



RT-LAB

Installation Guide

RT-LAB

Version 2020.2

Installation Guide

Q040 419-01



OPAL-RT
TECHNOLOGIES

© 2020 Opal-RT Technologies Inc. All rights reserved for all countries.

Information in this document is subject to change without notice and does not represent a commitment on the part of OPAL-RT Technologies. The software and associated files described in this document are furnished under a license agreement and can only be used or copied in accordance with the terms of the agreement. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information and retrieval systems, for any purpose other than the purchaser's personal use, without express written permission of OPAL-RT Technologies Incorporated.

Documents and information relating to or associated with OPAL-RT products, business, or activities, including but not limited to financial information; data or statements; trade secrets; product research and development; existing and future product designs and performance specifications; marketing plans or techniques, client lists, computer programs, processes, and know-how that have been clearly identified and properly marked by OPAL-RT as "proprietary information," trade secrets, or company confidential information. The information must have been developed by OPAL-RT and is not made available to the public without the express consent of OPAL-RT or its legal counsel.

ARTEMIS, RT-EVENTS, RT-LAB and DINAMO are trademarks of OPAL-RT Technologies, Inc. MATLAB, Simulink, Simulink Coder and Simscape Power Systems are trademarks of The Mathworks, Inc. LabVIEW is a trademark of National Instruments, Inc. All other brand and product names are trademarks or service marks of their respective holders and are hereby acknowledged.

We have done our best to ensure that the material found in this publication is both useful and accurate. However, please be aware that errors may exist in this publication, and that neither the authors nor OPAL-RT Technologies make any guarantees concerning the accuracy of the information found here or in the use to which it may be put.

Published in Canada

Contact Us

For additional information you may contact the Customer Support team at OPAL-RT
<https://www.opal-rt.com/contact-technical-support/>

Mail 1751 Richardson Street
Suite 2525
Montreal, Quebec
H3K 1G6

www.opal-rt.com

Toll-Free (US and Canada) 1-877-935-2323 (08:30-17:30 EST)

Phone 514-935-2323

Fax 514-935-4994

CHAPTER 1: INTRODUCTION

What to Expect from this Guide	5
About RT-LAB	5
Key Features	5
Intended Audience and Required Skills and Knowledge	6
Organization of Guide.....	6
Conventions	6

CHAPTER 2: BASIC CONCEPTS

Definitions	7
-------------------	---

CHAPTER 3: REQUIREMENTS

Hardware Requirements.....	9
Software Requirements.....	9
OPAL-RT Chassis Compatibility.....	12

CHAPTER 4: INSTALLING RT-LAB ON WINDOWS COMMAND STATIONS AND WINDOWS TARGET NODES

Installing MATLAB™ and Simulink toolboxes.....	15
Installing RT-LAB Software on your Host Station	15
Configuring your firewall application.....	17
Starting RT-LAB	17
Validating the installation	19
Installing the LabVIEW API	20
Installing TestStand	20
Configuring ScopeView with EMTP-RT.....	20
Setting Up the Windows PC as a Target Node.....	20

CHAPTER 5: CONNECTING YOUR REAL-TIME TARGET NODE TO YOUR NETWORK

Direct connection.....	23
Local area network configuration.....	25
Discovering target nodes	26

TABLE of CONTENTS

CHAPTER 6: INSTALLING RT-LAB ON AN OPAL-RT SIMULATOR

Installing RT-LAB on the Target Node	29
Uninstalling RT-LAB	31

CHAPTER 7: ACTIVATING A LICENSE ON THE TARGET NODE

Step 1: Request a License.....	33
Step 2: Install the license	35

Introduction

1.1 What to Expect from this Guide

You can expect to find information on how to install RT-LAB, set up a target machine as well as information on licensing options.

1.2 About RT-LAB

RT-LAB is the real-time technology that is revolutionizing the way model-based design is performed. Through its openness, it has the flexibility to be applied to any simulation and control problem, and its scalability provides a low-risk entry point for any application, allowing the developer to add compute-power where and when needed - whether it is to speed up simulations or for real-time hardware in the loop applications. RT-LAB provides tools for running and monitoring your simulations or controls on various runtime targets. An open architecture enables RT-LAB to work with the popular diagramming tool MATLAB/Simulink.

1.3 Key Features

Real-Time Model Development

- Fully integrated with MATLAB/Simulink/Stateflow/SimPowerSystems/Simulink Coder
- Specialized Blockset to prepare the model for distributed processing, inter-node communication and signal I/O
- "Six Clicks to Real-Time" User interface
- Comprehensive API for developing your own online application, including tools for LabVIEW, C++, MATLAB, Python and 3D Virtual Reality Tools

Run-time Features

- Distributed, synchronized execution across multiple processors, including, multiple CPU motherboards via Shared Memory and Dolphin Interconnect communication
- Integrated Signal Visualization and Control Panel
- Dynamic Signal Tracing - select any signal for monitoring during run-time
- On-line Parameter Editor - change parameters on-the-fly (manually or load parameter files)
- Extensive I/O card support - over 100 devices supported
- Real-time operating systems : OPAL-RT Linux (x86-based) or Petalinux (ARMv7-based)

Performance Features

- XHP Mode - high speed mode to give closed loop cycle times below 10 microseconds
- Optimized Hard-Real-Time Scheduler - high performance, low jitter

1.4 Intended Audience and Required Skills and Knowledge

The RT-LAB Installation Guide is intended for those who will be called upon to install and manage RT-LAB.

1.5 Organization of Guide

There are several guides offered in the list of RT-LAB documentation:

- Installation Guide
- User Guide
- C-API Reference Guide
- Python API Reference Guide

Here we take a look at the Installation Guide. The topics covered are:

Introduction - This section provides basic information such as what you should expect from the guide, the intended audience, the chapter breakdown and conventions.

Basic Concepts - This section describes the different configurations that can be applied to your RT-LAB system.

Requirements - This section describes hardware and software requirements.

Installing RT-LAB on Windows Command Stations and Windows Target Nodes - This section describes the installation of RT-LAB host and target components on a Windows machine.

Connecting your Real-Time Target Node to your Network - This section explains how to connect your OPAL-RT chassis to your Windows command station.

Installing RT-LAB on an OPAL-RT Simulator - This section describes the installation of RT-LAB target components on an OPAL-RT chassis.

Activating a License on the Target Node - This section explains how to request and install a license on a target node.

1.6 Conventions

OPAL-RT guides use the following conventions:

Table 1: General and Typographical Conventions

THIS CONVENTION	INDICATES
Bold	User interface elements, text that must be typed exactly as shown.
Note:	Emphasizes or supplements parts of the text. You can disregard the information in a note and still complete a task.
Warning:	Describes an action that must be avoided or followed to obtain desired results.
Recommendation:	Describes an action that you may or may not follow and still complete a task.
Code	Sample code.
<i>Italics</i>	Reference work titles.
Blue Text	Cross-references (internal or external) or hypertext links.

Basic Concepts

Before you begin with the installation, it is important to determine the configuration you wish to use. This section describes the different configurations that can be applied to your RT-LAB system.

RT-LAB makes a distinction between the computer or computers used for development, and the computer or computers used to run the simulation, although the same computer may be used for both development and simulation.

2.1 Definitions

2.1.1 Command Station

Development is always done on Windows. The computer used to run the **RT-LAB** user interface, design software including Simulink, and/or monitoring software such as LabVIEW, is known as the **Command Station**. It is possible to include more than one **Command Station** in an RT-LAB system.

2.1.2 Target Nodes

The computers used to run your simulation are called **target nodes**, **targets**, or **simulators**. Real-time **target nodes** can run using OPAL-RT Linux (x86-based) or Petalinux (ARMv7-based) real-time operating systems. The Windows command station can be used simultaneously as a **target node** for slow pseudo real-time purposes. For super-computer performance, RT-LAB can be set up using a network of multi-processor computers.

Requirements

3.1 Hardware Requirements

Command station computer(Windows):
 Any Intel or AMD x86-64 processor
 3,8 GB of disk space
 4 GB of RAM (including use of MATLAB/Simulink)

3.2 Software Requirements

3.2.1 Legend

- **YES/Y:** Supported.
- **-:** Not supported.
- *****: Partial support (read notes below table).

3.2.2 Microsoft Windows Compatibility

Table 2: RT-LAB Versions vs. Windows Versions

RT-LAB version	Vista	7	8	10	Comment
RT-LAB 11.0.x	YES	YES	YES	-	
RT-LAB 11.1.x	YES	YES	YES	Partial support of Windows 10 for RT-LAB 11.1.	
RT-LAB 11.2.x	YES	YES	YES	YES	
RT-LAB 11.3.x	YES	YES	YES	YES	
RT-LAB 2019.1	YES	YES	YES	YES	
RT-LAB 2019.2	YES	YES	YES	YES	
RT-LAB 2019.3	YES	YES	YES	YES	Windows 32 bits are no longer supported
RT-LAB 2020.1	YES	YES	YES	YES	
RT-LAB 2020.2	YES	YES	YES	YES	

*: Partial support of Windows 10 for RT-LAB 11.1.

