# RAYQC®

## Quality Control
## for Software Packaging

Getting Started with RayQC
7.0

RayQC is part of RaySuite.

•rayNET

RayQC 7.0 Getting Started with RayQC

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

# Welcome to RayQC!

RayQC is a flexible "rule-based" quality control tool, which allows its users to create XML based checklists and execute them in both manual and automatic modes. The ability to extend RayQC functionality via plug-in interface and adapting it as per the user requirements, makes RayQC an extensible and versatile QC tool, which can be used all across the application life cycle.

Key Features

- Highly interactive user interface.

- Single point of access to major functionalities via RayQC **Dashboard**.

- Easily create checklists with the graphical Checklist Editor interface.

- XML based checklist structure.

- Auto-validation of created checklist templates.

- Save the result state of a checklist as a project file to resume working on it at a later time.

- Functionality extension via plug-in interface.

- Example checklists assist users to create their very first Hello World checklists in no time.

- Export test results in various formats.

- Manual, semi-automatic, and automatic modes of operation.

- Extendability with Application Compatibility and Windows Installer Check modules.

- Integration with other RaySuite products.

## In this Guide

This guide is meant to provide an overview of the frontline functionalities offered by RayQC and to assist with first steps in this astonishing new world of quality assurance.

## System Requirements and Product Activation
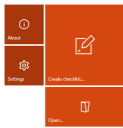
[Hardware Requirements](#)


[Software Requirements](#)


[Product Activation](#)

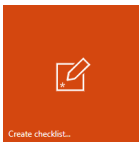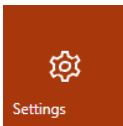## Home Snap-in


[Dashboard](#)


[Recent](#)


[Menu Bar](#)

## Create New Checklist


[Create checklist](#)

## Settings


[Settings](#)

## Open an Existing Checklist


[Open checklist](#)

## Open a Checklist Project

**Open project**

## Need Help?
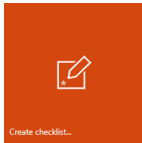
**Request RayQCsupport**

**Contact your Raynet sales representative**

## Additional information

Visit www.rayqc.de for further information regarding the product and current community incentives.

Raynet is looking forward to receiving feedback from your RayQC experience. Please contact your Raynet service partner or use our Support Panel to add your ideas or requirements to the RayQC development roadmap!

# System Requirements

The given requirements name prerequisites for devices running the RayQC application.

## Hardware Requirements

### Minimal

- CPU Pentium IV / Core2 processor
- 2 GB RAM
- 1 GB free hard disk
- 1280x1024 screen resolution

### Recommended

- CPU Intel Core i5 or i7
- 8GB RAM
- 40 GB free hard disk (software library usage)

## Supported OS

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

- Windows Vista
- Widnows 7
- Windows 8
- Windows 8.1
- Windows 10

- Windows Server 2008 SP1-SP3
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016

## Prerequisite Software

- .NET 4.5 Client & Full for Windows Vista up to Windows 8 systems (both 32-bit and 64-bit)
- Windows Management Framework (including Windows PowerShell 3.0, WinRM 2.0, and BITS 4.0)
  Please verify if the named or later versions are available on your device before using internal or external plug-ins in checklists.
  For further details and download resources, visit http://support.microsoft.com/en-us/kb/968929.

**Be aware:**
In order to be able to use external plug-ins with RayQC, it has to be ensured that the PowerShell version supported by the device that hosts the application matches the PowerShell version of the

actual plug-in script. It is highly recommended to synchronize the PowerShell version among all
devices that are assigned for QA execution to prevent compatibility issues in the first place.

**RayFlow**

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

**Hyper-V integration**

- Both host and guest machine must have PowerShell 3.0 or newer installed.
- Windows Remote Management
- RayPack Studio Tools for Hyper-V need to be installed on the guest machine.

The tools can be installed from a Windows Installer package that is present in the RayPack
subfolder `Tools\HyperVTools\Packaging Suite Tools for Hyper-V.msi`.
The installation of the tools is required, so that the user can see interactive prompts and windows
on Hyper-V machines. It is recommended to install the tools as a part of the base snapshot.

**VMware Workstation / ESX integration**

RayQC supports the following products:

- VMware vSphere 5.5 and newer
- VMware Workstation 10 and newer
- VMware Workstation 7, 8, 9 and for VMware vSphere 4.x, 5 and 5.1 are experimentally
  supported.

To use any of VMware Workstation / ESX machines, one of the following must be installed in an
appropriate version:

- VMware workstation
- VMware VIX API (https://my.vmware.com/web/vmware/details?
  productId=26&downloadGroup=VIX-API-162)
- vSphere

The required VIX API version depends on the systems that it needs to connect to. The below
table presents the supported versions of VMware products depending on the installed VIX API
version.

| VIX API Version | VMware Platform Products | Library Location |
| --- | --- | --- |
| 1.11 | vSphere 5, Workstation 8 or earlier | Workstation-8.0.0-and-vSphere-5.0.0 |
| 1.12 | vSphere 5.1, Workstation 9 or earlier | Workstation-9.0.0-and-vSphere-5.1.0 |
| 1.13 | vSphere 5.5, Workstation 10 or earlier | Workstation-10.0.0-and-vSphere-5.5.0 |
| 1.14 | Workstation 11 or earlier | Workstation-11.0.0 |
| 1.15.0 | Workstation 12 or earlier | Workstation-12.0.0 |

# Additional Resources

Further information regarding RayQC can be found in several resources which are available.

- The *Operations Supplement* provides information about third-party software packages redistributed with RayQC.

- The *Release Notes* provide an overview about the changes and new features that are part of this version of RayQC.

- The *User Guide* provides detailed information about the RayQC Advanced Collision management tool.

- The product website www.rayqc.de 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.

# Product Activation

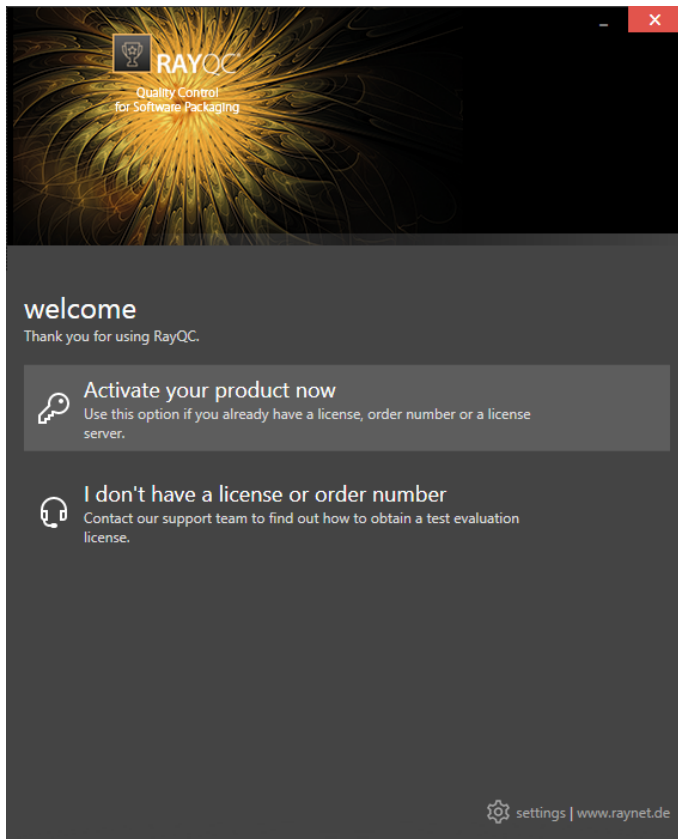The product can be activated using one of the following methods:

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

If RayQC 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.
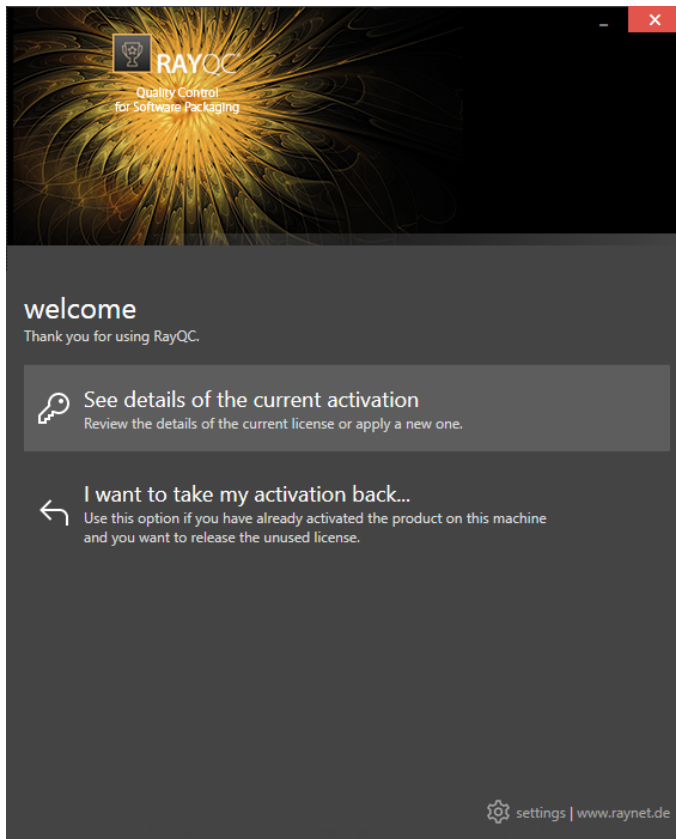
## License Wizard

This section describes the usage of the licensing wizard.

