

RT-LAB Version 10.5 Installation Guide

Q012012-01

I M A G I N A T I O N

T O

R E A L - T I M E



© 2007 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, Real-Time Workshop and SimPowerSystem are trademarks of The Mathworks, Inc. LabVIEW is a trademark of National Instruments, Inc. QNX is a trademark of QNX Software Systems Ltd. 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 at the following coordinates:

Tool-Free (US and Canada)	1-877-935-2323 (08:30-17:30 EST)
Phone	1-514-935-2323
Fax	1-514-935-4994
E-mail	support@opal-rt.com info@opal-rt.com sales@opal-rt.com
Mail	1751 Richardson Street Suite 2525 Montreal, Quebec H3K 1G6
Web	www.opal-rt.com

What to Expect from this Guide	1
--	---

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
Components List	7

CHAPTER 3: REQUIREMENTS

Hardware Requirements	9
Software Requirements	9
MATRIXx Compatibility	10
National Instruments LabVIEW Compatibility	11
National Instruments TestStand Compatibility	11
Microsoft Visual C++ compatibility	11
QNX RTOS Compatibility	12
Redhawk / Redhat Linux OS Compatibility	12
Python compatibility	13
RT-Events Compatibility	13
ARTEMIS Compatibility	14
TestDrive Software Compatibility	14
Xilinx XSG Compatibility	15

CHAPTER 4: INSTALLING RT-LAB COMPONENTS ON WIN32 MACHINES

Installing RT-LAB onto your Command Station	17
Configuring the Target Nodes	17
Retrieving the License File	18
Installing the License File	20

Setting Up the Command Station for Windows Target Configurations	21
Installing updates	22
Installing the LabVIEW API	22
Installing for TestStand	23
Configuring ScopeView with EMTP-RT	23

CHAPTER 5: INSTALLING RT-LAB COMPONENTS ON QNX 6

Installing QNX 6.3.2 or QNX 6.5	25
Installing RT-LAB on QNX 6.x Targets	25
Installing RT-LAB updates on target	26

CHAPTER 6: INSTALLING RT-LAB COMPONENTS ON REDHAT LINUX MACHINES

Installing the Operating System	27
Installing RT-LAB Target Node	27
Installing RT-LAB updates	27
Uninstalling RT-LAB	28

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/Real Time Workshop
- 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, Visual Basic, Python and 3D Virtual Reality Tools
- Supports models from CarSim/TruckSim, GTPower, AMESim, Dymola, as well legacy models in FORTRAN 77/95

Run-time Features

- Distributed, synchronized execution across multiple processors, including, multiple CPU motherboards, Shared Memory or FireWire inter-processor 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 80 devices supported
- Choice of RTOS : QNX, RedHawk Linux

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

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 Components on Win32 Machines - This section describes the installation of RT-LAB host and target components on a Win32 machine.

Installing RT-LAB Components on QNX 6 - This section describes the installation of RT-LAB target components on a QNX6 machine.

Installing RT-LAB Components on Redhat Linux Machines - This section describes the installation of RT-LAB host and target components on a Redhawk Linux machine.

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 or simulation.

2.1 Definitions

2.1.1 Command Station

Development is always done on Windows NT/2000/XP/Vista/Seven or Redhawk Linux. 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** or **targets**. Target nodes can run using Windows NT/2000/XP/Vista/Seven, QNX 6.x (real-time) operating systems, or Redhat Linux (real-time) operating systems, allowing for a variety of simulation requirements. For super-computer performance, RT-LAB can be set up using a network of multi-processor computers.

2.2 Components List

There is a setup file for each operating system. The files are located on the RT-LAB CD. Here is a list of the files and where you can find them:

- For Windows NT/2000/XP/Vista/Seven - **rtlab-win32-[version_name].exe** in **rtlab\setup\win32**
- For QNX6 - **rtlab-qnx6-[version_name].tgz** in **\rtlab\setup\qnx6**
- For RedHat - **rtlab-redhawk-version_name.rpm** in **rtlab\setup\redhawk**

Requirements

3.1 Hardware Requirements

- Recommended configuration for a command station computer (Windows or Redhawk Linux OS): Pentium/AMD based PC with 1GB.
- Recommended configuration for a computation node: Pentium/AMD based PC with 128MB.

3.2 Software Requirements

3.2.1 Microsoft Windows Compatibility

Table 2: RT-LAB Versions vs. Windows Versions

	NT 4.0	2000	XP*	VISTA**	SEVEN**
RT-LAB 7.2.x	√	√	√	-	-
RT-LAB 8.0.x	-	√	√	-	-
RT-LAB 8.1.x	-	√	√	√	√
RT-LAB 8.2.x	-	√	√	√	√
RT-LAB 8.3.x	-	√	√	√	√
RT-LAB 8.4.x	-	√	√	√	√
RT-LAB 10.0.x	-	√	√	√	√
RT-LAB 10.1.x	-	√	√	√	√
RT-LAB 10.2.x	-	√	√	√	√
RT-LAB 10.3.x	-	√	√	√	√
RT-LAB 10.4.x	-	√	√	√	√
RT-LAB 10.5.x	-	√	√	√	√

Because RT-LAB uses Windows services and this feature is not available in Windows 95/98/ME, you need to be running Windows NT4/2000/XP/Vista/Seven to use RT-LAB. Furthermore, Windows NT4/2000/XP/Vista/Seven is much more stable and reliable than Windows 95/98/ME.