3.2.3 MATLAB Compatibility

Table 3: RT-LAB Versions vs. MATLAB Versions

	R2011 B	R2012 A	R2012 B	R2013 A	R2013 B	R2014 A	R2014 B	R2015 A SP1	R2015B	R2016 A	R2016 B	R2017A	R2017B	R2018A	R2018 B	R2019 A	R2019B
RT-LAB 11.0.x	Y	Y	Y	Y	Y	-	Y	Y	-	-	-	-	-	-	-	-	-
RT-LAB 11.1.x	Y	Y	Y	Y	Y	-	Y	Y	-	-	-	-	-	-	-	-	-
RT-LAB 11.2.x	Y	Y	Y	Y	Y	-	Y	Y	11.2.1	11.2.1	11.2.1	11.2.2	-	-	-	-	-
RT-LAB 11.3.x	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	11.3.4	-	-	-	-
RT-LAB 2019.1	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-
RT-LAB 2019.2	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-
RT-LAB 2019.3	Y	*	*	*	*	-	*	*	Y	Y	Y	Y	Y	Y	Y	-	-
RT-LAB 2020.1	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y
RT-LAB 2020.2	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y

WARNING: Support of MATLAB R2011b is deprecated. It will be abandoned in RT-LAB 2019.2.

Since RT-LAB 11.1.4, **64-bit versions of MATLAB are fully supported** (including the Simulink console).

Note: ePHASORSim (included in the installation of RT-LAB) does not yet support 64-bit versions of MATLAB.

Required Elements:

- **MATLAB:** Underlying mathematical engine
- **Simulink:** Graphical design environment
- **Simulink Coder** (formerly known as Real-Time Workshop): Code generator

3.2.4 National Instruments LabVIEW Compatibility

LabVIEW is not required for RT-LAB to work. However, RT-LAB's API for LabVIEW can be used to create custom user applications.

Table 4: RT-LAB Versions vs. LabVIEW Versions

	2012	2013	2014	2015	2016	2017
RT-LAB 11.0.x	Y	Y	Y	-	-	-
RT-LAB 11.1.x	-	Y	Y	Y	Y	-
RT-LAB 11.2.x	-	Y	Y	Y	Y	Y
RT-LAB 11.3.x	-	Y	Y	Y	Y	Y
RT-LAB 2019.1	-	Y	Y	Y	Y	Y
RT-LAB 2019.2	-	Y	Y	Y	Y	Y
RT-LAB 2019.3	-	Y	Y	Y	Y	Y
RT-LAB 2020.1	-	Y	Y	Y	Y	Y
RT-LAB 2020.2	-	Y	Y	Y	Y	Y

3.2.5 National Instruments TestStand Compatibility

TestStand is not required for RT-LAB to work. However, it can be used to create custom test sequences for various quality assurance duties.

3.2.6 Microsoft Visual C++ compatibility

Microsoft Visual C++ is required in order to use the Windows command station as a target node.

RT-LAB 2019.1 is only compatible with **MSVC++ 2010**.

3.2.7 Real-Time Operating System Compatibility

RT-LAB targets run on OPAL-RT Linux (x86-based) and Petalinux (ARMv7-based) real-time operating systems. Petalinux is only used for targets with an ARM processor, meaning the OP4200 chassis.

Table 5: RT-LAB Versions vs. Real-Time Operating Systems

	OPAL-RT LINUX (X86-BASED)	PETALINUX (ARMV7- BASED)
RT-LAB 11.0.x	Y	-
RT-LAB 11.1.x	Y	Y
RT-LAB 11.2.x	Y	Y
RT-LAB 11.3.x	Y	Y
RT-LAB 2019.1	Y	Y
RT-LAB 2019.2	Y	Y
RT-LAB 2019.3	Y	Y
RT-LAB 2020.1	Y	Y
RT-LAB 2020.2	Y	Y

RT-LAB 2019.1 no longer supports targets running on **QNX** real-time operating systems.

3.2.8 Python compatibility

RT-LAB 11 is compatible with all Python versions between **Python 2.2** and **Python 2.7**.

	Python 2.2	Python 2.3	Python 2.4	Python 2.5	Python 2.6	Python 2.7	Python 3.7
RT-LAB 11.0.x	Y	Y	Y	Y	Y	-	-
RT-LAB 11.1.x	Y	Y	Y	Y	Y	-	-
RT-LAB 11.2.x	Y	Y	Y	Y	Y	-	-
RT-LAB 11.3.x	Y	Y	Y	Y	Y	-	-
RT-LAB 2019.1	Y	Y	Y	Y	Y	-	-
RT-LAB 2019.2	Y	Y	Y	Y	Y	-	-
RT-LAB 2019.3	-	-	-	-	Y	Y	Y
RT-LAB 2020.1	-	-	-	-	Y	Y	Y
RT-LAB 2020.2	-	-	-	-	Y	Y	Y

WARNING: Support of **Python 2.2**, **Python 2.3**, **Python 2.4**, and **Python 2.5** is deprecated. It will be abandoned in RT-LAB 2019.2.

3.2.9 RT-EVENTS Compatibility

RT-EVENTS is a block set of Simulink developed by OPAL-RT. It may be used in real-time when compiled with RT-LAB. Here is a compatibility table with RT-LAB, but it is always recommended to use the RT-EVENTS version that is packaged with RT-LAB on the website download or on the installation disc.

Table 6: RT-LAB Versions vs. RT-EVENTS Versions

	RT-EVENTS 4.0.X	RT-EVENTS 4.1.X	RT-EVENTS 4.2.X	RT-EVENTS 4.3.X	RT-EVENTS 4.4.X	RT-EVENTS 4.5.X	RT-EVENTS 4.6.0	RT-EVENTS 4.6.1
RT-LAB 11.0.x	Y	-	-	-	-	-	-	-
RT-LAB 11.1.x	-	Y	-	-	-	-	-	-
RT-LAB 11.2.x	-	-	Y	-	-	-	-	-
RT-LAB 11.3.x	-	-	-	Y	-	-	-	-
RT-LAB 2019.1	-	-	-	-	Y	-	-	-
RT-LAB 2019.2	-	-	-	-	Y	-	-	-
RT-LAB 2019.3	-	-	-	-	-	Y	-	-
RT-LAB 2020.1	-	-	-	-	-	-	Y	-
RT-LAB 2020.2	-	-	-	-	-	-	-	Y

3.2.10 ARTEMIS Compatibility

ARTEMIS is a Simulink blockset developed by OPAL-RT. It may be used in real-time when compiled with RT-LAB. Here is a compatibility table with RT-LAB, but it is always recommended to use the ARTEMIS version that is packaged with RT-LAB on the website download or on the installation disc.

	ARTEMIS 7.0.X	ARTEMIS 7.1.X	ARTEMIS 7.2.X	ARTEMIS 7.3.X	ARTEMIS 7.4.X	ARTEMIS 7.5.X	ARTEMIS 7.6.0	ARTEMIS 7.6.1
RT-LAB 11.0.x	Y	-	-	-	-	-	-	-
RT-LAB 11.1.x	-	Y	-	-	-	-	-	-
RT-LAB 11.2.x	-	-	Y	-	-	-	-	-
RT-LAB 11.3.x	-	-	-	Y	-	-	-	-
RT-LAB 2019.1	-	-	-	-	Y	-	-	-
RT-LAB 2019.2	-	-	-	-	Y	-	-	-
RT-LAB 2019.3	-	-	-	-	-	Y	-	-
RT-LAB 2020.1	-	-	-	-	-	-	Y	-
RT-LAB 2020.2	-	-	-	-	-	-	-	Y

3.2.11 TestDrive Software Compatibility

TestDrive is a software developed by OPAL-RT that allows the user to replace the Simulink console by a LabVIEW panel.

RT-LAB 2019.1 is compatible with **TestDrive v2.8**.

3.2.12 RT-XSG Compatibility

Integration of RT-LAB with the Xilinx System Generator toolbox is provided by the OPAL-RT RT-XSG product. Compatibility of RT-XSG with MATLAB and Xilinx System Generator versions can be found in RT-XSG Installation Guide.

RT-LAB 2019.1 is compatible with all versions of RT-XSG between **RT-XSG 2.0** and **RT-XSG 3.2**.

RT-XSG users should refer to the **OPAL-RT RT-XSG Toolbox User Guide** in order to validate compatibility between the FPGA boards in OPAL-RT chassis, Xilinx ISE Design Suite (including System Generator for DSP) and MATLAB.