On the initial start of RayQC, the licensing wizard is shown. If the need to transfer an existing license arises, the license wizard 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 been already activated

## Activate your product now

This option should be used to activate the product using one of the following methods:
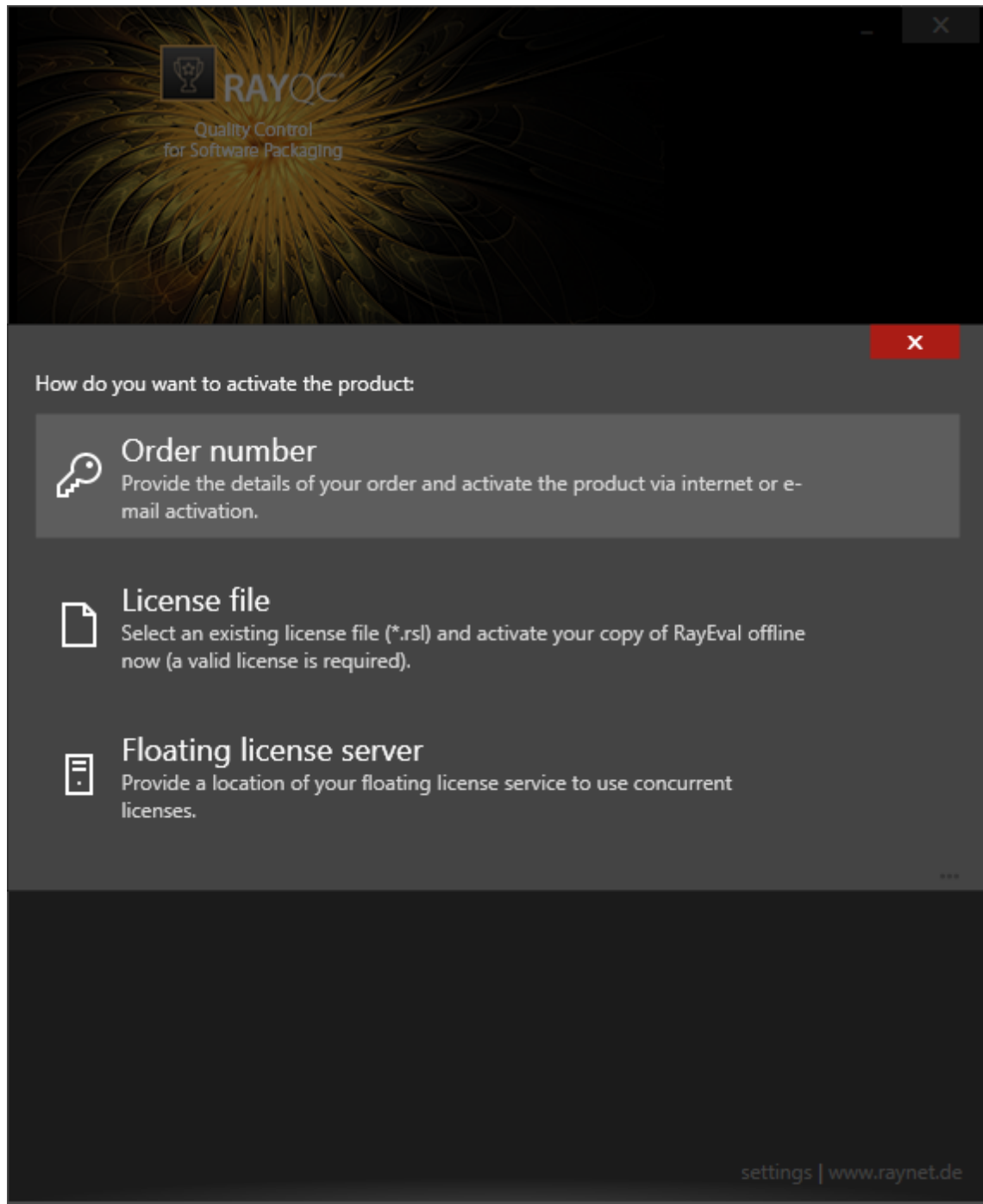
- **Order number**
  Online activation using a valid order number received from Raynet (recommended for most users)

- **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 to reactivate the product using a different order number or a 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 RayQC. For in-depth information please read this *section*. This options is only visible if the product has been already activated.

Once the license file has been generated or copied to the correct location the following will be shown...
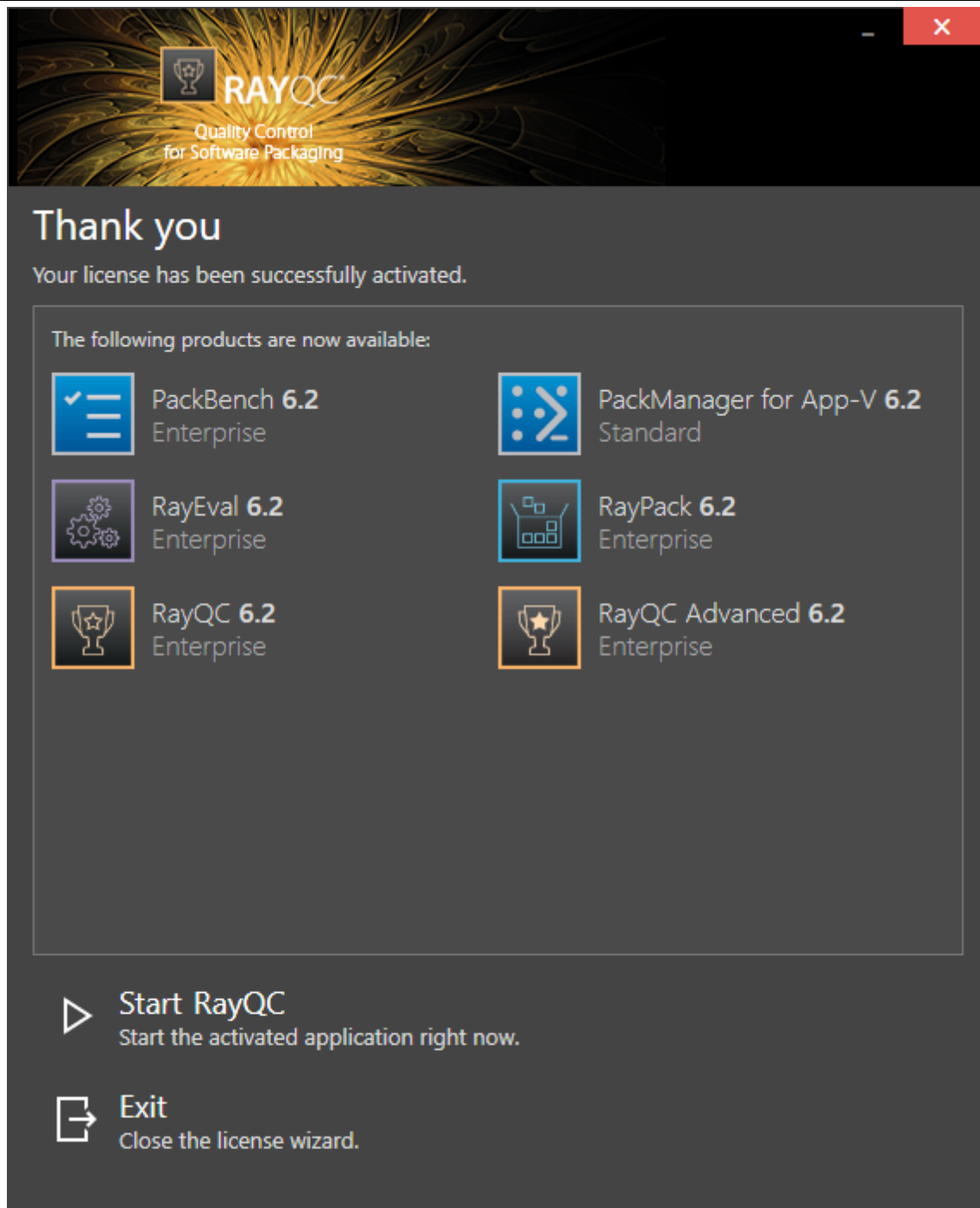
> **Note:**
> Depending on the license, more available products may be shown, as pictured above.
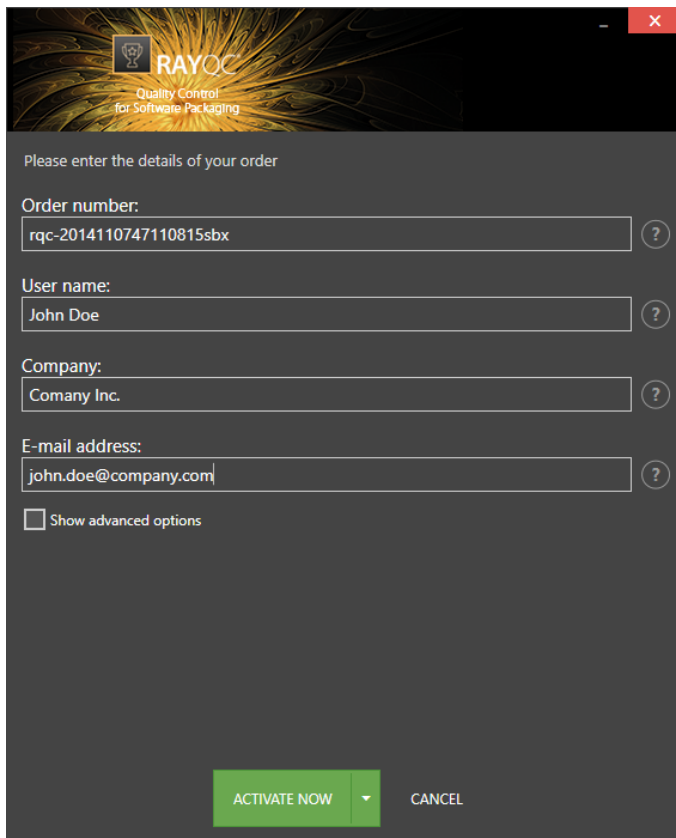
Then the option of starting RayQC or just closing the activation wizard is made available.

## Troubleshooting

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

## Order Number

RayQC 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 RayQC (in the same location as the `RayQC.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 RayQC 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 RayQC. It does not need to be the same name used to order RayQC.