*RT-LAB has been tested on Windows XP 32 bits and 64 bits edition.

** Starting with v8.3, RT-LAB is fully compatible with Windows Vista and Windows 7. However, it will work with v8.1 and v8.2 with some manual configurations.

3.2.2 MATLAB Compatibility

Table 3: RT-LAB Versions vs. MATLAB Versions

	R2006B	R2007B	R2008A	R2008B	R2009B 32 BITS	R2010A 32 BITS	R2010B 32 BITS	R2011A 32 BITS	R2011B 32 BITS
RT-LAB 8.0.x	-	-	-	-	-	-	-	-	-
RT-LAB 8.1.x	√	-	-	-	-	-	-	-	-
RT-LAB 8.2.x	√	√	√	-	-	-	-	-	-
RT-LAB 8.3.x	√	√	√	√	-	-	-	-	-
RT-LAB 8.4.x	√	√	√	√	-	-	-	-	-

Table 3: RT-LAB Versions vs. MATLAB Versions

	R2006B	R2007B	R2008A	R2008B	R2009B 32 BITS	R2010A 32 BITS	R2010B 32 BITS	R2011A 32 BITS	R2011B 32 BITS
RT-LAB10.0.x	√	√	√	√	√	-	-	-	-
RT-LAB10.1.x	√	√	√	√	√	-	-	-	-
RT-LAB10.2.x	√	√	√	√	√	√	√	-	-
RT-LAB10.3.x	√	√	√	√	√	√	√	√	-
RT-LAB10.4.x	√	√	√	√	√	√	√	√	√
RT-LAB10.5.x	—	√	√	√	√	√	√	√	√

	R2010A 64 BITS	R2010B 64BITS	R2011A 64BITS	R2011B 64 BITS
RT-LAB10.5.x	√	√	√	√

RT-LAB 10.5 and later support some versions of 64-bit MATLAB: R2010a 64bits, R2010b 64bits, R2011a 64bits, R2011b 64bits.

Required Elements:

- **Matlab:** Underlying mathematical engine
- **Simulink:** Graphical design environment
- **Real-Time Workshop** (renamed as Simulink Coder in Matlab versions from R2011A): Code generator
- **Stateflow** (supported but optional): Stateful programming environment
- **Stateflow coder** (required if Stateflow is used): Code generator for Stateflow

3.3 MATRIXx Compatibility

Table 4: RT-LAB Versions vs. MATRIXx Versions

	V62.2	V71.X
RT-LAB 7.2.x	√	-
RT-LAB 8.0.x	√	-
RT-LAB 8.1.x	√	-
RT-LAB 8.2.x	√	-
RT-LAB 8.3.x	√	-
RT-LAB 8.4.x	√	-
RT-LAB 10.0.x	√	-
RT-LAB 10.1.x	√	-
RT-LAB 10.2.x	√	-
RT-LAB 10.3.x	√	-
RT-LAB 10.4.x	√	-
RT-LAB 10.5.x (end of support)	—	-

3.4 National Instruments LabVIEW Compatibility

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

The API is pre-compiled for LabVIEW 8.5.1 so you will have to “mass compile” it if you use a different version of LabVIEW. This is automatically done when installing the API using the installer as described in [Installing the LabVIEW API](#).

Table 5: RT-LAB Versions vs. LabVIEW Versions

	V8.5.X	V8.6.X	V2009	V2010/ V2010SP1
RT-LAB 10.2.x	√	√	√	√
RT-LAB 10.3.x	√	√	√	√
RT-LAB 10.4.x	√	√	√	√
RT-LAB 10.5.x	√	√	√	√

3.5 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.

Table 6: RT-LAB Versions vs. TestStand Versions

	V3.0	V3.1	V4.1
RT-LAB 7.2.x	√	√	–
RT-LAB 8.0.x	√	√	–
RT-LAB 8.1.x	√	√	–
RT-LAB 8.2.x	√	√	–
RT-LAB 8.3.x	√	√	√
RT-LAB 8.4.x	√	√	√
RT-LAB 10.0.x	√	√	√
RT-LAB 10.1.x	√	√	√
RT-LAB 10.2.x	√	√	√
RT-LAB 10.3.x	√	√	√
RT-LAB 10.4.x	√	√	√
RT-LAB 10.5.x	√	√	√

3.6 Microsoft Visual C++ compatibility

Microsoft Visual C++ is required on command station for some RT-LAB fonctionnalités (Windows target).

Table 7: RT-LAB Versions vs. MSVC++ Versions

	MSVC++ 6.0	MSVC++ 2005	MSVC++ 2008	MSVC++ 2010
RT-LAB 8.0.x	√	–	–	–
RT-LAB 8.1.x	√	–	–	–
RT-LAB 8.2.x	√	–	–	–
RT-LAB 8.3.x	√	–	–	–
RT-LAB 8.4.x	√	–	–	–

	MSVC++ 6.0	MSVC++ 2005	MSVC++ 2008	MSVC++ 2010
RT-LAB 10.0.x	√	–	–	–
RT-LAB 10.1.x	√	–	–	–
RT-LAB 10.2.x	√	–	–	–
RT-LAB 10.3.x	√	√	√	√
RT-LAB 10.4.x	√	√	√	√
RT-LAB 10.5.x	√	√	√	√

3.7 QNX RTOS Compatibility

QNX is used as a computation environment for RT-LAB. Models can be executed in real-time or as pure (unsynchronized) simulations. For pure simulation (no synchronization and no IO), Windows NT/2000/XP computation nodes can be used instead of QNX.