3.3 OPAL-RT Chassis Compatibility

OPAL-RT chassis have reconfigurable FPGA boards for which the compatibility varies depending on the version of RT-LAB. It is important to note that the same OPAL-RT chassis (ex: OP5600) may be powered by different FPGA boards (ex: OP5142_1, OP5142_2, ML605_1, and ML605_2).

In the column header, the different possibilities of FPGA boards are under each OPAL-RT chassis. The FPGA boards are identified with their firmware filename prefix. For example, the "MMPK7_1-EX-0001-2_2_2_4-RESOLVER_IO-02-01.bin" firmware file is for the "MMPK7_1" FPGA board in the "OP4500" OPAL-RT chassis.

Table 8 below presents the compaibility between RT-LAB versions and OPAL-RT chassis that serve as target nodes. In other words, all these OPAL-RT chassis have processor power. The OP5031 chassis is the only one without an FPGA board and, without an I/O expansion box, has no I/O capability.

Table 8: RT-LAB vs OPAL-RT Chassis (Target Nodes)

	OP5031 (no FPGA board)	OP5600 OP5142_1 OP5142_2 ML605_1 ML605_2	OP5707 VC707_2	OP4500 MMPK7_1 MMPK7_2	OP4510 TE0741_2 TE0741_3 TE0741_4	OP4200 MEZX5_ 1	OP5650 OP5143
RT-LAB 11.0.x	Y	Y	Y	Y	Y	-	-
RT-LAB 11.1.x	Y	Y	Y	Y	Y	Y	-
RT-LAB 11.2.x	Y	Y	Y	Y	Y	Y	-
RT-LAB 11.3.x	Y	Y	Y	Y	Y	Y	11.3.4
RT-LAB 2019.1	Y	Y	Y	Y	Y	Y	Y
RT-LAB 2019.2	Y	Y	Y	Y	Y	Y	Y
RT-LAB 2019.3	Y	Y	Y	Y	Y	Y	Y
RT-LAB 2020.1	Y	Y	Y	Y	Y	Y	Y
RT-LAB 2020.2	Y	Y	Y	Y	Y	Y	Y

Table 9 below presents the compatibility between RT-LAB versions and OPAL-RT chassis that serve as I/O expansion boxes. In other words, these chassis have no processor power. They are either used to extend the I/O capability of a target node or to add more custom FPGA logic.

Table 9: RT-LAB vs OPAL-RT Chassis (I/O Expansion Boxes)

	OP5607: VC707_2	OP7000 OP7160_1 OP7160_2 OP7161_1 OP7161_2	OP7020 VC707_1	OP4520 TE0741_2 TE0741_3 TE0741_4	OP7170
RT-LAB 11.0.x	Y	Y	Y	Y	-
RT-LAB 11.1.x	Y	Y	Y	Y	-
RT-LAB 11.2.x	Y	Y	Y	Y	-
RT-LAB 11.3.x	Y	Y	Y	Y	-
RT-LAB 2019.1	Y	Y	Y	Y	-
RT-LAB 2019.2	Y	Y	Y	Y	Y
RT-LAB 2019.3	Y	Y	Y	Y	Y
RT-LAB 2020.1	Y	Y	Y	Y	Y
RT-LAB 2020.2	Y	Y	Y	7	Y

Installing RT-LAB on Windows Command Stations and Windows Target Nodes

Here we describe the installation of RT-LAB host and target components on a Windows PC.

This chapter will guide you through:

- [4.1 - Installing MATLAB™ and Simulink toolboxes](#)
- [4.2 - Installing RT-LAB Software on your Host Station](#)
- [4.3 - Configuring your firewall application](#)
- [4.4 - Starting RT-LAB](#)
- [4.5 - Validating the installation](#)
- [4.6 - Installing the LabVIEW API](#)
- [4.7 - Installing TestStand](#)
- [4.8 - Configuring ScopeView with EMTP-RT](#)
- [4.9 - Setting Up the Windows PC as a Target Node](#)

4.1 Installing MATLAB™ and Simulink toolboxes

RT-LAB provides full integration with MATLAB™ / Simulink. Please make sure this environment is available before installing RT-LAB.

1. Install a version of MATLAB™. You need the following toolboxes from MathWorks:
 - Simulink
 - Simulink Coder
 - SimPowerSystems (only for electrical simulations)
 - MATLAB™ Coder (optional)
 - Stateflow (optional)
2. Close all instances of MATLAB™.

4.2 Installing RT-LAB Software on your Host Station

The software can be downloaded directly from:
<http://www.opal-rt.com/software-downloads-documentation-0>.

1. Click **Download Now**.
 - Username: client
 - Password: clientOther option: a DVD is provided in the installation kit you received.

2. Wait for the Setup Wizard to open. If the wizard fails to start automatically, browse the contents of the disk and run **Setup/Setup.exe**. (You will need administrator privileges.)



Figure 1: Setup wizard

NOTE: If you are using the MinGW compiler, please search the **RT-LAB User Help** using the text **"mingw"** after RT-LAB installation for any further help required with this portion of the installation.

3. Follow the on-screen instructions to go through the installation process.
4. Please wait while RT-LAB is installed. This will only take a few minutes.
5. Once RT-LAB has been installed on your computer, click Finish to close the Wizard.

The RT-LAB installation software creates or modifies the following files:

- <MATLAB_ROOT>\toolbox\local\startup.m, for each version of MATLAB.
- <MATLAB_ROOT>\toolbox\local\setup_rtlab.m, for each version of MATLAB.

The RT-LAB installation software also adds the following paths to MATLAB:

- <RT-LAB installation dir>\common\bin;
- <RT-LAB installation dir>\Simulink\rtw\c\common
- <RT-LAB installation dir>\Simulink\libr[xxxxx];
- <RT-LAB installation dir>\Simulink\m;
- <RT-LAB installation dir>\common\bin\x32;

where **xxxxxx** is r2011b for MATLAB 7.13, r2012a for MATLAB 7.14 and so on.

4.3 Configuring your firewall application

The RT-LAB installation wizard asks the user's permission to automatically configure Windows Firewall. However, if you choose not to allow this automatic configuration, or if you are using a third-party firewall application, then a manual configuration will be necessary; you must allow the following programs through your firewall application before launching RT-LAB:

- <RT-LAB installation dir>\common\ui\RT-LAB.exe
- <RT-LAB installation dir>\common\bin\Controller.exe
- <RT-LAB installation dir>\common\bin\MetaController.exe
- <RT-LAB installation dir>\common\bin\OpalD.exe
- <RT-LAB installation dir>\common\bin\OpalDLoader.exe
- <RT-LAB installation dir>\common\bin\x32\OpalNode.exe
- <RT-LAB installation dir>\common\bin\x32\PanelViewerXXXX.exe
(where XXXX represents the supported LabVIEW versions)

As an example, the following procedure shows how to manually configure Windows Firewall (once again, performing these steps is not necessary if you have accepted the automatic configuration).

- Launch Windows Firewall.
- Click on **Allow a program or feature through Windows Firewall**.
- In the new window, click on **Allow another program...**
- In the new window, click on **Browse...** and choose the programs listed above.

If you have any questions about the configuration of your firewall application, please contact the third party software's technical support. If RT-LAB presents any undesirable behavior arising from this configuration, please contact OPAL-RT support team (support@opal-rt.com).

4.4 Starting RT-LAB

To launch the RT-LAB interface:

- Double-click the RT-LAB shortcut on your Desktop,
- Or use the Windows start menu (**All Programs/OPAL-RT/RT-LABv2017.X.Y.Z/RTLAB**).

You will then be asked to select a workspace.

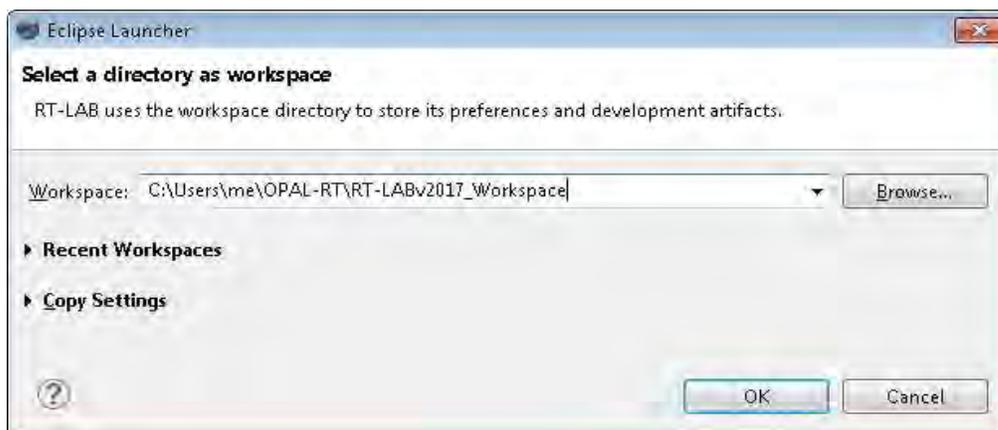


Figure 2: Selecting a workspace

The workspace is the directory in which RT-LAB will store all the files required for your simulations. We recommend you create a new and empty directory, as this will help you understand the exercises in this guide.