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

**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 RayQC are shown.

**Hardware ID:**
This is a 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.
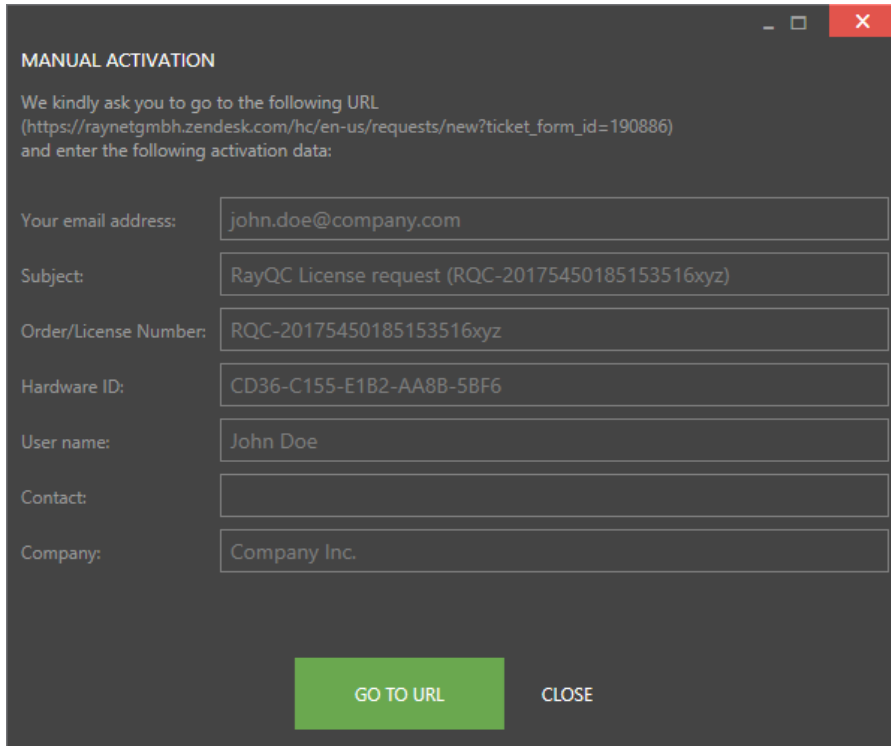
**Transfer the license**
If this option is selected, the order number and details may be used to activate RayQC on a second machine, that has differing hardware (which obviously has a different Hardware ID). This assumes that RayQC has been deinstalledfrom the machine on which it was previously activated 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:
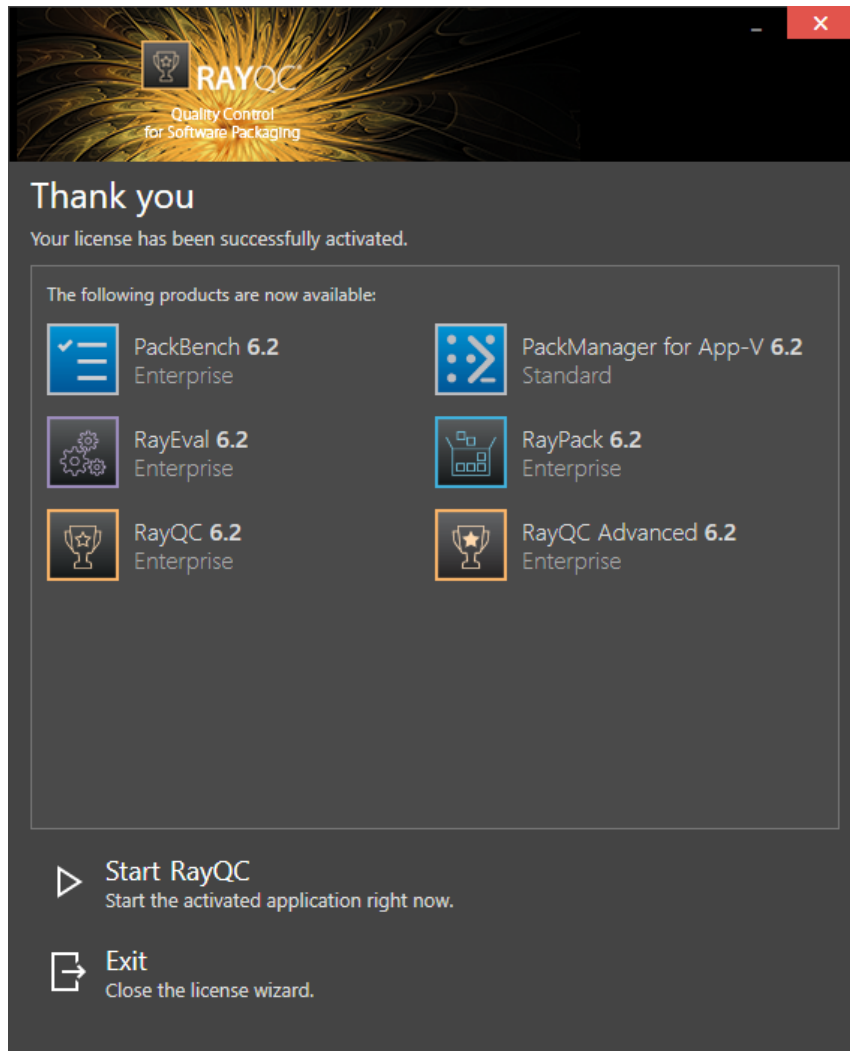
> **Note:**
> Depending on the license, more available products may be shown. As an example, see the image above.

The option of starting RayQC 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 RayQC.

## License File

If a license is already available, or a license file has been received as a result of activating RayQC via email, then all that is required is to copy the license file into the installation directory of RayQC (the directory in which the `RayPack.exe` resides). Clicking on the **I have a license** button

on the **License wizard** dialog opens a dialog box which allows to choose the license file. Once chosen, the file will be copied automatically to the RayQC installation directory. Please ensure that sufficient permissions to allow the creation / copying of a file to the installation directory of RayQC are available.
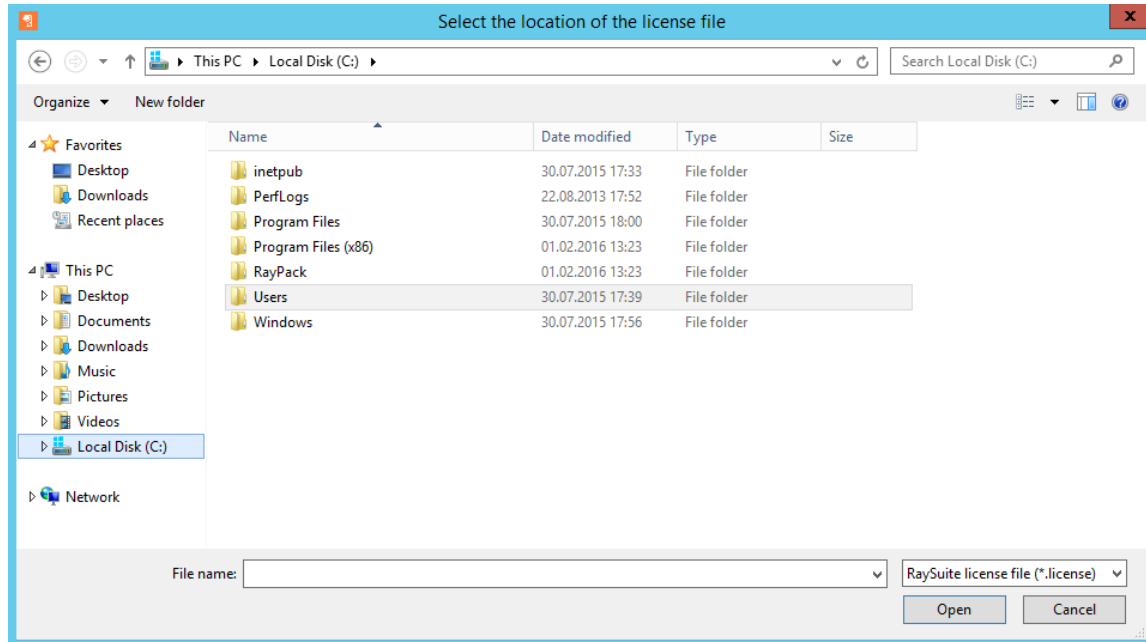


Once the license file has been copied to the correct location the following will be shown:
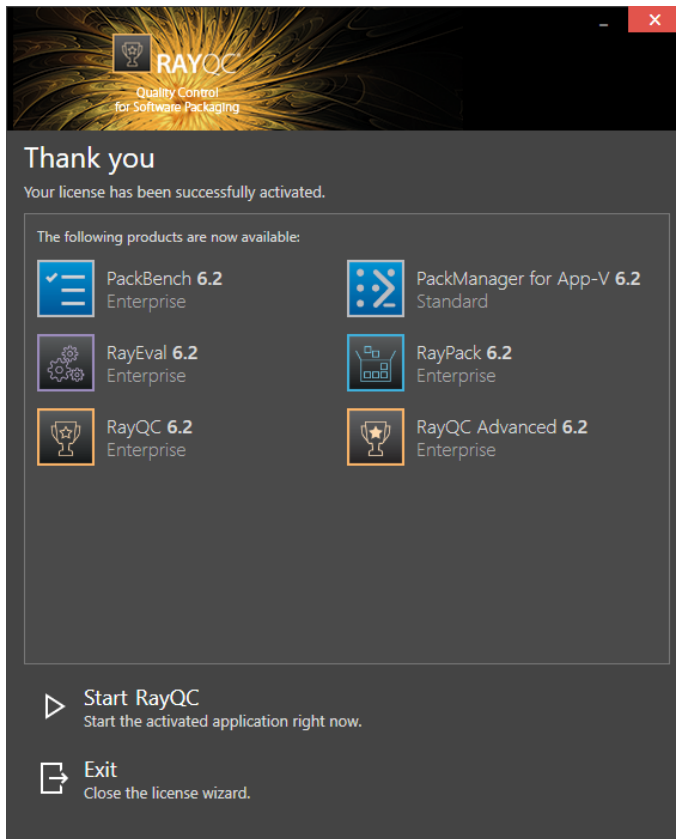
**Note:**
Depending on the license, more available products may be shown. As an example, see the image above.

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

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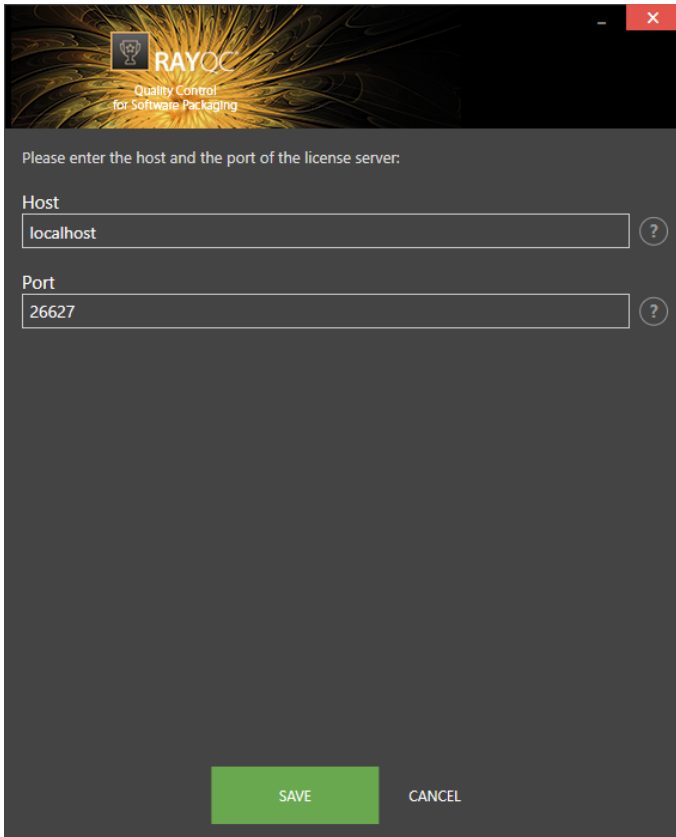
If there are any problems during the activation process, please contact our *help desk* for receiving assistance in activating RayQC.

## Floating License Server

RayQC 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 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 RayQC. This allows potential customers to test and work with RayQC 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 RayQC.

## I Want to Take My Activation Back

Deactivating an existing license for RayQC 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 RayQC Installation

1. Launch RayQC 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 RayQC 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 RayQC 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 RayQC.