Table 8: RT-LAB Versions vs. QNX Versions

	V6.1	V6.2.1	V6.3.0	V6.3.2	V6.5
RT-LAB 7.2.x	√	√	√	√	–
RT-LAB 8.0.x	√	√	√	√	–
RT-LAB 8.1.x	√	√	√	√	–
RT-LAB 8.2.x	-	-	-	√	–
RT-LAB 8.3.x	-	-	-	√	–
RT-LAB 8.4.x	-	-	-	√	–
RT-LAB 10.0.x	-	-	-	√	–
RT-LAB 10.1.x	-	-	-	√	–
RT-LAB 10.2.x	-	-	-	√	√
RT-LAB 10.3.x	-	-	-	√	√
RT-LAB 10.4.x	-	-	-	√	√
RT-LAB 10.5.x	-	-	-	√	√

3.8 Redhawk / Redhat Linux OS Compatibility

Redhawk Linux OS may be used as a command station or as a target node. Models can be executed in real-time or as pure (unsynchronized) simulations.

Table 9: RT-LAB Versions vs. RedHawk/Redhat Linux Versions

	REDHAWK 2.2	REDHAWK 2.3	REDHAWK 4.1	REDHAWK 4.2	REDHAT 5.2
RT-LAB 7.2.x	√	√	-	-	-
RT-LAB 8.0.x	√	√	√*	-	-
RT-LAB 8.1.x	√	√	√	√**	-
RT-LAB 8.2.x	-	-	-	√	-
RT-LAB 8.3.x	-	-	-	√	-
RT-LAB 8.4.x	-	-	-	√	-
RT-LAB 10.0.x	-	-	-	√	√
RT-LAB 10.1.x	-	-	-	√	√
RT-LAB 10.2.x	-	-	-	√	√

Table 9: RT-LAB Versions vs. RedHawk/Redhat Linux Versions

	REDHAWK 2.2	REDHAWK 2.3	REDHAWK 4.1	REDHAWK 4.2	REDHAT 5.2
RT-LAB 10.3.x	-	-	-	√	√
RT-LAB 10.4.x	-	-	-	√	√
RT-LAB 10.5.x	-	-	-	√	√

*RedHawk 4.1 is only supported with RT-LAB 8.0.2 or later.

**RedHawk 4.2 is only supported with RT-LAB 8.1.7 or later.

3.9 Python compatibility

Python is required on command station and target for some RT-LAB fonctionnalités.

By default, the command station uses its own Python 2.6 release.

Note that Redhawk 4.2 only support Python versions up to 2.4.

Table 10: RT-LAB Versions vs. Python Versions

	PYTHON 2.2.X	PYTHON 2.3.X	PYTHON 2.4.X	PYTHON 2.5.X	PYTHON 2.6.X
RT-LAB 7.2.x	√	√	√	-	-
RT-LAB 8.0.x	√	√	√	-	-
RT-LAB 8.1.x	√	√	√	-	-
RT-LAB 8.2.x	√	√	√	-	-
RT-LAB 8.3.x	√	√	√	√	√
RT-LAB 8.4.x	√	√	√	√	√
RT-LAB 10.0.x	√	√	√	√	√
RT-LAB 10.1.x	√	√	√	√	√
RT-LAB 10.2.x	√	√	√	√	√
RT-LAB 10.3.x	√	√	√	√	√
RT-LAB 10.4.x	√	√	√	√	√
RT-LAB 10.5.x	√	√	√	√	√

3.10 RT-Events Compatibility

RT-Events is a blockset of Simulink developed by Opal-RT. It may be used in real-time when compiled with RT-LAB.

Table 11: RT-LAB Versions vs. RT-Events Versions

	RTE 2.0	RTE 2.1X	RTE 2.2.X	RTE 2.4.X	RTE 3.0.X	RTE 3.1.X	RTE 3.2.X	RTE 3.3.X	RTE 3.4.X	RTE 3.5.X	RTE 3.6.X	RTE 3.7.X
RT-LAB 7.2.x	-	-	-	√	√	-	-	-	-	-	-	-
RT-LAB 8.0.x	-	-	-	√	√	-	-	-	-	-	-	-
RT-LAB 8.1.x	-	-	-	√	√	√	√	√	√	√	-	-
RT-LAB 8.2.x	-	-	-	-	-	√	√	√	√	√	-	-
RT-LAB 8.3.x	-	-	-	-	-	√	√	√	√	√	-	-
RT-LAB 8.4.x	-	-	-	-	-	√	√	√	√	√	-	-
RT-LAB 10.0.x	-	-	-	-	-	√	√	√	√	√	-	-

Table 11: RT-LAB Versions vs. RT-Events Versions

	RTE 2.0	RTE 2.1X	RTE 2.2.X	RTE 2.4.X	RTE 3.0.X	RTE 3.1.X	RTE 3.2.X	RTE 3.3.X	RTE 3.4.X	RTE 3.5.X	RTE 3.6.X	RTE 3.7.X
RT-LAB 10.1.x	-	-	-	-	-	-	-	-	√	√	-	-
RT-LAB 10.2.x	-	-	-	-	-	-	-	-	√	√	-	-
RT-LAB 10.3.x	-	-	-	-	-	-	-	-	√	√	√	-
RT-LAB 10.4.x	-	-	-	-	-	-	-	-	√	√	√	√
RT-LAB 10.5.x	-	-	-	-	-	-	-	-	√	√	√	√

3.11 ARTEMIS Compatibility

ARTEMIS is a blockset of Simulink developed by Opal-RT. It may be used in real-time when compiled with RT-LAB.