Click the **OK** button and wait for the Welcome page to appear.

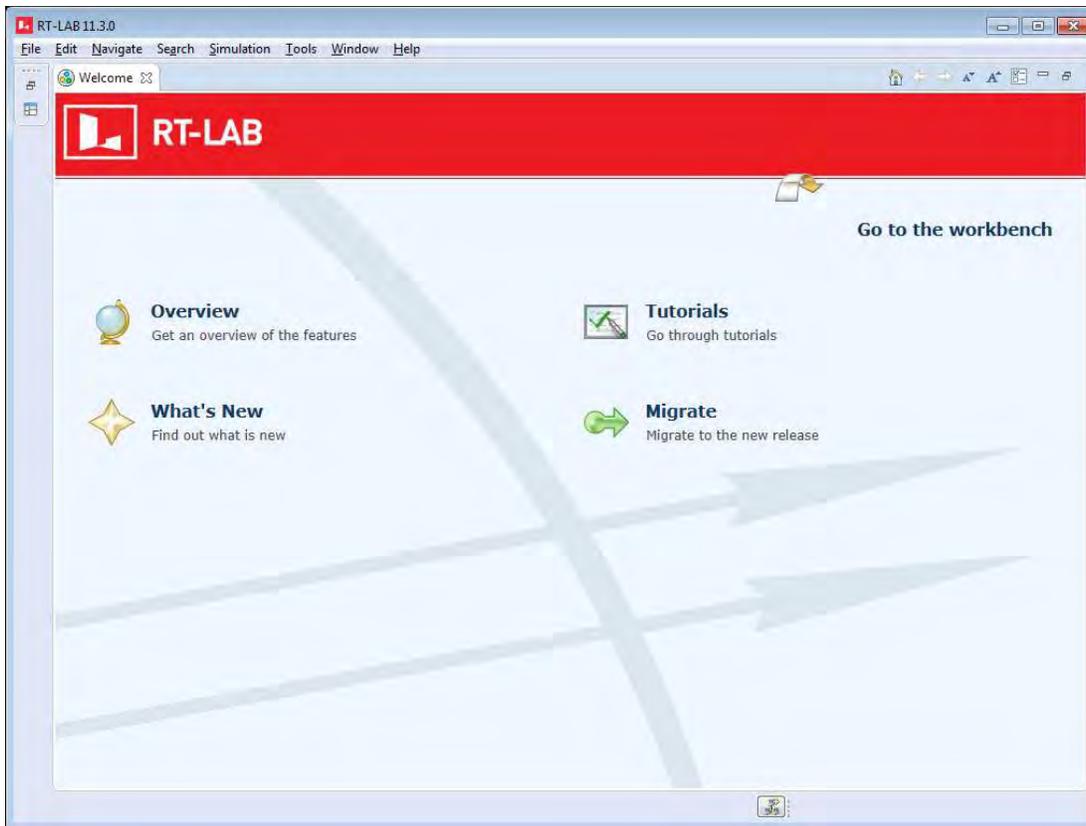


Figure 3: Welcome page

The Welcome page provides quick access to tutorials and documentation. You can open it later from the menu **Help / Welcome**. For now, click **Go to the workbench** to open the main RT-LAB window.

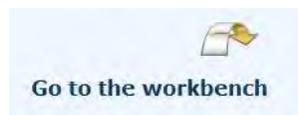


Figure 4: Go to the workbench

You now have access to the workbench.

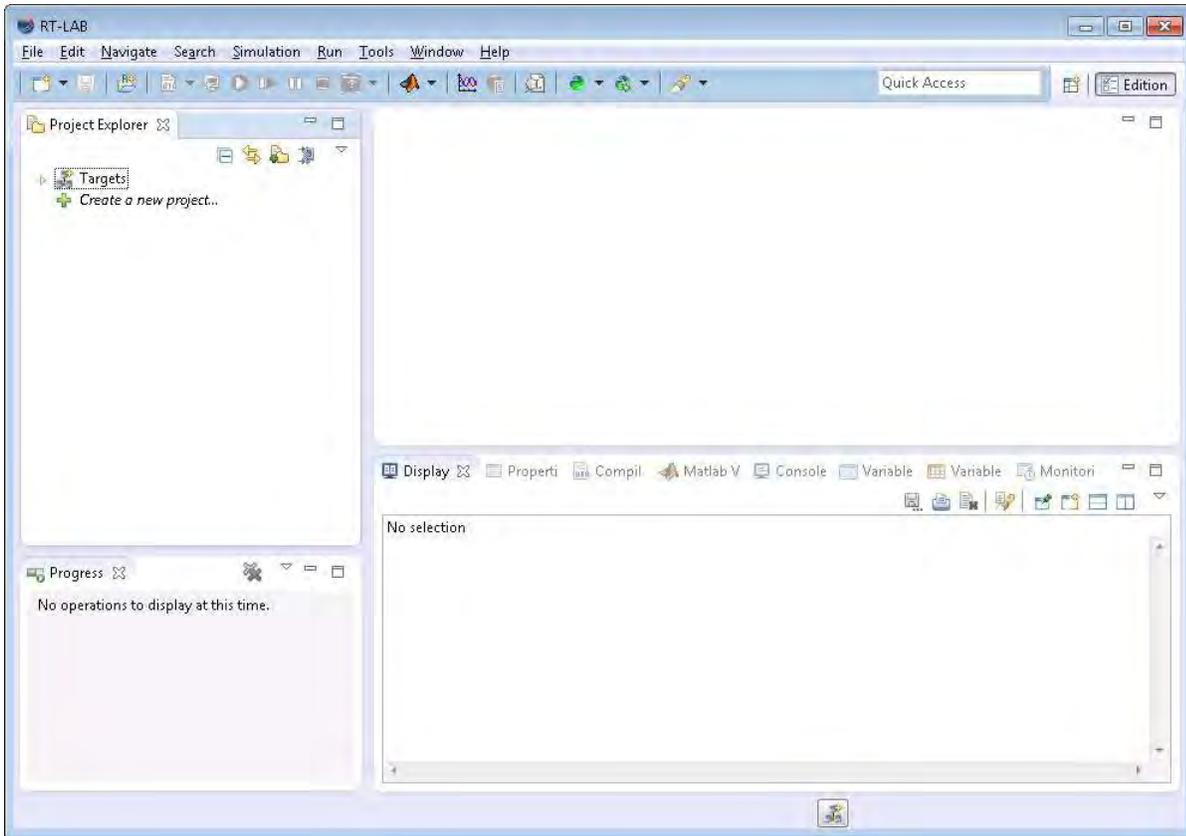


Figure 5: RT-LAB workbench

4.5 Validating the installation

For validation, we recommend that you verify that all toolboxes have been installed in MATLAB™.

1. Open MATLAB™.
2. Type `ver` in the command window of MATLAB™.
3. Verify that you see the following toolboxes from OPAL-RT:
 - RT-LAB
 - ARTEMIS Blockset (optional depending on license)
 - RT-EVENTS Blockset (optional depending on license)
 - RT-XSG (optional depending on license)

If any of the OPAL-RT toolboxes have not been installed in your instance of MATLAB™, you can add them manually. Run the following “m” script in your MATLAB™ to install them.

- RT-LAB: `C:\OPAL-RT\RT-LAB\[RT-Lab version]\simulink\m\setup_rtlab.m`
- ARTEMIS Blockset: `C:\OPAL-RT\ARTEMIS\[ARTEMIS version]\art_m\setup_artemis.m`
- RT-EVENTS Blockset: `C:\OPAL-RT\RT-EVENTS\[RT-EVENTS version]\rte_m\setup_rtevents.m`
- RT-XSG: `C:\OPAL-RT\RT-XSG\[RT-XSG version]\rtxsg_m\setup_rtxsg.m`

4.6 Installing the LabVIEW API

The installation of the LabVIEW API is done through an executable (.exe file) located in the RT-LAB installation directory. One installed, there will be one VI (Virtual Instrument) per API function. LabVIEW must be installed first, and the API must be installed for each version of LabVIEW you want to use.

1. Run `<RT-LAB installation dir>\labview\XXXX\Volume\setup.exe` (where XXXX represents the desired LabVIEW version)
2. Follow the instructions and click **Finish** once the installation is complete.

4.7 Installing TestStand

During the installation process, RT-LAB will add certain files to TestStand if TestStand was installed prior to installing RT-LAB. RT-LAB makes all the necessary adjustments so that most of the RT-LAB API functions can be readily used in TestStand. The supported versions of TestStand are listed in the [National Instruments TestStand Compatibility](#) section.