# The Home Screen

Once RayQC is installed on a machine, launching the program executable from the application installation directory (e. g., `C:\Program Files(x86)\RayQC`) invokes the RayQC *home screen*. For those who have worked with any other RaySuite application before, the Home screen schema should be quite familiar, since it is a common interface type for all products of this family.
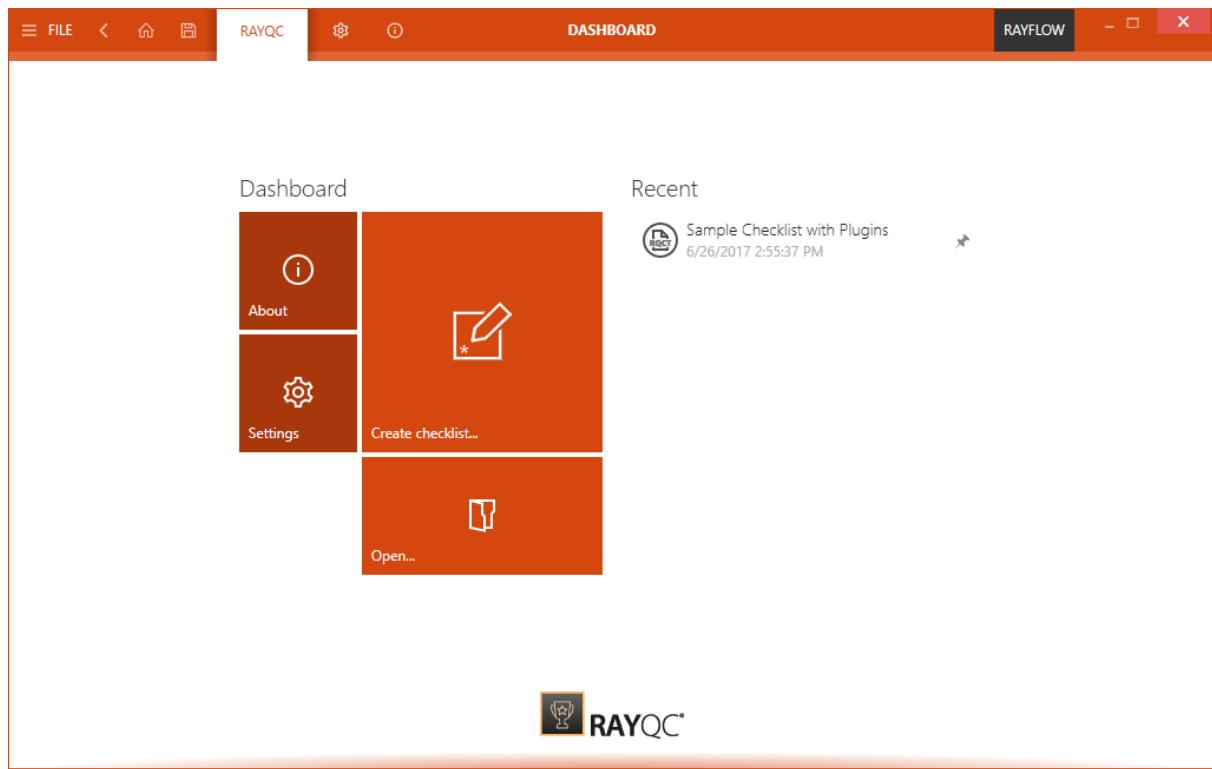


**RayQC Home Screen**

When RayQC is invoked as a tool via RayFlow, it usually does not load the home screen, but directly opens a checklist template or project file.

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

## The Main Tool Bar

Throughout RayQC the Main Tool Bar is visible, which, depending on the contents of the view shown, adds or removes menu items dynamically. As a rule of thumb, the items shown below are always present on the Main Tool Bar.

RayQC Main Toolbar

## Home
Choosing this button will return the user to the Home Screen. If any projects and or files are opened, and there is a requirement to save any changes, there will be a prompt to save before returning to the Home Screen.

## Save
Saves the current file / project. This button is only active if any changes have been made and required saving.

## 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 was 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 and module.

## Window title
The window title displays the current scope of activity. If an editor is active, the file name of the currently opened project is part of the window title as well.

## 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.

## FILE
This opens the **FILE** menu. The **FILE** menu is dynamically created, depending on what Tool is currently active. Please refer to the *the FILE menu* section to read more about it.

## Application Context
The illustrations above show the main application context status, which is RayQC. Depending on the set of licensed modules and add-ons, further contexts may be added.

## SETTINGS
Opens the s *ettings* for RayQC.
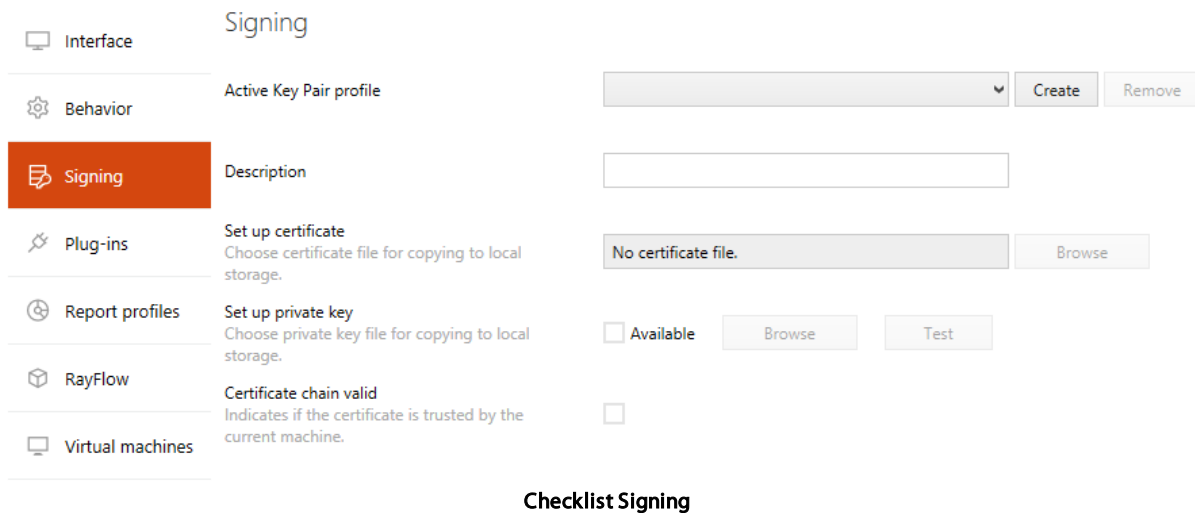
## ABOUT
Contains information about the currently active product instance.

**HELP**
Opens this help file.

## The Content Area

The content area is the core for actual application activity. According to the active application context, it contains the dashboard (as shown in the screenshot at the beginning of this topic), the checklist viewer or editor, dialogs for settings editing, and so on.



**Checklist Signing**

## The Swipe Bars

While the Main Tool Bar is designed to provide access to general application functionality, there are additional swipe bars for local activity options at the bottom of some views. This may either be a set of buttons for running checklist evaluations, options for saving or discarding settings, or navigation helpers that allow to switch between related views, such as between the checklist viewer and editor.



This is a swipe bar as it is used within the settings area, with buttons to accept or cancel changes at its right.

The Home screen of RayQC contains a Dashboard and an optional Recent section for quick file access. Whenever the application is launched without a specific file trigger, this view is the starting point for the new RayQC session.

The contents of the Home screen may be configured via the **Settings** area. Users can decide whether or not the Recent list is visible at the right-hand side of the Dashboard. The screenshots

below show the two operational modes - the first one without the Recent list, the second one with the visible Recent list.

## The Dashboard Tiles
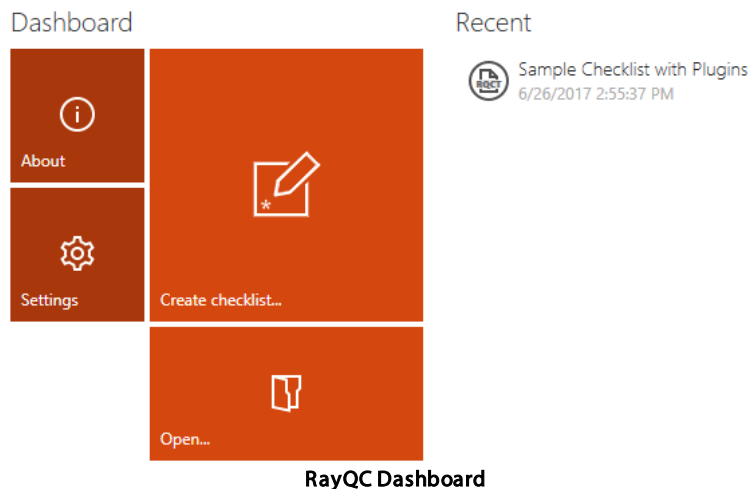

**RayQC Dashboard**

Clicking any of the tiles on the **Dashboard** opens a specific RayQC view:

- For first time users, it is highly recommended to take a look at the *Get Started Guide*, available from the **about** area.

- After gaining overall knowledge of RayQC, a walk through the **settings** section should be performed. Especially for those RayQC instances, that have to operate connected to a RayFlow server.

- With a properly set up configuration, the first checklist should be created, which requires clicking on the **create checklist** tile.

- Once a checklist is created, it may be opened for evaluation execution or structural editing. Either way, the **open checklist** allows to select a checklist template from the file system.

- And finally, when a checklist has been saved as a project, and the project has to be opened along with the latest status of element test results, the **open project** tile has to be clicked.

## The Recent List

If configured to be visible, the **Recent List** is shown on the right-hand side of the **Dashboard** content area. It lists recently accessed checklists and files of the currently logged in Windows User Profile.

The list contains the last six files that have been opened in RayQC by default. The seventh

opening of a file removes the oldest item from the recent list in order to add the new one. Clicking on a list item immediately opens the related file in RayQC.

---

**Be aware:**
If the physical file that is connected to a recent list item has been removed or renamed, it cannot be found and opened any more. Usually RayQC checks the availability of the recent objects whenever the Home screen is loaded.
However, if an item becomes physically unavailable whilst the Home screen is active, RayQC will display a message when the item is clicked, and ask the user if the file should be removed from the recent list. It is recommended to do so.

---

The default behavior outlined above may be changed by pinning files to the recent list. At the left hand side of each recent list item, there is a pin icon. It is gray for items that are not pinned to the list, and black for those who are currently pinned for permanent visibility. Clicking the pin icon switches between the pinned and unpinned status of an item.

---

**Note:**
Unpinning an item from the recent list may remove it from the visible list scope if it has not been opened in RayQC for a while. The rule is, that all pinned files are shown chronologically by last opening date, and the free slots of the list are filled with the recently opened files, which are as well ordered by last opening time. Therefore, if a pinned file has not been opened lately, it may very well be crowded out by other files.

---



Recent Item Menu

The file icons used within the list indicate the specific file type by its color: template files are represented by dark orange icons, projects have a lighter orange and a checkmark within their icon.

Right-clicking an item present within the recent list reveals the context menu. Users have three options to select from:

- **Remove from this list**: Removes the checklist item from the recent list.
- **Show folder in Windows Explorer**: Opens a new explorer window which shows the folder containing the checklist.

