provide logging functionality. Please refer to the online-documentation provided for the log4net project (http://logging.apache.org/log4net/) in order to get further details regarding available configuration and usage options. log4net can be adjusted to connect directly with databases or event-loggers. There are numerous options for layout and behavior manipulations. PackBench system administrators with a slight affection for perfection are highly welcome to configure their very own logger version.

# Configuration Files

## Client

Client configuration is stored in XML format in the following location:

```
%APPDATA%\RayBench\RayBench.config
```

Below is a sample content of the configuration file:

```
<?xml version="1.0"?>
<ClientConfig LastProjectId="0" ProjectSelector="true" Animations="true" Language="en"
 Mode="Standalone">
  <DatabaseConnection>
    <Type>MsSql2008</Type>
    <ServerAddress>PBPR05NB000079</ServerAddress>
    <DatabaseName>PackBench</DatabaseName>
    <InstanceName>SQLEXPRESS</InstanceName>
    <InMemory>false</InMemory>
    <FilePath>C:\db\db.db</FilePath>
    <IsTrusted>False</IsTrusted>
    <UserName>USERNAME</UserName>
    <UserPassword>USERPASSWORD</UserPassword>
  </DatabaseConnection>
  <ServerTcpConnection Host="localhost" Port="23624" />
</ClientConfig>
```

### ClientConfig Element

Contains the general settings of PackBench.

- **`Mode`**
  Two values are supported: `Standalone` to use a standalone configuration, and `ServerClient` to use client.

- **`LastProjectId`**
  Defines the identifier of last project in use.

- **`Animations`**
  When set to `True`, animated elements of the User Interface will be used. Set it to `False` to disable rich animations.

- **`Language`**
  This value is reserved for further use. Currently, only English is supported.

- **`ProjectSelector`**
  This value is reserved for further use.

## DatabaseConnection Element

This element defines how PackBench accesses the SQL database. This element is ignored in the Server-Client scenario.

- **`Type`** (REQUIRED)
  Defines the type of database to be used. Currently supported values are: `MsSql2008, MySql, MySql2012, SqlLite`. It is recommended to leave the default value MsSql2008 for full compatibility with SQL Server 2008 and newer. Support for SQLite and MySql is experimental.

- **`ServerAddress`** (REQUIRED)
  Name or IP address of the database server.

- **`DatabaseName`** (REQUIRED)
  The name of the database.

- **`InstanceName`**
  The name of the instance. When empty or missing, a default instance is used.

- **`IsTrusted`**
  Determines whether Windows Authentication will be used. When set to `False`, empty, or missing then SQL Authentication is used instead. SQL Authentication requires that the `UserName` and `UserPassword` elements are specified.

- **`UserName`**
  User name used to authenticate to SQL Server using SQL Authentication.

- **`UserPassword`**
  Password used to authenticate to SQL Server using SQL Authentication. This value may be written in open text, but when configured from PackBench UI it is always encrypted.

- **`InMemory`** (SQLite only)
  When set to `True`, then in-memory database mode will be used. When equal to `False`, empty, or missing then a single ordinary disk file will be used.

- **`FilePath`** (SQLite only)
  File path of the database.

### ServerTcpConnection Element

This element defines how PackBench accesses the server:

- **`Host`**
  The name or IP address of PackBench server. For standalone mode, this value must be always `localhost`.

- **`Port`**
  A numeric value specifying the port on which the connection occurs. Standard value is `23624`.

## Server

Server configuration is stored in XML format in the following location:

```
%PROGRAMDATA%\RayBench\RayBenchServer.config
```

Below is a sample content of the configuration file:

```xml
<?xml version="1.0"?>
<ServerConfig>
  <DatabaseConnection>
    <Type>MsSql2008</Type>
    <ServerAddress>PBPR05NB000079</ServerAddress>
    <DatabaseName>PackBench</DatabaseName>
    <InstanceName>SQLEXPRESS</InstanceName>
    <InMemory>false</InMemory>
    <FilePath>C:\db\db.db</FilePath>
    <IsTrusted>False</IsTrusted>
    <UserName>USERNAME</UserName>
    <UserPassword>USERPASSWORD</UserPassword>
  </DatabaseConnection>
  <PersistenceType>Database</PersistenceType>
  <RayFlowConnection />
  <Port>23624</Port>
</ServerConfig>
```

### DatabaseConnection Element

This element defines how PackBench accesses the SQL database.