Table 12: RT-LAB Versions vs. ARTEMIS Versions

	ARTEMIS V3.0	ARTEMIS V4.0	ARTEMIS V4.1	ARTEMIS V5.0	ARTEMIS V6.0	ARTEMIS V6.1	ARTEMIS V6.2
RT-LAB 7.2.x	√	√	-	-	-	-	-
RT-LAB 8.0.x	√	√	-	-	-	-	-
RT-LAB 8.1.x	√	√	√	-	-	-	-
RT-LAB 8.2.x	-	√	√	√	-	-	-
RT-LAB 8.3.x	-	√	√	√	-	-	-
RT-LAB 8.4.x	-	√	√	√	-	-	-
RT-LAB 10.0.x	-	√	√	√	-	-	-
RT-LAB 10.1.x	-	-	-	-	√	√	√
RT-LAB 10.2.x	-	-	-	-	√	√	√
RT-LAB 10.3.x	-	-	-	-	√	√	√
RT-LAB 10.4.x	-	-	-	-	√	√	√
RT-LAB 10.5.x	-	-	-	-	√	√	√

3.12 TestDrive Software Compatibility

TestDrive is a Software developed by Opal-RT that allows you to replace the Simulink console by a Labview panel.

Table 13: RT-LAB Versions vs. TestDrive Versions

	TESTDRIVE V2.2.0 AND V2.2.1	TESTDRIVE V2.2.2 AND V2.2.3	TESTDRIVE V2.5.X	TESTDRIVE V2.6.X	TESTDRIVE V2.8
RT-LAB 8.1.1 and 8.1.2	√	–	–	–	–
RT-LAB 8.1.3 to 8.1.6	–	√	√	–	–
RT-LAB 8.1.7	–	–	–	√	–
RT-LAB 8.2.x	–	–	–	√	–
RT-LAB 8.3.x	–	–	–	√	–
RT-LAB 8.4.x	–	–	–	√	–
RT-LAB 10.0.x	–	–	–	–	–
RT-LAB 10.1.x	–	–	–	–	–
RT-LAB 10.2.x	–	–	–	–	√
RT-LAB 10.3.x	–	–	–	–	√
RT-LAB 10.4.x	–	–	–	–	√
RT-LAB 10.5.x	–	–	–	–	√

3.13 Xilinx XSG Compatibility

Starting with v8.3, integration of RT-LAB with the Xilinx System Generator toolbox is provided by the Opal-RT RT-XSG product. Table 14 below gives the compatibility between RT-LAB and RT-XSG versions. Compatibility of RT-XSG with Matlab and Xilinx System Generator versions can be found in RT-XSG Installation Guide.

Table 14: RT-LAB vs RT-XSG

	RT-XSG 1.0	RT-XSG 1.1	RT-XSG 1.2	RT-XSG 1.3	RT-XSG 2.0	RT-XSG 2.1
RT-LAB 8.3.x	√	√	√	*	*	*
RT-LAB 8.4.x	√	√	√	*	*	*
RT-LAB 10.0.x	√	√	√	√	*	*
RT-LAB 10.1.x	√	√	√	√	√	*
RT-LAB 10.2.x	√	√	√	√	√	√
RT-LAB 10.3.x	√	√	√	√	√	√
RT-LAB 10.4.x	√	√	√	√	√	√
RT-LAB 10.5.x	√	√	√	√	√	√

Note: The symbol (*) indicates partial support. For instance, new FPGA controller boards added in the newer versions of RT-XSG are not included in older versions of RT-LAB.

In earlier RT-LAB versions, the XSG option was integrated with RT-LAB. Table 15 and 16 below gives the compatibility matrix of RT-LAB, Matlab and Xilinx ISE versions. Description of the required elements for each Xilinx version supported in these previous RT-LAB releases follows.

Table 15: Xilinx ISE versions versus FPGAs supported

RT-XSG-compatible board	Xilinx ISE v10.1.03, System Generator for DSP v10.1.03 ^a	Xilinx ISE and DSP Edition v11.	Xilinx ISE v12.1, System Generator for DSP v12.1	Xilinx ISE v12.2, System Generator for DSP v12.2	Xilinx ISE v12.3, System Generator for DSP v12.3	Xilinx ISE v12.4, System Generator for DSP v12.4	Xilinx ISE v13.1, System Generator for DSP v13.1	Xilinx ISE v13.2, System Generator for DSP v13.2	Xilinx ISE v13.3, System Generator for DSP v13.3	Xilinx ISE v13.4, System Generator for DSP v13.4	Xilinx ISE v14.1, System Generator for DSP v14.1
Opal-RT OP5130 (VirtexIIPro)	X	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Opal-RT OP5142 (Spartan3)	X	o	X	X	X	X	X	X	X	X	X
Xilinx ML50x (Virtex5)	X	o	X	X	X	X	X	X	X	X	X
Xilinx ML60x (Virtex6)	N/A	N/A	X	X	X	X	X	X	X	X	X
Opal-RT OP7000 (Spartan3)	N/A	N/A	X	X	X	X	X	X	X	X	X