# The File Menu

Clicking the **FILE** button contained within the Main Tool Bar opens the file menu. This menu allows users to quickly access common functions.
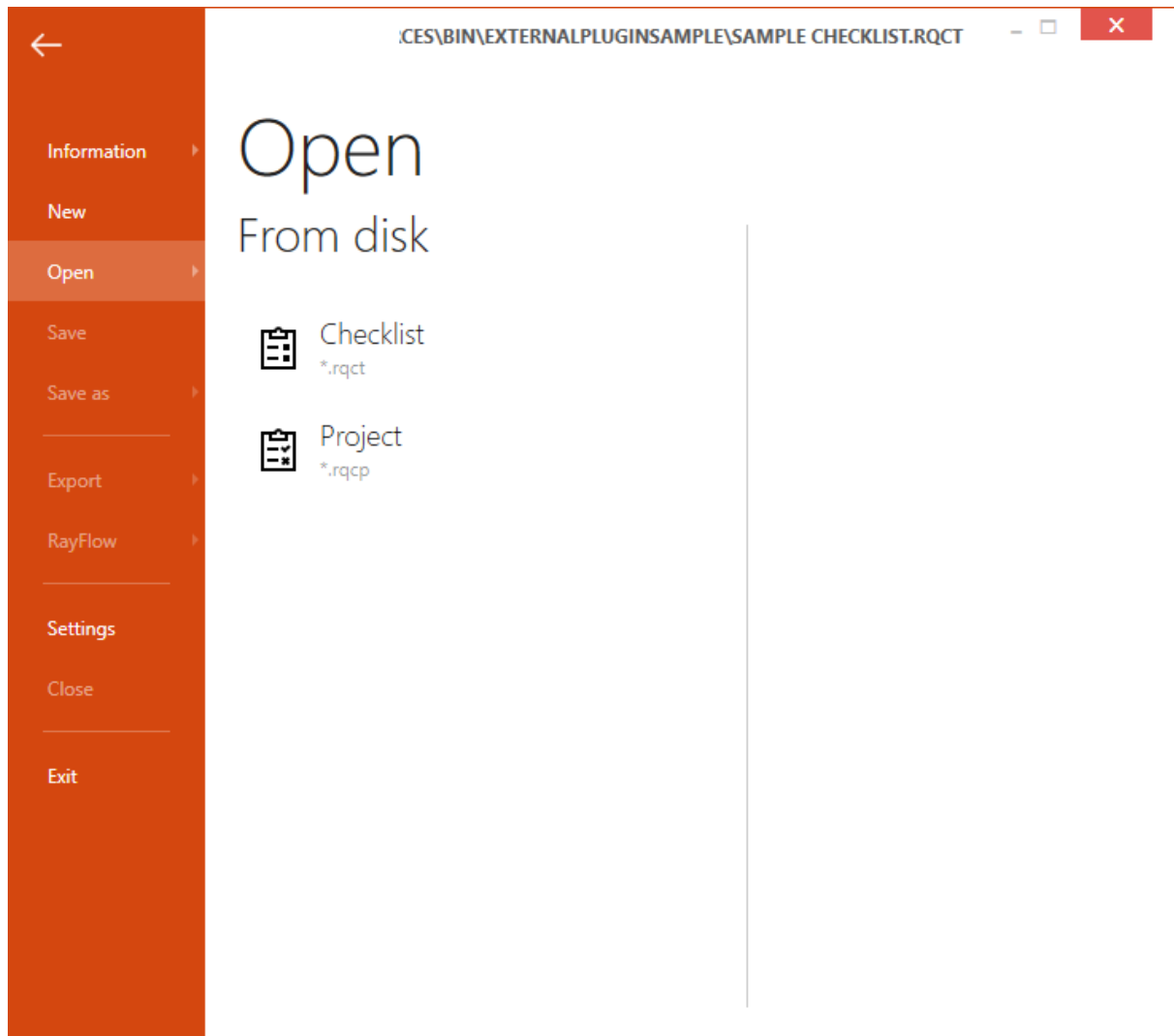


**File Menu**

## New

Opens the dialog that allows to create a new checklist file. If a checklist or project is loaded and there are outstanding changes, the user will be asked to save them before continuing.

## Open

Open an existing checklist or project. If a checklist or project is loaded and there are outstanding changes, the user will be asked to save them before continuing.

### Save

Saves any outstanding changes in the currently open checklist or project. Please note that this button is only active when pending changes are detected.

### Save as

Enables the saving of the currently opened checklist or project under another name, location or file type.

### Create Report

This option calls the create report dialog for the definition of target file name and type.

### RayFlow

This functionality is only available when RayQC is embedded in the RaySuite framework, and therefore connected to a RayFlow server. Users may either upload a report, or trigger the update of data fields in RayFlow project, which are defined as part of the post processing routine for a particular checklist.

### Options

Opens the **Settings** view.

### Close

Closes the current checklist or project. If any changes are pending, the user will be asked to save them before continuing.

### Exit

Closes the current project and the whole RayQC application. If any changes are pending, the user will be asked to save them before continuing.

# Open an Existing Checklist

This chapter of the document describes how an existing checklist can be loaded into the RayQC **Checklist Viewer** and can further be edited using the **Checklist Editor**.

## To Open a Checklist for Execution

Follow the steps to open a checklist in RayQC:

1. A user can open a checklist by either using the **open checklist** tile from the dashboard or by selecting the **OPEN -> Open Checklist** item in the File Menu.

> **Note:**
> By using the **open checklist tile** on the dashboard, a user can only open checklist templates (`*.rqct`), whereas using the **OPEN button** from the menu bar allows either the opening of a checklist template or project (`*.rqcp`).

   When either of these options is selected, RayQC opens an explorer dialog titled **Open a checklist**.

2. Navigate to the location where the desired checklist is located.

3. Select the checklist file (`*.rqct`), and click on the **Open** button in the dialog.

   The checklist is now loaded into the RayQC **Checklist Viewer**, where it can be executed. In order to edit the underlying checklist structure, users have to switch to the Checklist Editor. This can be achieved by either clicking the Edit this checklist button from the swipe bar available at the bottom of the screen, or using the shortcut **Shift + Tab**.

The Checklist viewer is the user interface view that is active when a checklist is opened for review / execution within RayQC. Therefore, to display the Checklist Viewer:

- Use the open checklist tile from the **Dashboard** on the Home screen
- Click on one of the template items from the **Recent list** on the Home screen
- Use the Open view from the **FILE** menu and select the template file type
- Hit **Control + O** to browse the Windows system for a project file

> **Be aware:**
> Whenever a file is opened in RayQC, it is checked for structural validity. Files that contain poorly formed source structures are rejected, and cannot be displayed within the Checklist Viewer or Editor.
> When users try to open files that have originally been saved in one of the deprecated RayQC file formats, they will be rejected if they do not match the current template or project file format restrictions.

Either way, the selected file (checklist or project) is opened within the Checklist Viewer. The screenshot below shows this user interface state with one of the RayQC sample checklists opened:
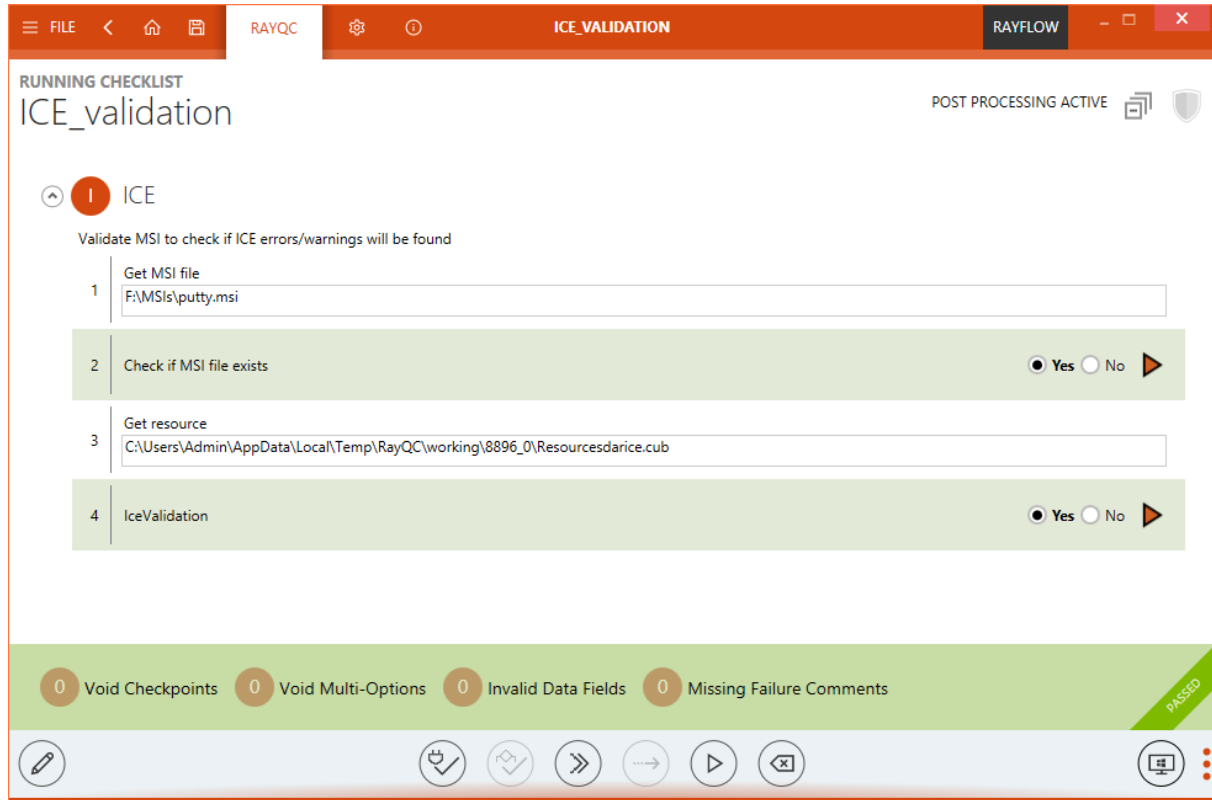


**Checklist Viewer Overview**

The application window is separated into different areas with specialized functionality and data as requested for the optimal user interaction support.

## Main Tool Bar

The menu bar has been extended with the actual checklist name of the currently viewed project on the left-hand side.

> **Be aware:**
> Any checklist opened in the Checklist Viewer is automatically transformed into a RayQC project, residing within the temporary system memory until it is saved permanently on any system location. Leaving the Checklist Viewer without saving the changes made to the actual checklist run / evaluation results does not take any effect on the underlying checklist. Saving the changes of the current checklist evaluation / run creates a RayQC project file type by default. To manipulate the checklist structure underneath the RayQC project, it must be executed within the Checklist Editor view. Please refer to the Checklist Editor section within the RayQC User Guide to get more details on how to manipulate checklist structures.