- **`Type`** (REQUIRED)
  Defines the type of database to be used. Currently supported values are: `MsSql2008, MySql, MySql2012, SqlLite`. It is recommended to leave the default value MsSql2008 for full

compatibility with SQL Server 2008 and newer. Support for SQLite and MySql is experimental.

- **`ServerAddress`** (REQUIRED)
  Name or IP address of the database server.

- **`InstanceName`**
  The name of the instance. When empty or missing, a default instance is used.

- **`DatabaseName`** (REQUIRED)
  The name of the database.

- **`IsTrusted`**
  Determines whether Windows Authentication will be used. When set to `False`, empty or missing then SQL Authentication is used instead. SQL Authentication requires that the `UserName` and `UserPassword` elements are specified.

- **`UserName`**
  User name used to authenticate to SQL Server using SQL Authentication.

- **`UserPassword`**
  Password used to authenticate to SQL Server using SQL Authentication. This value may be written in open text, but when configured from PackBench UI it is always encrypted.

- **`InMemory`** (SQLite only)
  When set to `True`, then in-memory database mode will be used. When equal to `False`, empty or missing then a single ordinary disk file will be used.

- **`FilePath`** (SQLite only)
  File path of the database.

## Port Element

A numeric value specifying the port on which the server is listening occurs. Standard value is `23624`.

## PersistenceType Element

This value is intended for a further use and should not be changed.

## RayFlowConnection Element

This value is intended for a further use and should not be changed.

# Reference and Cheat Sheets

## PackBench Variables

This chapter contains a reference of variables that can be used in various configuration places. Bear in mind that certain variables are available only in specific contexts. The use of the *Variables Editor* is recommended, as it can recognize the context and suggest the right set of variables available for processing.

### Run Variables

The following variables are available in the run-context. Some of them can be accessed only from a task.

| Variable name | Description | Notes |
|---|---|---|
| `PkgAppName` | Represents the application name of the run package. | |
| `PkgLanguage` | Represents the language of the run package. | |
| `PkgVersion` | Represents the version of the package run. | |
| `PkgVendor` | Represents the name of the vendor of the package run. | |
| `PkgFileName` | Represents the full path to the entry file name of the run package. | |
| `PkgName` | Represents the name of the run package. | |
| `PkgCreated` | Represents the date of creation of the run. | |
| `PkgDir` | Represents the full path to the package directory. | This value does not have a trailing backslash. |
| `PkgPlatform` | Represents the human-readable platform of the run package. | |
| `RunStatus` | Represents a human-readable status of the run package. | |
| `ToolPath` | Represents a full path to the tool assigned to a task. | Available only in task-context. |
| `ToolStatus` | Represents a human readable status of a tool's visibility. | Available only in task-context. |

| Variable name | Description | Notes |
|---|---|---|
| `WorkflowName` | Represents the name of the workflow. | |
| `WorkflowId` | Represents the numeric identifier of a workflow. | |
| `WorkflowStatu` | Represents a human-readable status of the workflow. | |
| `ProjectId` | Represents the numeric identifier of a project. | |
| `ProjectStatus` | Represents a human-readable status of the project. | |
| `ProjectPath` | Represents the default path where projects are saved. | This value is already resolved. |
| `ProjectName` | Represents the name of the project. | |

# General Variables

The following variables are available in all contexts:

| Variable name | Description | Notes |
|---|---|---|
| `DateTime` | Inserts the current date and time. | This value may be not safe to use as file or directory name. In order to make it safe, use `Safe` modifier, for example (`$DateTime->Safe`). |