One additional adjustment to TestStand that you may want to perform manually to avoid having to locate OpalApiTs.dll when you open sequences that use the RT-LAB API is the following:

1. Run TestStand and log in as administrator.
2. Go to the **Configure** menu and open the **Search Directories** window.
3. Make sure the item **PATH environment variable** is checked.
4. Click **OK** to apply any changes.

Warning: It is strongly discouraged to add the complete path to the location of OpalApiTs.dll for a certain RT-LAB release because this path will not be updated when you install a newer version of RT-LAB.

4.8 Configuring ScopeView with EMTP-RT

RT-LAB and EMTP-RV can be installed on the same host computer and can both use ScopeView. Normally, both applications will use two different versions of ScopeView but, if required, it is possible to configure EMTP-RV to start the version that is packaged with RT-LAB. To do so, follow these steps:

1. Click the Windows **Start > Run ...** menu or click **WIN+R**.
1. Start the Registry Editor by typing **regedit.exe**
2. Locate the following registry:
HKEY_LOCAL_MACHINE\SOFTWARE\ScopeView.
3. Change the InstallPath value to the ScopeView installation folder package with RT-LAB. Example: `<RT-LAB installation dir>\ScopeView.`

4.9 Setting Up the Windows PC as a Target Node

The Windows command station may also be used as a target node for slow pseudo real-time applications. Please note that you must use an OPAL-RT chassis with a real-time operating system as a target node in order to achieve real-time performances.

4.9.1 Installing Visual C++

As the model is going to be compiled on the command station with a Windows target node, you must install Microsoft Visual C++ 10.0. If you want multiple users on this station to be able to use Microsoft Visual C++ you must install it as administrator. When the installation program prompts you to register environment variables answer yes. The installer registers those variables in the current user local variables. To compile as another user, you must first copy those variables from the current user variables to all user variables.

4.9.2 Registering Variables

If Visual C++ 10.0 is already installed, the user performing the installation may have issues with the installer that sets environment variables. To correct the situation, you need to do the following:

1. Press the Windows Start button, and start typing **Edit the system environment variables**. Click on the match in the **Control Panel** section.

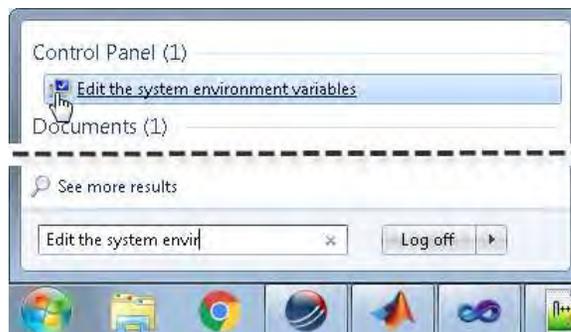
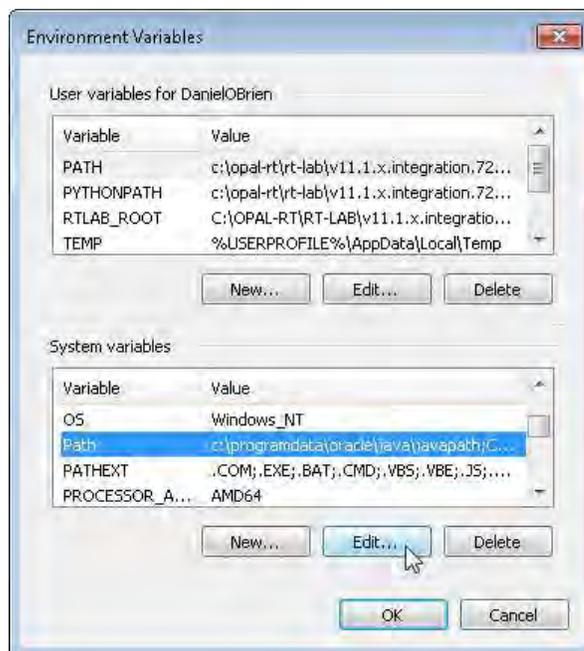


Figure 6: Edit the system environment variables

2. In the **System Properties** window, under the **Advanced** tab, click on **Environment Variables...**
3. In the **System variables** section, select the **Path** variable and click **Edit...**



4. Append the following paths to the **Variable value** field (with a semi-colon between each path). Note that these paths correspond to a typical installation of Visual Studio.

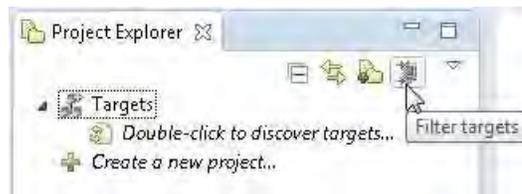
C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\Tools;
C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\bin;

Your command station is now ready to compile, execute and debug models.

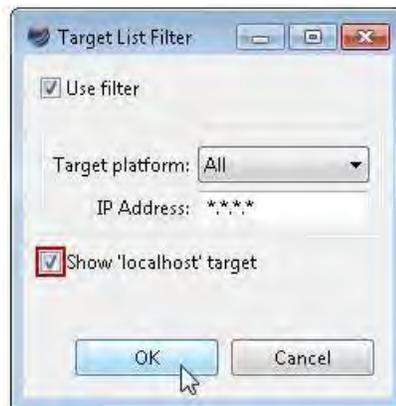
4.9.3 Displaying localhost

In order to display the Windows command station as a target node:

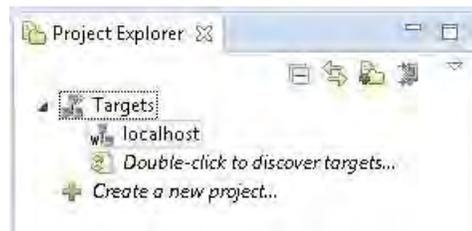
1. In the Project Explorer, click **Filter targets**.



2. Check **Show 'localhost' target**.



3. Click **OK**.



Please note that a license is needed to use a target node. For more information, read about [Activating a License on the Target Node](#).

Connecting your Real-Time Target Node to your Network

5

It is now time to unpack your real-time simulator (also called “target” or “target node” in this document). The simulator communicates with the Windows command station (host computer) via TCP/IP. The simulator can be connected either directly to your host computer ([Direct connection](#)) or via your local area network ([Local area network configuration](#)).

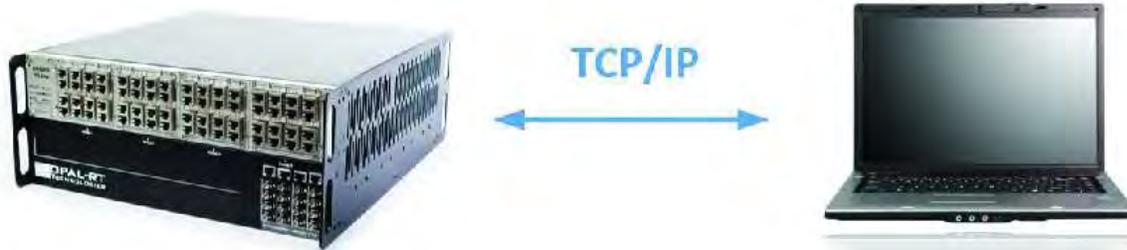


Figure 7: Real-Time Simulator (Target) and Windows Command Station (Host) Communication

This chapter will guide you through:

- [5.1 - Direct connection](#)
- [5.2 - Local area network configuration](#)
- [5.3 - Discovering target nodes](#)

5.1 Direct connection

First, locate the IP address of your simulator. The target has a fixed IP address. You can find the IP address in two different places:

- On a sticker attached to your simulator, or,
- In the “SystemDescription [Customer Name]_[Project Number].pdf” located on your DVD. You will find the IP address within the “Simulator General Overview” section.

Use the Ethernet cable found in the box to connect the simulator from its Ethernet port to your computer’s LAN port.

In order to establish communication, you must now modify the IP address of your host computer so that it is in the same sub-network as the simulator.