# Content Area

## Checklist Area

The actual content area begins with the listing of the checklist items in their group containers. According to the order and nesting designed within the Checklist Editor, all items that do not depend on conditional options are displayed with their type specific input controls. Users may directly enter the results of their checklist execution, call help files for further information on the checklist in general, or a specific checklist element. plug-ins can be executed and comments may be entered. All in all, the checklist area is the place where the actual end-user evaluation works with RayQC.

Checklist elements are equipped with an index value, which is unique within all items of the same parent group container. The index does not only give information about the position of the element within the item sequence of the box, but also about the indentation level of the item. The index is a multi-level indicator value, with a colon separating the different tree levels. For example, an item with the index value 2.3 is the third child of the second checklist item within a group.

The elements within a group are displayed with alternating background colors in order to support easy visual element distinction. Once Checkpoint elements have been evaluated, an additional background color markup is applied to them. If it fails, the background will turn slightly orange; if it passes, the background will turn slightly green.

Please refer to the Checklist Structures section of RayQC User Guide for details regarding the different options that may be applied towards checklist design and functionality.

## Task Bar

The task bar below the checklist elements displays a set of **status indicators**, which support users in their goal to completely evaluate the checklist elements with the least possible effort. At the left-hand side, there are four color coded **status boxes**, indicating the completeness of each checklist element group:

- Missing Entry Selections
- Missing Multi-Option Selections
- Missing User Comments
- Missing Failure Comments



**Taskbar Status Indicator**
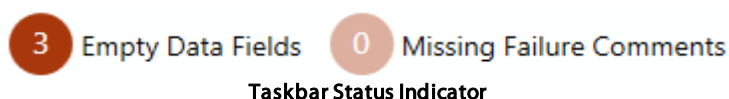
A number greater than zero in one of these items indicates that the checklist evaluation task is

not complete yet, since there is at least one more user input or activity required (See the right number presented within the illustration above). Whenever a task group is incomplete, the button is opaque, and when a task group is complete they are shown partially transparent (See the left box presented within the illustration above). The appearance is designed as a quick hint for missing information, while the number indicates the amount of open tasks that need completion before the checklist evaluation / run is complete.

Clicking on one of the task buttons with a non-zero number focuses the first incomplete checklist element according to the task group title. For example, the button for missing data fields is opaque and displays a 3 on the left side. Therefore, there are 3 checklist elements that have not been entered yet. Clicking on the button scrolls the checklist area to a scope that displays the first empty data field. As soon as the user enters a value for that element, the number of missing items displayed in the task bar button is decreased by one. Clicking the button again loads the next data field checklist element that needs user interaction into the visible scope of the checklist area.

Since checklists may become comprehensive and complex for in-depth quality assurance procedures, the task bar indicators are helpers for those situations where checklists may not be fully evaluated within one working session. They also help to keep track on conditionally displayed checklist elements. Therefore, it is recommended to use the task bar buttons after the initial run through the entire checklist and complete the tasks with their guidance.

Another visual indicator within the task bar is the **result ribbon** at the right-hand side. When a checklist is opened for execution as a project for the first time, the default ribbon state is a gray background and the label **NOT FINISHED**.



**Checklist Viewer Result Ribbon**

Hovering over the ribbon reveals a summary of checklist properties, such as the number of successful and failed checks, as well as, the total number of currently available checklist elements. This total is updated according to the actual state of element availability as derived from conditional statement examination.

**Checklist Viewer Result Ribbon with Tooltip**
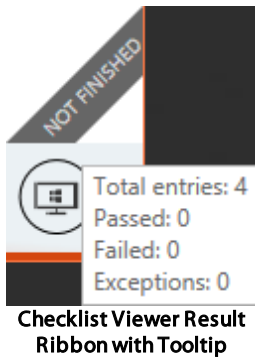
Once all checklist elements are evaluated, the result of the checklist run is available: The checklist test has either passed (indicated by a green background color and the label **PASSED** for the ribbon), or failed (indicated by an orange background color and the label **FAILED** for the ribbon).

Well, actually there are some additional circumstances and conditions that decide whether a checklist has been passed or failed, but please read about them within the Checklist Structures chapter of RayQC User Guide. For now it is just required to know that the ribbon at the right-hand side of the task bar actually indicates the result of a checklist project evaluation.

## Swipe Bar

The command bar contains controls to use in combination with the currently opened resource file:
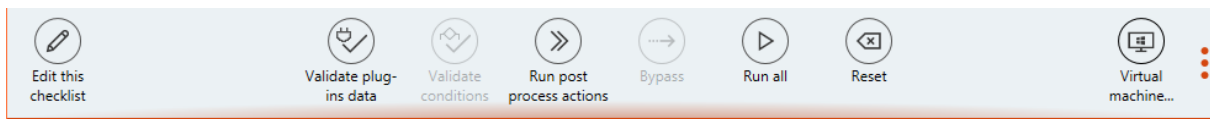


**Swipe Bar**

### Edit this Checklist

Clicking on this button opens the **Checklist Editor** with the template of the currently visible checklist already loaded for manipulation. As a handy alternative, use the swift **Shift + Tab** shortcut to switch between the Viewer and Editor mode.

### Validate Plug-ins Data

If the currently opened project contains plug-in calls, hitting this button checks whether they are logically correct or not. Possible reasons for conflicts are:

- Incorrect input parameter values or formats
- Invalid relations between elements and plug-ins (order of usage)

The result of the plug-in check is a message dialog, stating that all plug-in integrations are

flawless or that issues have occurred. If there are issues, the message dialog may be expanded to display details on the conflicts found within the plug-in definitions. In this case, it is recommended to change the mentioned plug-in parameters and recheck the checklist until all issues have been cleared.

## Validate Conditions

If the currently opened project contains conditions, hitting this button checks whether they are logically correct or not. Possible reasons for conflicts are:

- A defined condition combination will never occur, because two or more condition terms demand different results from the very same checklist item.
- Condition literals within the same clause (bucket) and clauses within the same term are defined as duplicates, e. g. one term demands result A from checklist item no 1, and the next term of the same conditional construction also demands result A from checklist item no 1.

The result of the condition check is a message dialog, stating that all conditions are flawless or that issues have occurred. If there are issues, the message dialog may be expanded to display details on the conflicts found within the conditional statements. In this case, it is recommended to change the mentioned condition terms and recheck the whole condition set of the checklist until all issues have been cleared.

## Run Post Process Actions

This button can be enabled for a checklist via the **Enable post process actions** checkbox. This checkbox is available under the **Post Processing** tab of the checklist editor. Based on the conditions defined under this tab, a set of predefined actions will be executed either by clicking upon this button or automatically when a user selects the **Run All** button. Furthermore, post process action of automatically creating and uploading a report to RayFlow can also be initiated via the command line switch.

## Bypass

The bypass option is available if the checklist is configured to allow manual result bypassing. For those projects that base on checklists with bypass permission, the bypass button becomes available once the checklist elements have been fully evaluated and a result (**PASSED** or **FAILED**) is displayed in the ribbon at the right-hand side of the task bar.

Hitting the **Bypass** button displays the **BYPASS** dialog. Within this dialog, the user should write a note why the bypass was required. Setting the result bypass state to **Bypass** (by clicking on the radio control item option **Bypass**) reverts to the original checklist project result (e. g. from original result **FAILED** to the new, bypassed result **PASSED**).

Clicking the **OK** button within the **BYPASS** dialog saves the new result settings and closes the **BYPASS** dialog.
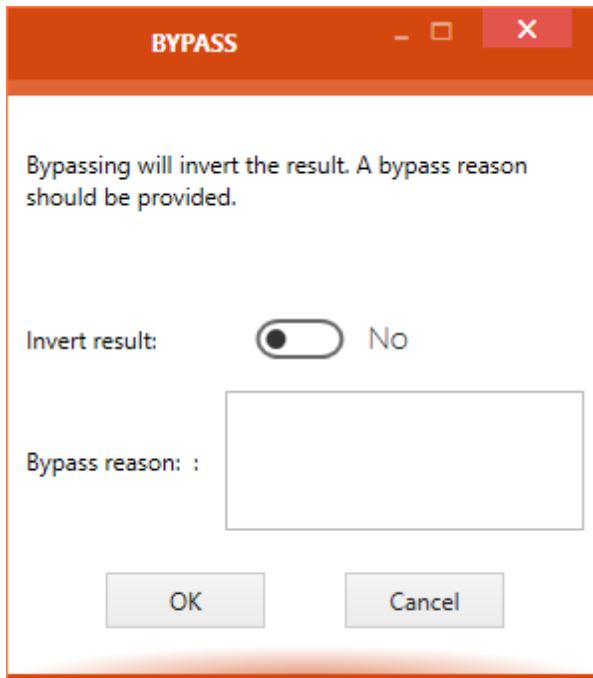
**ByPass Control**

It is possible to revert the bypass, which restores the original checklist evaluation result state again. To do so, users call the **BYPASS** dialog and move the Invert result toggle slide to **YES**. However, conditions regarding required circumstances for bypassing and bypass revocation have to be defined by the creator of the original checklist template used within the current project instance.

## Run All

If a project contains checklist items with plug-in usage, hitting the **Run All** button automatically executes all plug-ins at once. All plug-ins are run in turn, beginning from the one that has the highest position within the checklist tree. The execution is kept procedural, which means that plug-in B will start when plug-in A has finished. plug-in A and B will not be executed in paralle. Therefore, plug-ins that require input based on earlier plug-in execution results will always rely on current results when the Run All function is used.

If the Run All button is used after one or more plug-ins have been run (manually or automatically), RayQC displays a dialog asking the user if former results should be overridden by the new execution or if the Run All execution should be aborted instead.

**Be aware:**
On Run All, element's visibility will be evaluated after each plug-in execution and the checklist will be processed correctly. Only elements that requires a manual input will not be processed, and their condition will not be fulfilled by any plug-in.

## Reset

Using the **Reset** button clears all checklist element results that are part of the  respective

checklist in the Checklist Viewer. If a checklist has been opened in a project scope including a former evaluation state, resetting does not reset to that state, but to the default initial state as given from the checklist element definitions.

**Be aware:**
Reset does not reset the original checklist element settings of a checklist under construction. It simply removes the result information entered within the Checklist Viewer mode. Adjustments made towards the checklist elements within the Checklist Editor are kept unchanged.

## Show / Hide Swipe Bar Labels

The swipe bar comes in two display modes: expanded and minimal. The expanded mode displays a label for each button of the task bar, whereas the minimal mode contains only icons users may click on. Hitting the button with the three vertical dots at the right-hand side of the task bar switches between the expanded and minimal task bar display modes.

# Create New Checklist

The Checklist Editor is the RayQC interface for the manipulation of checklist structures and settings. To display a checklist template within the Editor environment:

- Use the open checklist tile from the **Dashboard** on the Home screen
- Click on one of the template items from the **Recent list** on the Home screen
- Use the Open view from the **FILE** menu and select the template file type
- Hit **Control** + **Shift** + **O** to browse the Windows system for a template file

Either way, the selected file (checklist or project) is opened within the **Checklist Viewer**. To switch to the Editor interface, use the **EDIT** button available from the swipe bar at the bottom of the application window area.