# PackBench and RaySuite Products

The following variables which represent installation paths of selected RaySuite products are available in all contexts:

| Variable name | Description | Notes |
|---|---|---|
| `BenchDirector` | Represents a full path to the root PackBench directory, as defined during the installation process. | This value ends with a trailing backslash. |
| `RayPackDirect` | Represents the full path to *RayPack* directory. | This value ends with a trailing backslash. The value is not resolved if |

| Variable name | Description | Notes |
|---|---|---|
| | | *RayPack* is not installed on the current machine. |
| RayQCDirector | Represents the full path to *RayQC* directory. | This value ends with a trailing backslash. The value is not resolved if *RayQC* is not installed on the current machine. |
| RayQCAdDirect | Represents the full path to *RayQC Advanced* directory. | This value ends with a trailing backslash. The value is not resolved if *RayQC Advanced* is not installed on the current machine. |
| RayEvalDirect | Represents the full path to *RayEval* directory. | This value ends with a trailing backslash. The value is not resolved if *RayEval* is not installed on the current machine. |
| PackPointDire | Represents the full path to *PackPoint* directory. | This value ends with a trailing backslash. The value is not resolved if *RayPack* is not installed on the current machine. |

# Folder Naming Conventions

The following variables which represent names for predefined folders are available in all contexts:

| Variable name | Description | Notes |
|---|---|---|
| ProjectSource | The name of folder for original sources. | This is a relative path. |
| ProjectDocsFo | The name of folder for documentation. | This is a relative path. |
| ProjectRepack | The name of folder for repackaged files. | This is a relative path. |
| ProjectOutput | The name of folder for output files. | This is a relative path. |
| ProjectWorkin | The name of folder for working files. | This is a relative path. |

# Value Modifiers

In order to modify the value (for example make it UPPERCASED or safe for a file name) use variable modifiers, described in chapter *Variable modifiers*.

# Variable Modifiers

In PackBench it is possible to use special modifiers to transform the value of variables. For example, to get only the file name from a variable that contains a full file path, the following syntax can be used:

```
($MyVariable->FileName)
```

If MyVariable is equal to `C:\temp\test.exe`, the result of the above syntax will only be "`test.exe`".

The following modifiers are possible:

| Modifier | Example | Description |
|---|---|---|
| `FileName` | `($MyVariable->FileName)` | Gets the file name from the value of a variable (for example: `C:\temp\test.exe` returns `test.exe`). |
| `Extension` | `($MyVariable->Extension)` | Gets the extension from the value of a variable (for example: `msi` or `txt`, without the leading dot). |
| `FileNameWithoutExtension` | `($MyVariable->FileNameWithoutExtension)` | Gets the file name without the extension from the value of a variable (for example: `C:\temp\test.exe` returns `test`). |
| `DirectoryName` | `($MyVariable->DirectoryName)` | Gets the directory path from the value of the variable or a path to the parent directories if the value is a directory (for example: both `C:\temp\test.exe` and `C:\temp` will return `C:\temp`). |
| `ToUpper` and `ToLower` | `($MyVariable->ToUpper)` | Returns an uppercased/lowercased value. |
| `IsNullOrEmpty` | `($MyVariable->IsNullOrEmpty)` | Returns `1` if the value of the variable is null or empty. Otherwise it will return `0`. |
| `IsNotNullOrEmpty` | `($MyVariable->IsNotNullOrEmpty)` | Returns `0` if the value of the variable is null or empty. Otherwise it will return `1`. |
| `Safe` | `($MyVariable->Safe)` | This can be used to create a safe path (for example: `Folder \ TEst >< test` will become `FolderTEstTest`). |

# Regular Expressions

Regular expressions (abbreviated to RegExp) are sequences of characters forming a search pattern that can be used to identify a textual material of a given pattern. The syntax of regular expression is standardized and used along many different products and operating systems, particularly for finding and replacing of functionalities.

PackBench uses regular expressions to provide a flexible way of defining compact text conditions, which can be further customized to exclude and include particular resources based on complicated rules and logic.