1. Click the network icon in your Windows icon tray, and then click on “Open Network and Sharing Center”,

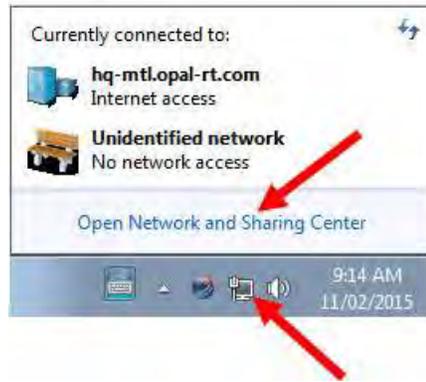


Figure 8: Open Network and Sharing Center

2. Click "Local Area Connection",



Figure 9: Local Area Connection

3. Click the Properties button,
4. Choose the connection "Internet Protocol Version 4 (TCP/IPv4)", and then click "Properties"

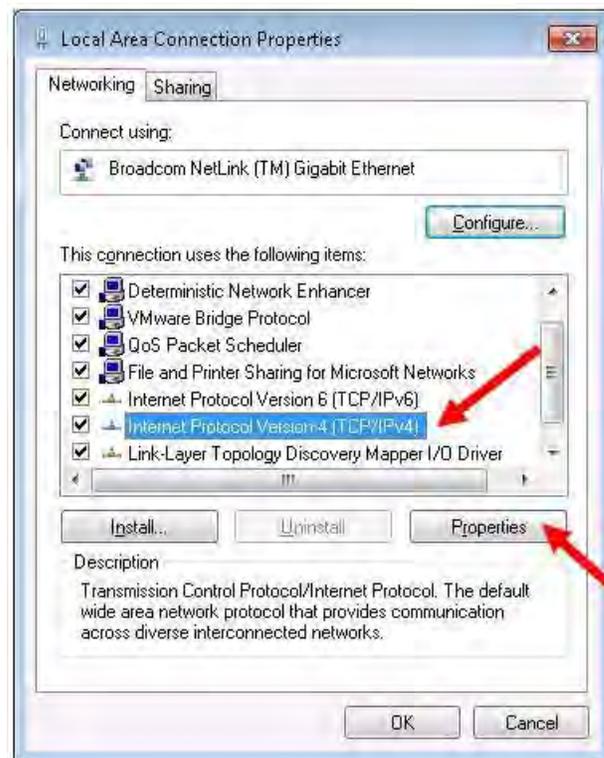


Figure 10: Windows Local Area Connection Properties

5. Choose “Use the following IP address” and enter the correct IP address and subnet mask. Usually, the first three numbers of the IP address should be the same as your simulator. The last number should be between 1 and 255 and should be different from your simulator. The subnet mask will be “255.255.255.0”

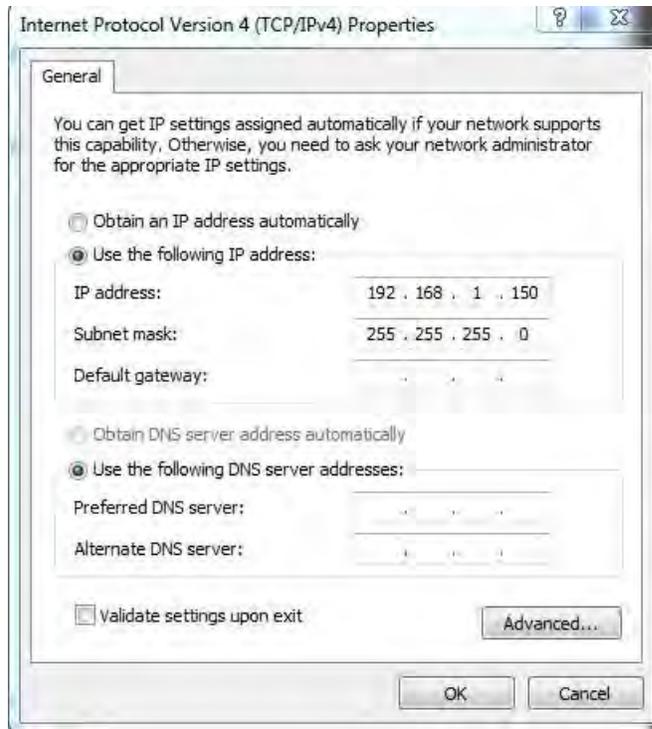


Figure 11: TCP/IPv4 Properties

5.2 Local area network configuration

If you want to use your simulator in a corporate network environment instead of a [Direct connection](#), the following requirements must be met.

1. The Windows command station can ping an IP address of the simulator.
2. The simulator is able to ping an IP address of the Windows command station.

Note: In order to do so, the Windows command station and the target node are either in the same subnet or they are configured with a default gateway that allows them to communicate together.

Your IT department should be able to meet the above requirements. If you are familiar with Linux distributions, you may connect a computer monitor, keyboard, and mouse to the simulator in order to set the appropriate configuration for your network. Here is a procedure to reconfigure the IP address of the simulator using RT-LAB on the Windows command station. It supposes that the Windows command station is in the same subnet as the simulator (which has a static IP address and the preconfigured subnet mask of 255.255.255.0).

1. Ask your IT department if you can add a static IP address component on the network to which your Windows command station will be able to communicate. They will most likely give you a range of IP addresses that the simulator can use.
2. Follow the [Direct connection](#) procedure in order to establish a temporary connection between the Windows command station and the target node.
3. Add your target in RT-LAB with the [Discovering target nodes](#) method.

4. The IP address of the simulator will need to be changed to an address given by your IT department. To do so, please follow these steps:
 - In RT-LAB, double click the target to open the **Target Editor**. In the **Overview** tab, you will see the actual IP address.
 - Change the IP address to the one from your IT department.
 - The system will ask you if you would like to change the IP address. Click **Yes**.

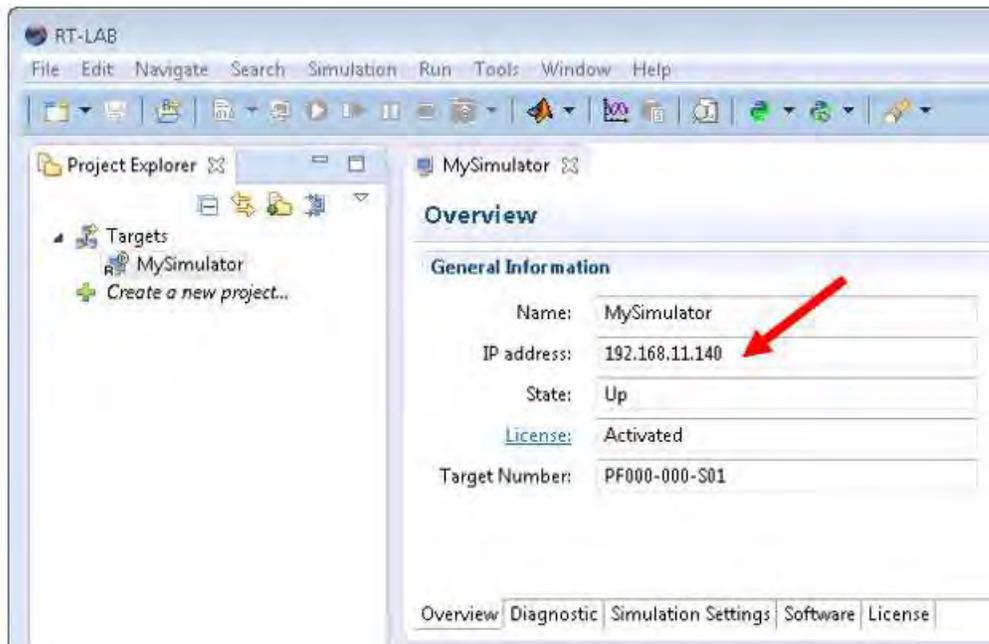


Figure 12: Change IP address

5. Disconnect the simulator from your Windows command station.
6. Connect both the simulator and your Windows command station to the network.
7. Return to the original network configuration for your Windows command station.
8. Remove the target you previously added in RT-LAB (right-click on it and click **Delete** then **OK**).
9. Add the target with the new IP address configuration using the **Discovering target nodes** method again.

5.3 Discovering target nodes

In RT-LAB's Project Explorer, double-click this item to discover available simulators on your network.

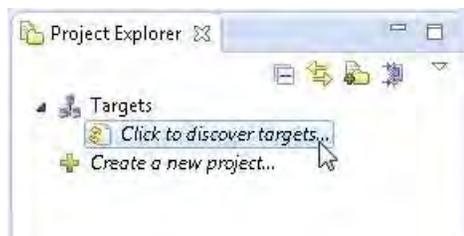


Figure 13: Discovering targets

After a short time, the following window should appear and show your simulator. Edit the name of your simulator as you wish, then click **Finish**.

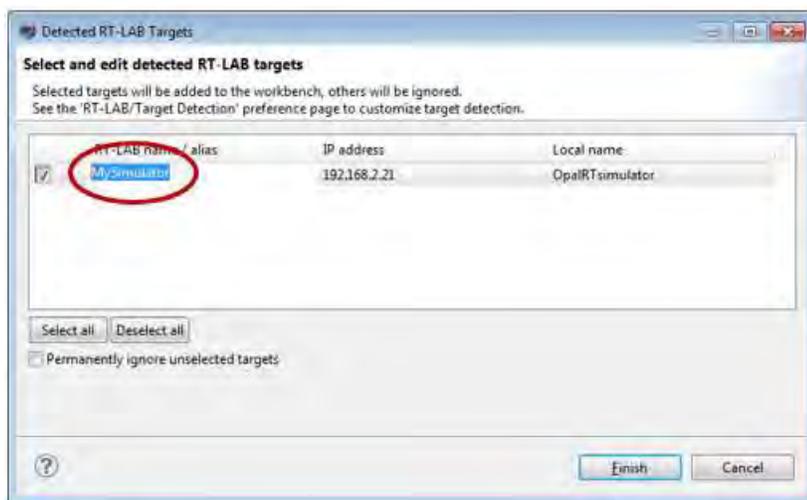


Figure 14: Detected targets

Your simulator is now available in the RT-LAB interface.