The screenshot below shows one of the RayQC checklist template examples ready for manipulation within the Checklist Editor interface:
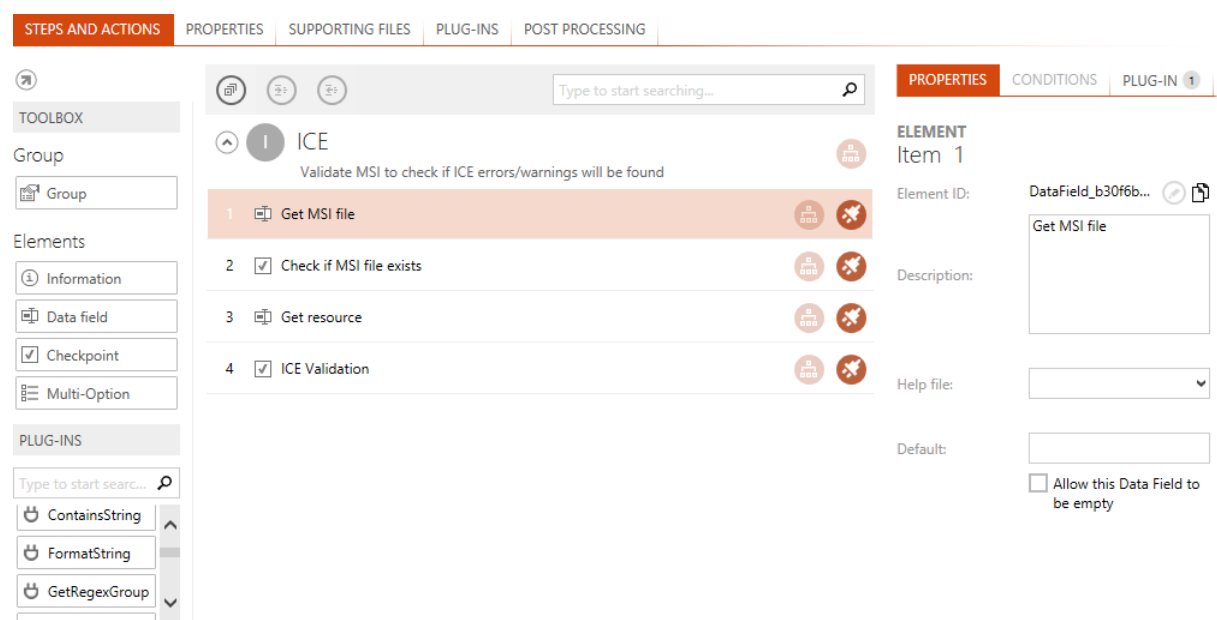


**Checklist Editor Overview**

The application window is separated into different areas with specialized functionality and data as required for optimal user interaction support.

## Main Tool Bar

The menu bar has been extended with the actual view name of the editor on the left-hand side. It

also provides direct access to the **FILE** menu that has options to trigger standard procedures such as saving the current checklist or creating a new one.

## Content Area

### Checklist Title

The title of the currently opened checklist may be edited by clicking on the edit button at its right-hand side. A direct value editor dialog is displayed and ready to accept the new title. As an alternative, the title may as well be modified from the Properties tab of the checklist editor interface.

### Checklist Canvas

All other properties of a checklist are available for manipulation via the tabbed views contained within the checklist canvas. Please refer to the Checklist Structures section of RayQC User Guide for details regarding the different options that may be applied towards checklist design and functionality.

## Swipe Bar

### View this Checklist

Clicking on this button opens the Checklist Viewer with the project representation of the currently visible checklist already loaded for testing and evaluation purposes. As a handy alternative, use the swift **Shift + Tab** shortcut to switch between the Viewer and Editor mode.

### Show / Hide Swipe Bar Labels

The task bar comes in two display modes, expanded and minimal. The expanded mode displays a label for each button of the task bar, whereas the minimal mode contains only icons users may click on. Hitting the button with the three vertical dots at the right-hand side of the task bar switches between the expanded and minimal task bar display modes.
The **Checklist Editor** is primarily divided into five areas: **Steps and Actions**, **Properties**, **Supporting Files**, **plug-ins** and **Post Processing**..

### Steps and Actions

The **Steps and Actions** tab generally provide a working platform to add elements from the Toolbox and edit them for the functionality / checks they are supposed to offer.

### Properties

The **properties** tab groups general checklist settings into one view. Properties defined here take

effect on any project file saved from the template.

## Supporting Files

Sometimes it is necessary to enrich checklist templates with more information than can be easily handled by simple textual descriptions for groups and elements. Supporting files are a decent way for checklist editors to add PDF or RTF documents as help content and PNG images as illustrations.
While each checklist element may be equipped with a specific help file, images may be used freely within checklist, group and element descriptions by using the markup options for these properties.

The benefit of organizing supporting files in a separate dialog is reusability. Once a supporting file has been added to the checklist, it may be utilized as often as the checklist editor sees fit. This is especially convenient in cases where the same supporting file has to be provided for several elements or groups with conditional availability during the actual checklist evaluation run. The file resources are stored once (directly within the checklist container) and referenced as often as required. By providing a freely usable pool of supporting files, it is possible to keep the overall file size of RayQC checklist containers at a necessary minimum.

## Plug-ins

The **plug-ins** tab allows its users to add local PowerShell / DLL plug-ins to the selected checklist.

## Post Processing

It is quite likely that RayQC will be used as a tool that is integrated into RayFlow. This means that evaluations are commonly triggered from a RayFlow server. In order to provide bidirectional communication, there is not only a way to get data from RayFlow into RayQC, but also to return information (report files and updated values for RayFlow data fields) back to the RayFlow server. In order to standardize and automate communication as best as possible, users should define certain post processing tasks. If post processing itself is activated for a checklist, RayQC checks for condition fulfillment and executes the post processing actions if one of the active conditions is met.

Post processing may either be triggered manually by using the post processing button from the Swipe bar of the Checklist Viewer or automatically as extension of a "Run All" procedure execution. The latter option is the required one for fully automated checklist evaluation.: RayFlow triggers the checklist run including the automation parameter, and RayQC automatically responds with the information defined within the post processing section.
However, it is also useful for users to be able to upload checklist evaluation results to RayFlow themselves, since not all checklists may be fully automated. This is where the button for post processing execution kicks in.

> **Be aware:**
> Post processing can only operate successfully, if the current evaluation session has a valid RayFlow connection. If no parameter injection and no RayFlow connection profile is given, there is no valid target for the data RayQC will send. The result is an error message, which will be displayed if post processing fails due to missing connectivity.

# Toolbox

The **toolbox** at the left contains items that may be added to the checklist structure: groups, elements, and plug-ins.

The toolbox allows adding new objects to the current checklist item flow. By simply dragging a group or element to the area with the already defined elements and dropping it at the desired target position, users add a default object that is ready for adjustments.

Use the arrow icons in the upper left corner of the toolbox to undock or dock the toolbox from the Checklist Editor. The undocked toolbox allows users that operate on monitors with small resolutions to organize their Checklist Editor interface according to their individual space requirements. Adjusting the height and the width of the undocked toolbox is possible by using the standard application window resizing functionalities provided by the underlying operating system.

The Element Menu lists the checklist elements that can be dragged and dropped to the logical view of the Checklist Editor.

To add an element to the current checklist structure, simply drag the type icon from the toolbox on the right to the target spot within the checklist at the left. The new item will be inserted exactly where it has been dropped before. The position (and indentation) of each checklist item may be changed later, but dropping it at the desired target position makes life easier right from the start.

### Group

The group element is the parent element of any test sequence. Each checklist has to contain at least one group to be valid. Groups are identified by their **title** and **description** property and contain an arbitrary combination of checklist elements (Information, Data Field, checkpoint or multi-option items).  A group must contain at least one element to be valid.
Groups may be equipped with conditions, which allows to checklist authors to dynamically include or exclude whole groups from a checklist sequence according to the result values of specific elements.

The arrow icon in the upper left corner of the group box allows to collapse and expand the group area. This is especially helpful when dealing with extended checklists, as it may become necessary to hide the content of those groups, that are momentarily irrelevant.

### Information

The Information is the simplest elements as it merely displays information. As any other checklist element, it may also provide a link to an external help file (PDF or RTF) and trigger plug-in execution. It can be displayed and hidden according to conditional statements.

### Checkpoint

The checkpoint is the standard element of a checklist. It is the only element type that has direct influence on the overall result of a checklist. A checkpoint

may either state a successful test step result or a failed one. Although it is possible to configure them to accept exceptions, this ability to clearly state the test result is of highest importance for automated checklist evaluation runs.

**Data Field**
The Data Field takes textual test results. These may either be entered manually by an evaluator, or automatically by a plug-in execution return value. A checklist cannot be finished until all mandatory Data Fields are filled with values. Even though, Data Field cannot influence the final checklist result, they can prevent an evaluation run from finishing.

**Multi-Option**
There may be situations in which a simple Yes-No logic is insufficient to make decisions. For such situations, there is the multi-option element that is able to have any number of result options. These items are commonly used to create checklist flow switches: Depending on the result of a specific test, there may be changes towards the visibility of checklist groups and / or items. Therefore, multi-option items make checklists very flexible and dynamic.

## Swipe Bar

The **Swipe Bar** is available at the bottom of the **Checklist Editor**. It includes functional options required to perform various actions including saving, exporting, execution and reset on your checklist. For further description of the options, please refer to the **Checklist Viewer** section of the **Open an Existing Checklist** chapter.

## To Create a New Checklist

This section describes how to create and extend a new checklist using the RayQC Checklist Editor.

Follow the steps described below to create a sample checklist:

1.  From the RayQC **Dashboard** select the **create checklist** tile to open the **Checklist Editor**.

    An explorer titled **Create checklist** is opened.

2.  Navigate to the desired storage location for the checklist; provide a name for the checklist file in the **File name** data field and click on **Save**.

    The checklist is now opened in the Checklist Viewer. A **Group** element has already been added to it, along with a **Comment** element ready for manipulation.

**Note:**
A checklist must contain at least one group, and a group has to contain at least one element. Therefore, RayQC prevents the deletion of the last object within a group or checklist.

3.  Click on the Edit this checklist from the swipe bar, which is available at the bottom of the Checklist Viewer

4. Click on the **Properties** tab of the checklist editor

5. Provide a title and description for the checklist in the **Checklist title** and **Checklist description** data fields respectively.

   The Tile bar is automatically updated with the new checklist tile.

6. If required, activate the **Allow Bypassing this list** checkbox to allow bypass on your checklist result, and further provide a valid bypass reason in the **Bypass message** data field.