The following cheat sheet presents some commonly used regular expression patterns and example patterns with explanations.

| Phrase | Description | Examples |
|---|---|---|
| Letters or digits | Literal meaning, case sensitive | `Abc` matching `Abc` but not `abc` |
| Pipe (`\|`) | Alternative | `A\|b` matching `A`, `b`, `Ab` but not `BC` |
| Asterisk (`*`) | Zero or more instances of the previous set | `A*b` matching `Aaaaab`, `Ab`, `bcd` but not `Ac`<br>`A\|b*` matching `a`, `b`, `bb`, `bbb` but not `CDE` |
| Plus (`+`) | One or more instances of the previous set | `A+b` matching `Ab`, `AAAb` but not `b`<br>`(a\|b)+cd` matching `acd`, `bcd`, `aaacd`, `abcd`, `bacd` but not `cd`<br>`Folder[0-9]+` matching `Folder1`, `Folder2`, `Folder21` but not `Folder` or `FolderA` |
| Question mark (`?`) | Zero or one instance of the previous set | `Ab?(c\|d)` matching `Ac`, `Ade`, `Abcd` but not `Abb` |
| Round brackets ( and ) | Grouping of sets | `(abc\|def)ghi` matching `abcghi`, `defghi`, `abcdefghi` but not `abcdef`<br>`(a\|b\|c)?def` matching `adef`, `cdefghi`, `def` but not `abc` |
| Caret (`^`) | On the beginning of the string marks that no characters are allowed before | `^abc` matching abc, abcdef but not defabc<br>`^(folder1\|folder2\|folder3)\\test` matching `folder1\test`, `folder2\test\test2` but not `C:\folder1\test` |
| Dollar (`$`) | At the end of the string marks that no characters are allowed after | `:\\test\\folder$` matching `C:\test\folder`, `D:\test\folder` but not `C:\test\folder1` |
| Square brackets [ and ] | Allowed set of characters | `[a-z]` matching `abc`, `def` but not `123`<br>`^[A-D]:\\Test\\` matching `C:\Test\` and `D:\Test\` but not `E:\Test\`<br>`folder[a-zA-Z0-9]` matching `folderA`, `folder1` but not `folder\test` |

| Phrase | Description | Examples |
|---|---|---|
| Caret in square brackets `[^]` | In square brackets – negation of character set | `C:\\folder[^\\]*\\test` will match `C:\folder\test` but not `C:\folder\subfolder\test` |
| Dot (`.`) | Any character | `Folder.\\test` will match `FolderA\`, `Folder4\`, `Folder$\` but not `Folder\` |
| Backslash (`\`) | Escape character | `C:\\test\\test2` matches `C:\test\test2` but not `C:\\test\\test2`<br>`Test\*` matches `Test*`<br>`Test\(a\)` matches `Test(a)` but not `Testa` |
| `(?i)` | Used at the beginning on the expression – do not match case | `(?i)abc` will match `abc`, `Abc`, `ABC123` but not `123` |

# Examples

```
^(?i)[{(]?[0-9A-F]{8}[-]?([0-9A-F]{4}[-]?){3}[0-9A-F]{12}[)}]?$
```

| Matches | Does not match |
|---|---|
| `{057B21232-ED42-11CE-BACD-00AA0CA7623}`<br>`(057B21232-ED42-11CE-BACD-00AA00CA7623)`<br>`057b21232ed4211cebacd00aa00ca7623`<br>`057B21232-ED42-11CE-BACD-00AA00CA7623` | `(123)`<br>`123-1412-1231`<br>`abc`<br>`aaa(057B21232-ED42-11CE-BACD-00AA00CA7623)` |

```
^(?i)(%windir%)\\Installer$
```

| Matches | Does not match |
|---|---|
| `%windir%\Installer`<br>`%windir%\INSTALLER` | `%windir%\Installer\123-123.msi` |

```
^(?i)%ProgramFiles\(x86\)%\\Common Files\\(InstallShield|Wise Installation)$
```

| Matches | Does not match |
|---|---|
| `%ProgramFiles(x86)%\Common Files\InstallShield`<br>`%ProgramFiles(x86)%\COMMON FILES\Wise Installation` | `%ProgramFiles(x86)%\Common Files\Microsoft`<br>`%ProgramFiles%\Common Files\InstallShield`<br>`%ProgramFiles(x86)%\Common Files\`<br>`%ProgramFiles(x86)%\Common Files\Wise Installation\Subfolder` |

```
^unins(|t|tall)\d*\.(cif|cfg|dat|dll|ini|exe|xml|lnk)$
```