If your simulator is not automatically detected:

1. Create a target node in RT-LAB: in the Project Explorer, right-click on Targets then select New / New Target.
2. Enter a name and the specific IP address for the target.
3. Now click **Ping** to ensure that the target is available. If not, please contact your network administrator or follow the instructions on this page: <http://www.opal-rt.com/kb-article/how-change-ip-address-or-ip-mask-redhat-target-through-telnet>. Otherwise, simply click **Finish** and your target will appear in the Project Explorer.

Installing RT-LAB on an OPAL-RT Simulator

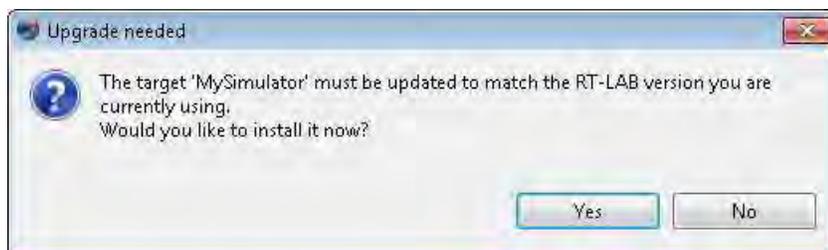
This section describes the installation of RT-LAB target components on an OPAL-RT simulator. In other words, it includes installation information for target nodes that are running on OPAL-RT Linux (x86-based) or Petalinux (ARMv7-based) real-time operating systems. OPAL-RT simulators already have RT-LAB installed on it. This section only applies to users that want to install a new version of RT-LAB.

Please note that a license is needed to use a target node. For more information, read about [Activating a License on the Target Node](#).

6.1 Installing RT-LAB on the Target Node

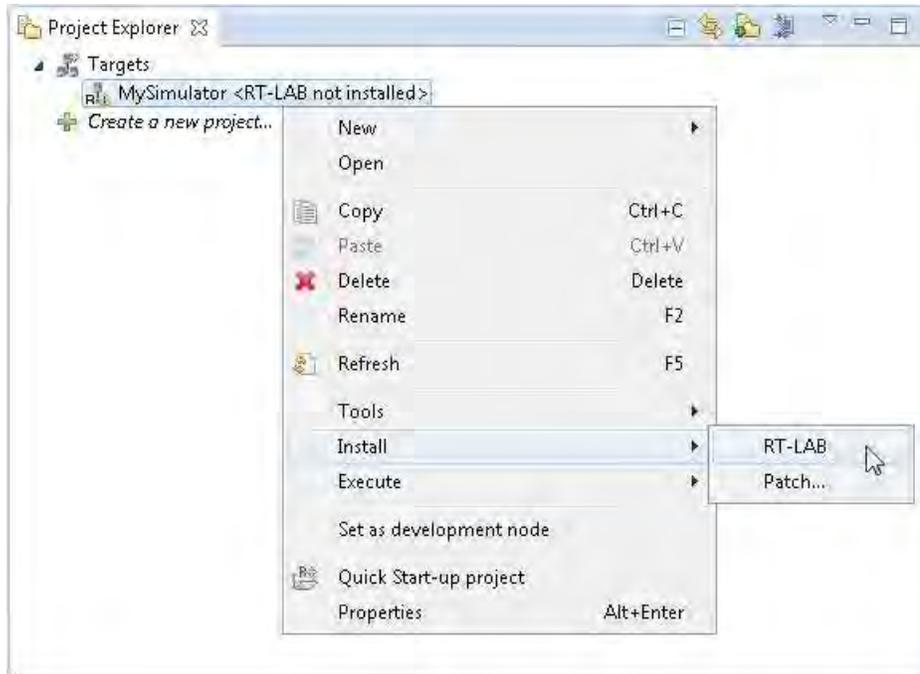
6.1.1 Installing from RT-LAB on the Windows Command Station

As soon as it is needed, RT-LAB will prompt you to install the new version of the software on the target. Simply accept to upgrade the target node automatically.



To force the installation, you may:

1. Open RT-LAB.
2. Right-click on the target you want to install RT-LAB.
3. Select **Install/RT-LAB** from the popup menu.



6.1.2 Installing from the Network on an OPAL-RT Linux (x86-based) target node

To install from the network:

1. Open My Computer window
2. Go to **C:\OPAL-RT\RT-LAB\vX.X.X\target\redhawk**
3. Open another **My Computer** window.
4. Enter **ftp://ntuser@xxx.xxx.xxx.xxx/** in the address bar, **ntuser** in the username field, and **ntuser** in the password field. Note that **xxx.xxx.xxx.xxx** is the target node's IP address.
5. You are now connected by FTP to your compilation node.
6. Copy the RT-LAB archive file **rtlab-redhawk-[version_name].rpm** by dragging and dropping from the first window to the second.
7. Connect to the target node by telnet (i.e. **Start>Run> telnet xxx.xxx.xxx.xxx**) and login as **root**, password **oprt935\$** (password may also be **redhat**)
8. Type the following command to install RT-LAB:
rpm -ivh --force /home/ntuser/rtlab-redhawk-[version_name].rpm.

6.1.3 Installing from the Network on a Petalinux (ARMv7-based) target node

To install from the network:

1. Open My Computer window
2. Go to **C:\OPAL-RT\RT-LAB\vX.X.X\target\petalinux**
3. Open another **My Computer** window.

-
4. Enter `ftp://ntuser@xxx.xxx.xxx.xxx/` in the address bar, `ntuser` in the user name field, and `ntuser` in the password field. Note that `xxx.xxx.xxx.xxx` is the target node's IP address.
 5. You are now connected by FTP to your compilation node.
 6. Copy the RT-LAB archive file `rtlab-petalinux-[version_name].deb` by dragging and dropping from the first window to the second.
 7. Connect to the target node by telnet (i.e. `Start>Run> telnet xxx.xxx.xxx.xxx`) and login as `root`, password `oprt935$` (password may also be `redhat`)
 8. Type the following command to install RT-LAB:
`dpkg -i --force-overwrite /home/ntuser/rtlab-petalinux-[version_name].rpm.`

6.2 Uninstalling RT-LAB

6.2.1 Uninstalling from the RT-LAB GUI

To uninstall from the RT-LAB GUI:

1. Open **RT-LAB**.
2. Double-click on the target to open the **Target Editor**.
3. Open the **Software Page**.
4. Select the version to uninstall.
5. Click **Uninstall**.

6.2.2 Uninstalling from the Network on an OPAL-RT Linux (x86-based) target node

To manually uninstall:

1. Connect to the OPAL-RT Linux (x86-based) target node by telnet (i.e. `Start>Run> telnet xxx.xxx.xxx.xxx`) and login as `root`, password `oprt935$` (password may also be `redhat`). Note that `xxx.xxx.xxx.xxx` is the target node's IP address.
2. Type the following command
`rpm -e rtlab-redhawk-[version_name]`

Note: Please note that you can list the installed RT-LAB versions by typing the following command:
`# rpm -qa | grep rtlab.`

6.2.3 Uninstalling from the Network on a Petalinux (ARMv7-based) target node

To manually uninstall:

1. Connect to the Petalinux (ARMv7-based) target node by telnet (i.e. `Start>Run> telnet xxx.xxx.xxx.xxx`) and login as `root`, password `oprt935$` (password may also be `redhat` or `root`). Note that `xxx.xxx.xxx.xxx` is the target node's IP address.

2. Type the following command
dpkg --remove rtlab-petalinux-[version_name]

Note: Please note that you can list the installed RT-LAB versions by typing the following command:
dpkg-query --list | grep rtlab

Activating a License on the Target Node

Since RT-LAB 10.5, each target must be activated with a valid license to be able to run a simulation.

Two simple steps must be followed to activate a license on a target.