7. Click on the **Steps and Actions** tab to further edit this checklist

8. Select the existing Group element and then edit its **Group title** and **Description**. The provided group title is updated in the group header.

9. Within the default group and below the existing comment element, in turn drag and drop a **Data Field** and **Checkpoint** element.

10. Click on the **View this checklist** button from the swipe bar available at the bottom of the checklist editor. Alternatively users can also use **Shift + Tab** shortcut to switch between the checklist viewer and checklist editor.

    The checklist is now loaded into the Checklist Viewer.

11. To save the created checklist, select the **Save** button from the **File** menu. To save the checklist state as checklist project file, select from File Menu **Save as** -> **Save as Project**

## Element Control Options

**Indents**

The indents tool within an element can be used to provide indentations to an element so that it is displayed offset.

**Text Formatting**

There are several options for formatting elements and descriptions, for example bold text and line breaks.

Text markers can be used for:

- The description text for all elements

- The group description

- The checklist description

To mark the text bold, simply enclose it in the [bold] tag. E.g. [bold]Text[/bold]

To use line breaks, use the [br] tag. E.g. This is a sample text.[br]Continue from the next line.

There are even more formatting options - please refer to the User Guide for details!

**Help**

The info text of an element is sometimes insufficient in describing all details of the element. For this reason, a guide or supplemental text in the form of a help file can be linked to the element. At present, RayQC allows the addition of RTF and PDF files as help files or a web link to the web page containing related information.

To Add a Help File to a Checklist Element

1. Open the checklist in the checklist editor

2. Click on the **Supporting Files** tab and then click on the **Add file...** button, which is available under the **Help files** option
   File explorer is opened

3. Navigate to the help file location

4. Select the file type (`.rtf` or `.pdf`) of your help file from the explorer drop-down menu. Select the file and click on the Open button

   The file is now added under the **Help files**

5. To add this help file to a checklist element, switch to the **Steps and Actions** tab

6. Select the desired element to which the help file is to be added
   This will activate the Properties tab in the details pane on the left

7. Select the file from the drop-down, which is available next to the **Help file** property

8. The help file icon is activated and is now present next to the element in the Checklist Viewer
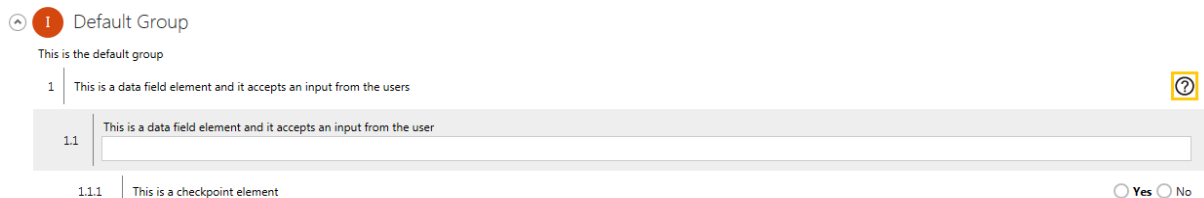


## Configuring Data Field

- A **Allow this Data Field to be empty** flag can be set for Data Field element. When it is set, entering the user comment is optional during checklist evaluation. This flag is available under the properties tab of the Details Pane on the left.

- The Default field is available under the element properties tab of the details pane. When a user sets a value for this field, this value is shown by default in the data field's element of the checklist in checklist viewer.

## Configuring Checkpoint

The multi-option and checkpoint elements can be configured for **Expected Value value**, **Allow exceptions** and as **Evaluate this element.**

## Expected Value

In order not to have to rephrase each checkpoint so that the **Yes** response matches a positive value, the **switchValue** attribute can be activated. As a result, the checkpoint is evaluated as passed when the **No** response was selected.

## Allow Exceptions

It may sometimes be necessary to configure a checkpoint so that even if it cannot be evaluated as Passed, the evaluation of the overall checklist will not be affected. This can be achieved with the **allowException** attribute.

## Evaluate this element

Set the **dontEvaluate** attribute to create a checkpoint that is not included in the evaluation of the checklist.

## Including plug-ins

Select the **plug-ins** attribute to use a plug-in in the respective element. By default, RayQC provides nine native plug-ins:

- **Logic**

- **File**

- **Folder**

- **Registry**

- **LocalSystem**

- **Msi**

- **Command**

- **RayFlow**

- **IniFile**

Functions available under each plug-in can be dragged and dropped from the toolbox over the element(It is not possible to add plug-in to an Information element) in the checklist editor. This will add an entry under the plug-in tab of details pane on the left

Apart from the native plug-ins, RayQC offers its users the possibility to include external PowerShell and DLL plug-ins. External plug-ins can be categorized into local and global plug-in. A local plug-in is native to the selected checklist and a global plug-in is available for use in all the checklists.

> **Note:**
> For detailed information on functions offered by native plug-ins and how to use plug-ins in a checklist, please refer to the RayQC User Guide.

# Open a Checklist Project

A checklist project file (*.rqcp) is a ZIP container which contains three files: the original checklist file (.xml), state file (.xml) and postprocess file (.xml). The state file contains the intermediate changes made to the checklists evaluation result. When a project file is opened, RayQC loads the checklist file in the Checklist Viewer and applies the state to it. The postprocess.xml file contains the post-processing configuration for the checklist.

To open a checklist project file:

Follow these steps to open a checklist project file in RayQC:

1. A user can open a checklist by either using the **open project** tile from the **Dashboard** or by selecting the **OPEN -> Open Project** item in the **File Menu**.

   When either of these options is selected, RayQC opens an explorer dialog titled **Open project.**

2. Navigate to the location where the desired checklist is stored.

3. Select the checklist file (.rqcp) and then click on the **Open** button in the dialog.

The checklist project is now loaded into the RayQC Checklist Viewer, where it can either be executed or can be loaded into edit mode for further manipulation and extension.

# Settings

The **Settings** area of RayQC is available from the **Settings** tile on the **Dashboard**. Additionally, it is available from the **Settings** tab of the Main Tool Bar, which is visible in all views of the application. It is recommended to check the settings at least once before the productive work with RayQC begins, since the configuration options offered there determine some properties of the user interface that may help achieve a quick orientation.
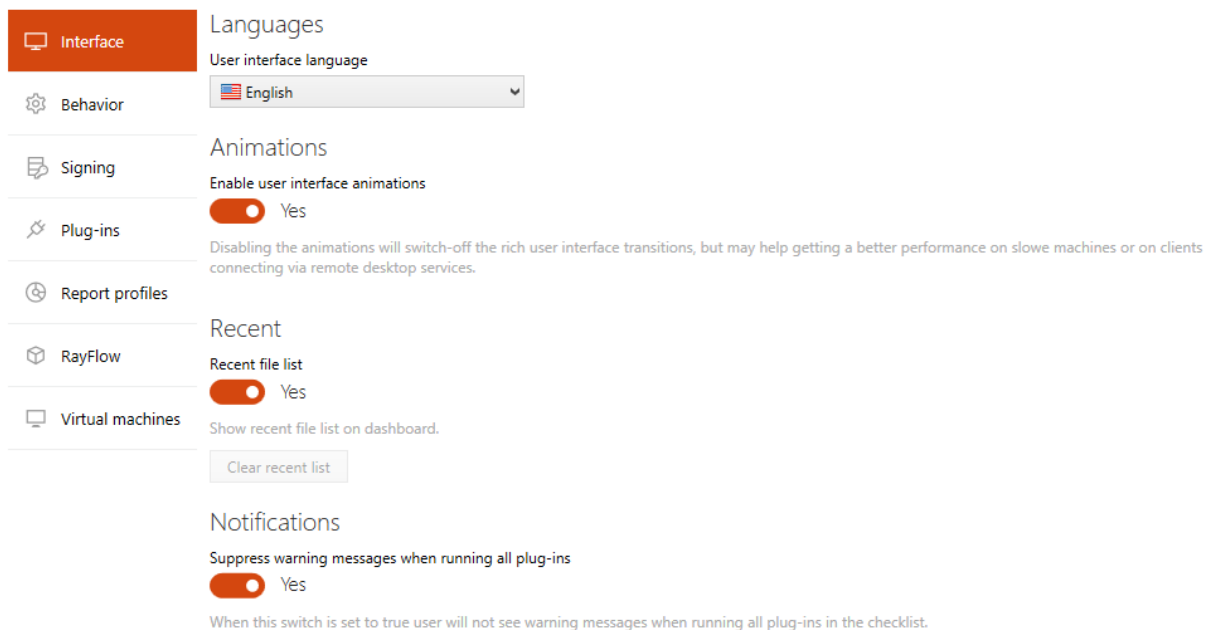


**RayQC Settings Screen**

## Available Settings Options

Within the **Settings** area, there are seven views with configuration groups:

- **Interface**

    Options for user interface settings regarding the RayQC application. Following options are available under this tab:
    - **Languages:** This drop-down menu allows the user to change the language of the GUI. Available languages are English and German. The changes will be applied after a restart of the application.
    - **Animations:** This switch allows its users to enable or disable UI animation
    - **Recent file list:** Enable or disable showing of recent file list on the RayQC dashboard

- **Behavior**

    Options for the settings regarding the behavior of the  RayQC application. Following options are available under this tab:

- **Write protect plug-in results:** Enable or disable plug-in controlled elements to be write protected
- **Reject checklists without signature:** When enabled, loading of checklist or a project is rejected if it is not signed
- **Reject untrusted certificates:** When enabled. loading of a signed checklist is rejected, if the related certificate is not trusted
- **Sign Checklist on saving:** To enable this option, a user will need to have a valid key/ certificate pair

- **Signing**

  Definition of options and upload of resources for checklist signing.

- **Plug-ins**

  Adding or removing global PowerShell and DLL plug-ins. This tab is disabled when a checklist is currently loaded into the RayQC.

- **Report Profiles**

  Definition of options and upload of resources for report generation. A user can define different report profiles, which is in turn used when publishing a report or uploading it back into RayFlow.

- **RayFlow**

  These settings are only required when your instance of RayQC is used in combination with RayFlow, Raynet's workflow management system for packaging processes. If RayFlow or any of the other solutions from the RaySuite product family are not being utilized in your IT infrastructure, please visit the Raynet website, or contact your Raynet sales representative for further information on how the RaySuite can improve your packaging related business processes.
- 
- **Virtual machines**

  These settings are used to configure virtual machines for usage with RayQC.

---

| | **Note:** |
|---|---|
| | For further information on the options available under the Settings page, please refer to the RayQC 7.0 User Guide |

---

# Need Help?

## Request RayQC Support

Our Raynet support team gladly assists you on any question or issue you encounter regarding RayQC. Feel free to sign in and open incidents via our Support Panel.

## Contact your Raynet Sales Representative

Our sales team is the right contact for any license or edition question you might encounter. Would you like to benefit from a professional RayQC training? Ask for dates and locations to find the right training to meet your needs. You are highly welcome to step in contact via sales@raynet.de.