| Matches | Does not match |
|---|---|
| uninst.exe<br>uninst_myApp.lnk<br>uninstallapp.cfg<br>uninst.ini | MyProgram_uninstall.cfg<br>uninstall.txt<br>unins.txt |

```
^_isreg32\.dll$
```

| Matches | Does not match |
|---|---|
| _isreg32.dll | _ISREG32.dll<br>isreg32.dll<br>_isreg.dll<br>_isreg32.dll.backup |

```
^(?i)(HKEY_LOCAL_MACHINE|HKEY_CURRENT_USER)\\(Software\\Wow6432Node|Software)
\\InstallShield
```

| Matches | Does not match |
|---|---|
| HKEY_LOCAL_MACHINE\Software\InstallShield<br>HKEY_LOCAL_MACHINE\Software\Wow6432Node<br>\InstallShield<br>HKEY_CURRENT_USER\SOFTWARE\Wow6432Node<br>\InstallShield<br>HKEY_CURRENT_USER\Software\INSTALLSHIELD | HKEY_CLASSES_ROOT\Software\InstallShield<br>HKEY_CURRENT_USER\Software<br>\Progrems\InstallShield |

```
^%USERPROFILE%\\(.+\\)?Temp$
```

| Matches | Does not match |
|---|---|
| %USERPROFILE%\Temp<br>%USERPROFILE%\test\Temp<br>%USERPROFILE%\test\test2\Temp | %USERPROFILE%\Temp2<br>“%USERPROFILE%\Temp”<br>%userprofile%\Temp |

# User Interface Shortcuts

The PackBench user interface offers a set of shortcuts for those packagers, who are used to leave the mouse aside and swiftly work their way through task sequences by keyboard.

| Scope | Hotkeys | Action |
|---|---|---|
| *Task list in Run view* | `Space` | Marks the current task as completed and goes to the next task. |
| *Task list in Run view* | `Ctrl + Space` | Marks the current task as skipped and goes to the next task. |
| *File list the Overview section of a run* | `F5` | Refreshes the view |

**Tip:**
Any time you access a context menu on any PackBench item, available hot-key combinations (if any) are listed next to each action..

# Help and Support

## Request PackBench Support

Our Raynet support team gladly assists you on any questions or issues you encounter regarding PackBench. Feel free to sign in and open incidents via our Raynet Support Panel.

## Join the RayPack Community

The RayPack community resides within our Knowledge Base .Once you have signed up for access to the Raynet support panel, you automatically have access to the Knowledge Base, too.

You will surely come to a point where you would love to suggest a new feature for the future development of PackBench. Maybe you need to find some tips & tricks to hit your target right. The RayPack community is your place for discussing such topics, for sharing and expanding your own experience.

Step in contact with Raynet's professional application management and packaging teams and learn how to polish your activities to a level of highest quality standards. Since Raynet has years and years of experience in the software management and packaging business, we know what to do, and how to do it. Don't row your boat alone when you have the chance to join our PackBench community for free.

## Contact Your Raynet Sales Representative

Our sales team is the right contact for any license or edition question you might encounter. You would like to benefit from a professional PackBench training? Ask for dates and locations to find the fitting training occasion. You are highly welcome to step in contact via sales@raynet.de.

# Additional Information

Visit www.raynet.de for further information regarding the product and current community incentives. It's also recommended taking a look at additional resources available at the Knowledge Base for Raynet products:
http://knowledgebase.raypack.net.

Raynet is looking forward to receiving your feedback from your PackBench experience. Please contact your Raynet service partner or use our Support Panel to add your ideas or requirements to the PackBench development roadmap!

Raynet GmbH
Technologiepark 20
33100 Paderborn, Germany
T +49 5251 54009-0
F +49 5251 54009-29
info@raynet.de
support@raynet.de

www.raynet.de