a. Supported with RT-XSG 2.1.5.127 or older versions

N/A - Xilinx unsupported

X - Xilinx and RT-XSG supported

o - RT-XSG unsupported

Table 16: Matlab versions vs Xilinx ISE versions

Xilinx / Matlab compatibility	Xilinx ISE v10.1.03, System Generator for DSP v10.1.03 ^a	Xilinx ISE and DSP Edition v11.	Xilinx ISE v12.1, System Generator for DSP v12.1	Xilinx ISE v12.2, System Generator for DSP v12.2	Xilinx ISE v12.3, System Generator for DSP v12.3	Xilinx ISE v12.4, System Generator for DSP v12.4	Xilinx ISE v13.1, System Generator for DSP v13.1	Xilinx ISE v13.2, System Generator for DSP v13.2	Xilinx ISE v13.3, System Generator for DSP v13.3	Xilinx ISE v13.4, System Generator for DSP v13.4	Xilinx ISE v14.1, System Generator for DSP v14.1
Matlab R2007a	X	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Matlab R2007b	X	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Matlab R2008a	X	o	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Matlab R2008b	N/A	o	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Matlab R2009a	N/A	o	o	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Matlab R2009b	N/A	N/A	X	X	X	N/A	N/A	N/A	N/A	N/A	N/A
Matlab R2010a	N/A	N/A	N/A	o	o	o	o	o	o	N/A	N/A
Matlab R2010b	N/A	N/A	N/A	N/A	N/A	X	X	X	o	N/A	N/A
Matlab R2010b SP1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	o	o	N/A	N/A
Matlab R2011a	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	o	o	o
Matlab R2011b	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	o	X

a. Supported with RT-XSG 2.1.5.127 or older versions

N/A - Xilinx unsupported

X - Xilinx and RT-XSG supported

o - RT-XSG unsupported

Installing RT-LAB Components on Win32 Machines

Here we describe the installation of RT-LAB host and target components on a Win32 machine.

4.1 Installing RT-LAB onto your Command Station

An installation file for RT-LAB on Windows NT, 2000, XP, Vista and Seven is included on the CD-ROM in the RT-LAB software package.

To install RT-LAB onto your Command Station:

1. Insert the RT-LAB CD-ROM into your **Command Station** computer's CD-ROM drive. The installation should start automatically. Click on the *Install OPAL-RT products* link and follow instructions. If the installation does not start automatically:
 - Go to **Start>Run....**
 - In the **Run** window, type **D:\rtlab\setup\win32\rtlab-win32-[version_name].exe** and click on **OK**. This assumes **D:** is assigned to your CD-ROM drive; if not, enter the CD-ROM drive's letter.
2. Follow the online installation instructions. The files are copied onto your hard disk.
3. Restart your computer after completing installation.

The RT-LAB installation software adds new paths to your MATLAB, creating or modifying the following file:

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

RT-LAB installation software adds:

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

where **xxxxxx** is r2006b for Matlab 7.3, r2007b for Matlab 7.5 and so on.

4.2 Configuring the Target Nodes

After the Windows installation, target nodes must be installed.

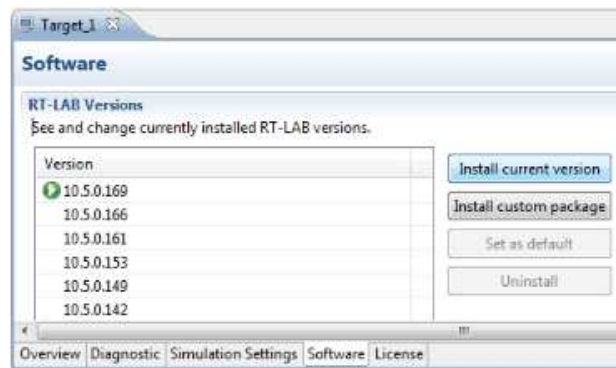
The target nodes must be added to the list of targets in the Project Explorer of RT-LAB. If you have installed RT-LAB over an existing version or have the configuration of an older version, you can skip this procedure.

1. Open RT-LAB.
2. In Project Explorer, right click on Targets.
3. Click the New Target menu item from the contextual menu New of Targets.
4. Specify the name and the IP Address of the target.

5. Press the Ping button to validate the entries.
6. Press Finish button.



7. Once your newly-added target appears in the "Targets" list, double-click on it and go to the "Software" tab. Click "Install current version".