Table 10: Steps to request and install license on target

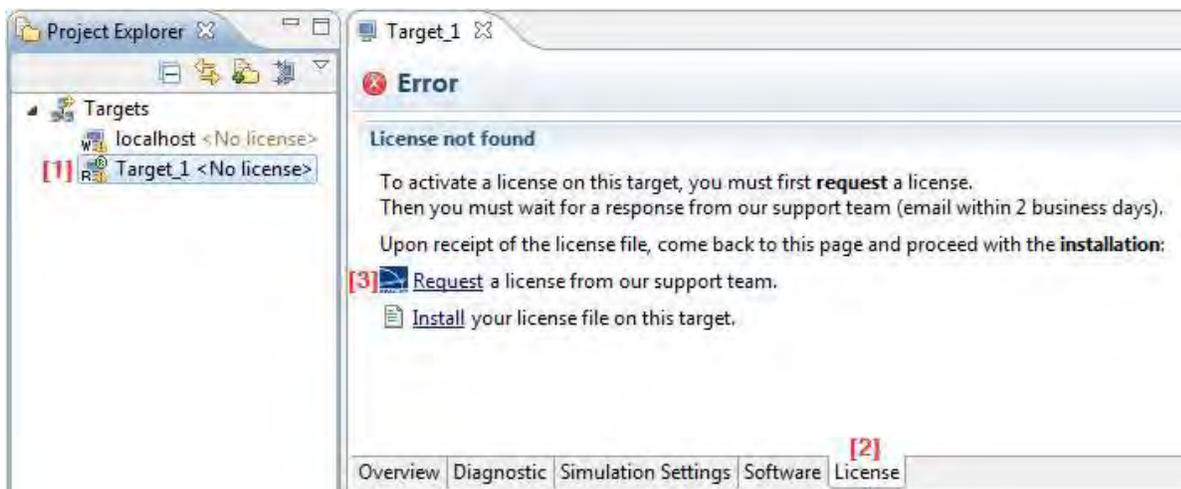
STEP	DESCRIPTION
Step 1: Request a license	Request a license if you don't have a license file.
Step 2: Install the license	Install the license on the target.

Before activating a target, you should ensure that your systems (host + target) are correctly configured by following these prerequisites:

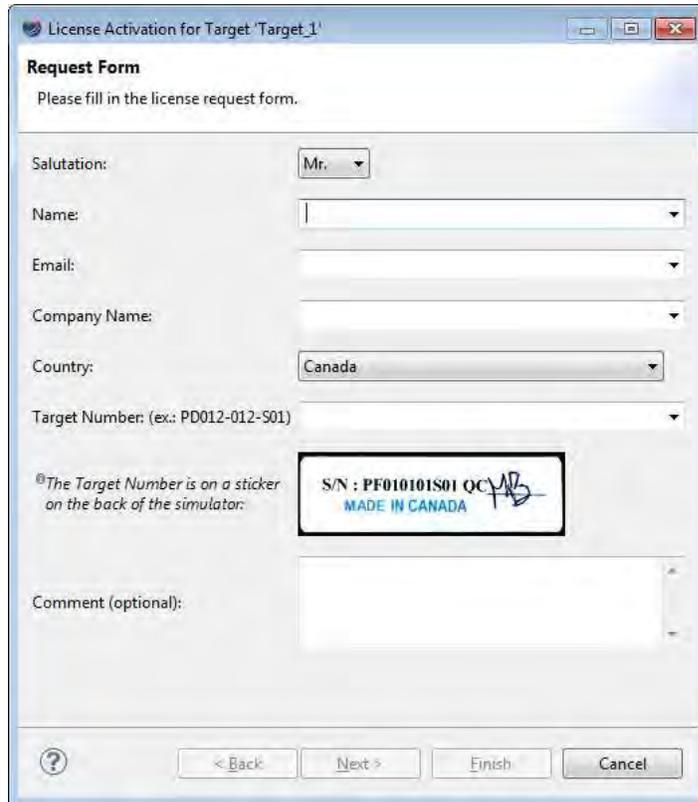
1. RT-LAB must be installed on your Windows host computer. See [Installing RT-LAB Software on your Host Station](#).
2. Each target must be available in RT-LAB. See [Connecting your Real-Time Target Node to your Network](#) for an OPAL-RT chassis or [Setting Up the Windows PC as a Target Node](#) for Windows.
3. RT-LAB must be installed on each target. See [Installing RT-LAB on the Target Node](#) for an OPAL-RT chassis.

7.1 Step 1: Request a License

1. Open the target (double click [1]), select the "License" tab [2] then click "Request" [3] to open a wizard that will guide you through the license request process.



2. Fill in the license request form then click "Next".



License Activation for Target 'Target_1'

Request Form
Please fill in the license request form.

Salutation: Mr. ▾

Name:

Email:

Company Name:

Country: Canada ▾

Target Number: (ex.: PD012-012-S01)

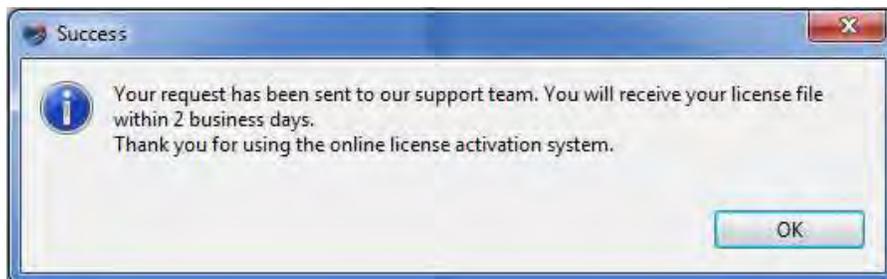
The Target Number is on a sticker on the back of the simulator.

S/N : PF010101S01 QCVB
MADE IN CANADA

Comment (optional):

Verify the request information and then click “Finish” to complete the process. The request will be automatically sent through a web service on the Internet.

Upon successful completion, you will receive this confirmation message:



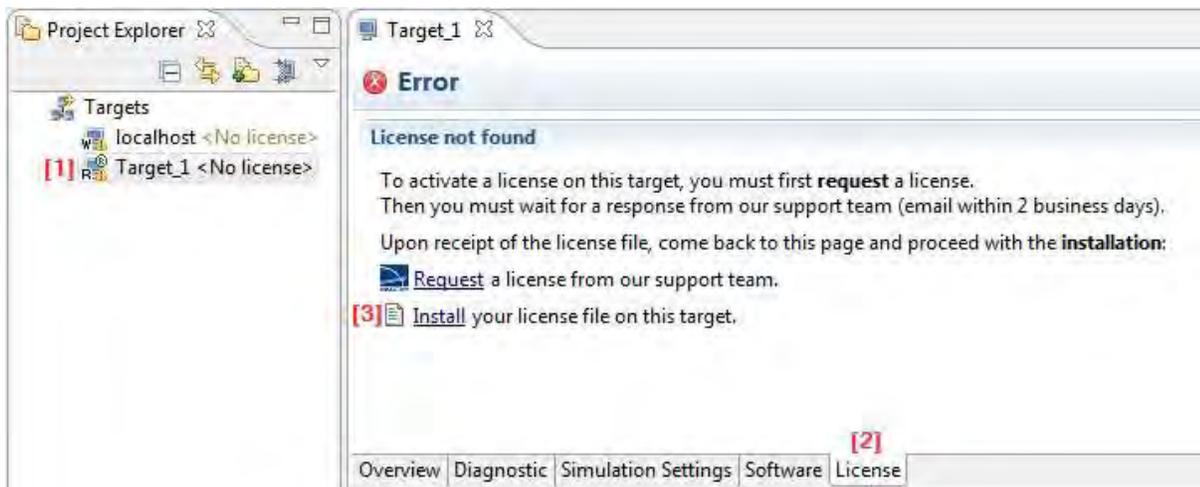
If your host is not connected to Internet or if a problem occurred, the wizard will prompt you to save the request file and send it by email to our support team (support@opal-rt.com):



Note: You can request licenses for more than one target from the RT-LAB menu *Help > Request a license*.

7.2 Step 2: Install the license

1. Before starting, ensure that you have received the required license for the target from our support team. Copy the license file to your command station.
2. Install the license: Open the target (double click [1]), select the "License" tab [2] then click "Install" [3] to open a file browser.



3. Browse to the license file and click "Open". The file will be installed on the target automatically.
4. The license page will be updated with the new license data.

This page contains the description, the license number, the expiration date and the list of available features for this target.

Note: You can install several license files at once from the RT-LAB menu *Help > Install a license file*.

7.2.1 IMPORTANT NOTES ABOUT LICENSES:

- Starting from RT-LAB version 10.5.x, the license is a binary file (.bin) instead of a ".lic" file.
- Starting from RT-LAB version RT-LAB 10.5.x, a license check is made at the load step.
- Starting from RT-LAB version RT-LAB 11.0.x, a license check is made at the **Compile and link generated code** build step. In order to compile your models, you must have a RT-LAB developer license.

- The license resides only on the target computer. RT-LAB 10.5.x+ can be installed with no license on an infinite number of host computers. For Windows target nodes, the license is placed in *C:\OPAL-RT\LicServer*.
- The RT-LAB 10.5.x+ license server is totally independent from the license server of RT-LAB 10.4.x and previous versions. Both license systems can cohabitate and be functional on the same hosts/targets.

The RT-LAB 10.5.x+ license server is totally independent from Acreesso's FLEXIm software previously used for the license management. It is now purely an OPAL-RT Technologies solution.