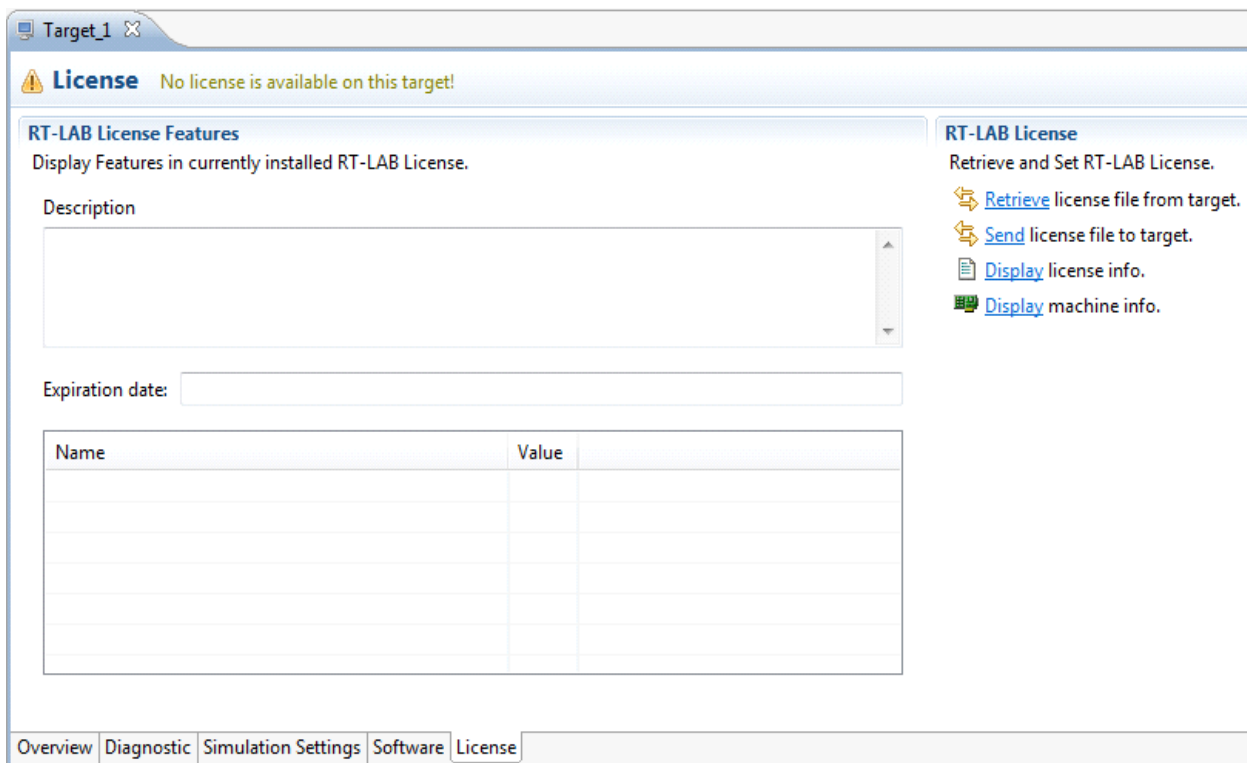
4.3 Retrieving the License File

Once RT-LAB 10.5 or later is installed you must retrieve the license information from the real time target after configuring the target nodes and installed on the target the new RT-LAB version.

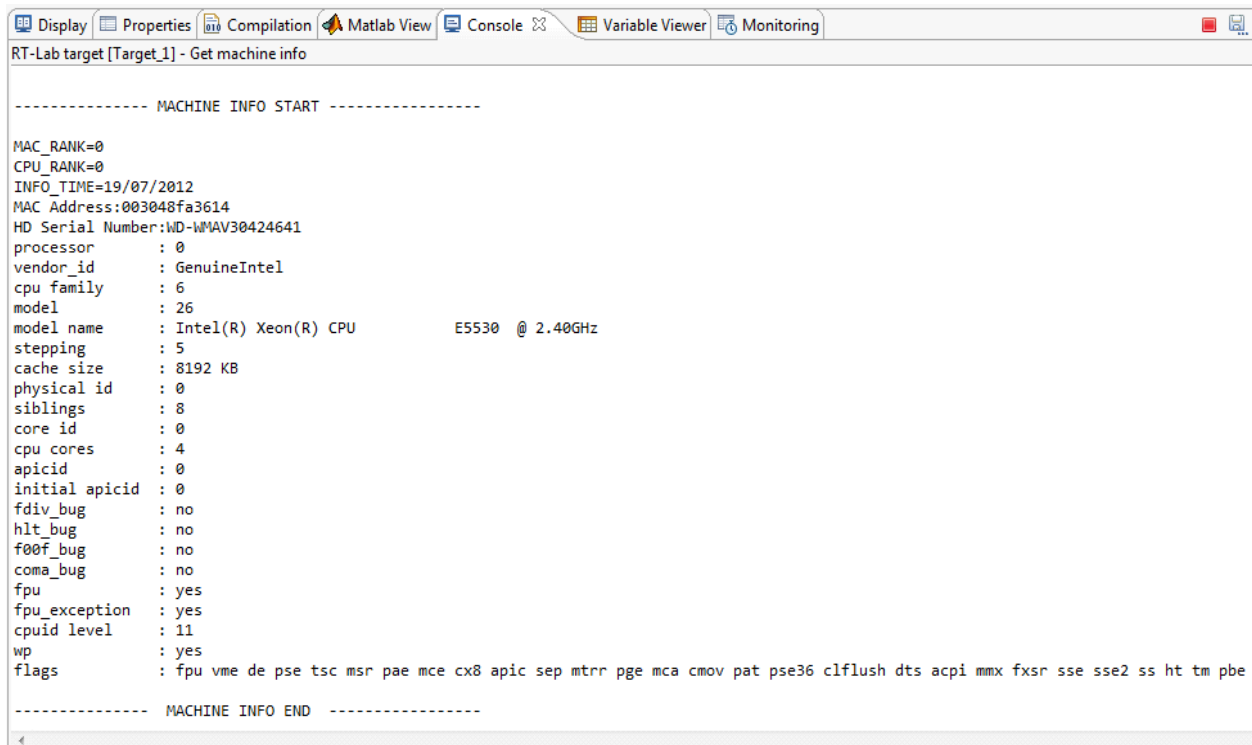
To obtain this information:

1. Follow the steps on section 4.2
2. Once completed, go to the "License" tab. In addition to a Description window, an Expiration date field and a feature table, you will have four options:
 - Retrieve license file from target:** With this option, the user can retrieve the license file from the target and it saves it on the host computer. Mainly used for debug purposes. This option works only if there is already a license on the target.
 - Send license file to target:** With this option, the user can install his license on the target. A browser opens on the host to select the license file and it is then sent and installed on the target automatically.
 - Display license info:** With this option, the user can display in a text format the information contained in the license file found on the target. Mainly used for debug purposes. The information appears in the "Console" tab of RT-LAB. This option works only if there is already a license on the target.
 - Display machine info:** With this option, the user can display in a text format the hardware information about his target. This information is used to bind the license

to the target. The information appears in the "Console" tab of RT-LAB. This option works only if RT-LAB 10.5.x+ is already installed on the target.



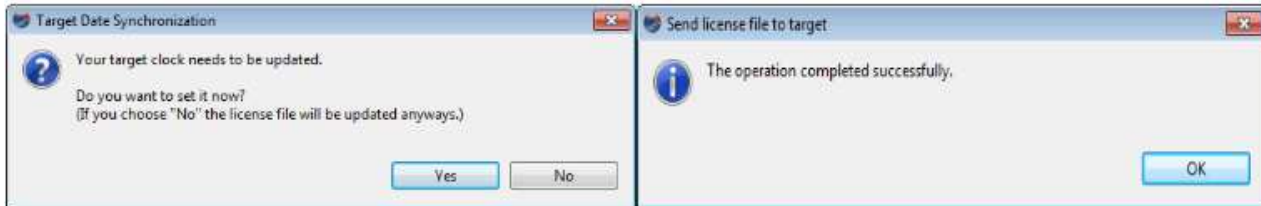
3. Do the "**Display machine info**" and save the result in a text file.
4. Once completed, go to URL: <http://www.opal-rt.com/support/license-activation-center>. An specific field allows you to add the "Display machine info" result.



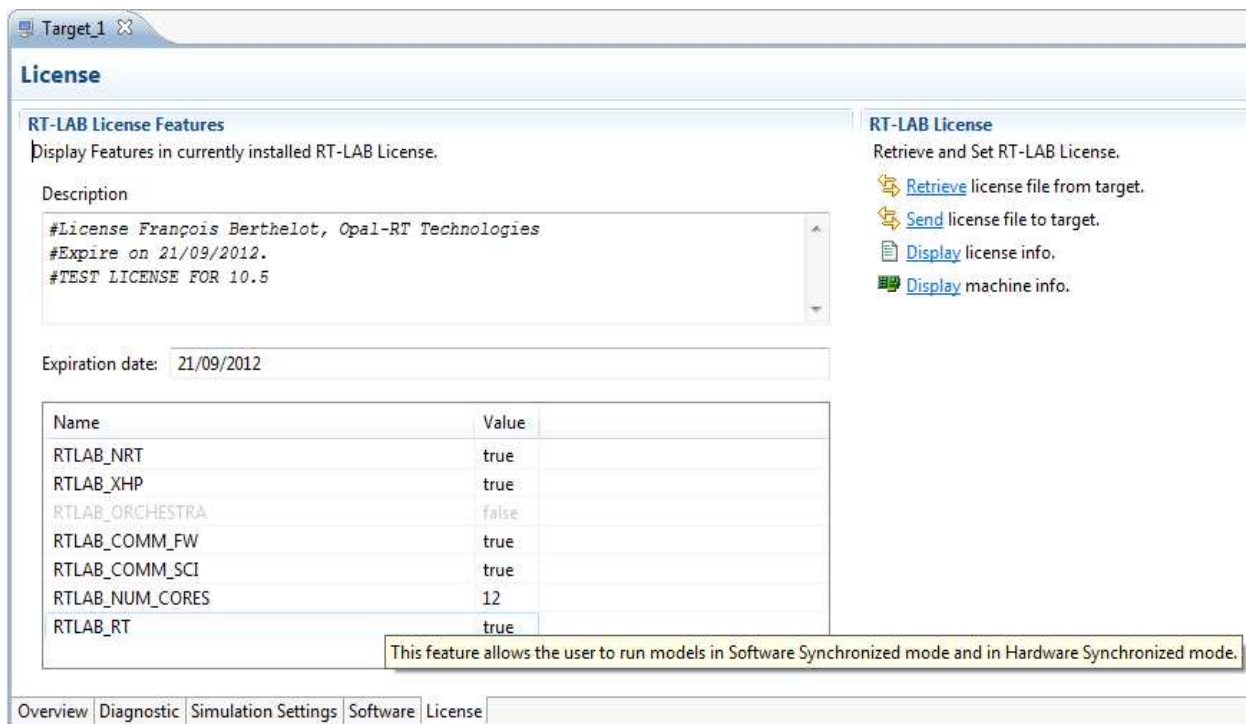
4.4 Installing the License File

To install the license file:

1. When you receive your license, do the **"Send license file to target"**. If there is a request about updating your target clock, select Yes. A pop-up with mention **"The operation completed successfully"** should appear at the end of the operation. The Description, Expiration date and feature table fields should also be filled with the license information.

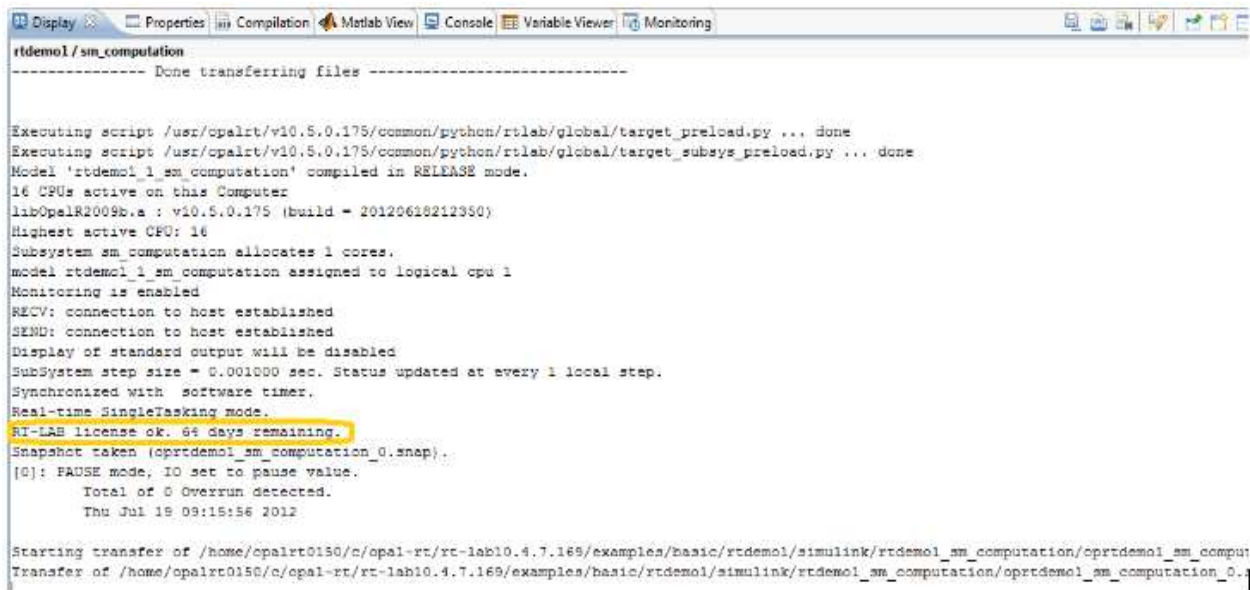


2. The "License" tab should now display the license information in the Description, Expiration date and feature table fields. Dragging your mouse on the features in the table will give you a tooltip with more information about that particular feature. If the mention "No license is available on this target!" still appears on the License tab even if the information of the license appeared, close the complete target panel and double-click again on your target in the "Targets" list of the Project Explorer



3. Validate that your license works. To do so, run a very simply example model such as rtdemo1.mdl (C:\OPAL-RT\RT-LAB\v10.5.x\Examples\Basic\rtdemo1). In RT-LAB 10.5.x+, the license check is made uniquely at load. If you can load rtdemo1, it means your license is functional. You should expect to see a line in

the load log about the remaining number of days of the license or a mention similar to "RT-LAB license ok".



```
rtdemo1 / sm_computation
----- Done transferring files -----

Executing script /usr/opalrt/v10.5.0.175/common/python/rtlab/global/target_preload.py ... done
Executing script /usr/opalrt/v10.5.0.175/common/python/rtlab/global/target_subsys_preload.py ... done
Model 'rtdemo1_1_sm_computation' compiled in RELEASE mode.
16 CPUs active on this Computer
libOpalR2009b.a : v10.5.0.175 (build = 20120616212350)
Highest active CPU: 16
Subsystem sm_computation allocates 1 cores.
model rtdemo1_1_sm_computation assigned to logical cpu 1
Monitoring is enabled
RECV: connection to host established
SEND: connection to host established
Display of standard output will be disabled
SubSystem step size = 0.001000 sec. Status updated at every 1 local step.
Synchronized with software timer.
Real-time SingleTasking mode.
RT-LAB license ok. 64 days remaining.
Snapshot taken (oprtdemo1_sm_computation_0.snap).
[0]: PAUSE mode, 10 set to pause value.
Total of 0 Overrun detected.
Thu Jul 19 09:15:56 2012

Starting transfer of /home/opalrt0150/c/opal-rt/rt-lab10.4.7.169/examples/basic/rtdemo1/simulink/rtdemo1_sm_computation/oprtdemo1_sm_compu
Transfer of /home/opalrt0150/c/opal-rt/rt-lab10.4.7.169/examples/basic/rtdemo1/simulink/rtdemo1_sm_computation/oprtdemo1_sm_computation_0..
```

IMPORTANT NOTES ABOUT LICENSES IN RT-LAB 10.5.x+:

- The license is now a binary file (.bin) instead of a ".lic" file
- The license check is made at load only. You can open RT-LAB, edit and compile your models with no license.
- The license stands 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 users, the license stands under C:\OPAL-RT\LicServer.
- Once installed, the license is located under /usr/opalrt/license/ and is renamed "opallicense.bin" no matter its initial name. If you "Retrieve license file from target", you can expect to have a .bin file named "opallicense.bin". The content is exactly the same as the initial .bin file you received from Opal-RT even if its name is different. In the same directory, there may be other files named differently. These are internal files that should not be removed nor renamed.
- The license server of RT-LAB 10.5.x+ is totally independant from the license server of RT-LAB 10.4.x and previous versions. Both license servers can cohabitate and be functional on the same hosts/targets. Be sure to update the both your host and target to either 10.5.x or to 10.4.x and earlier.
- The license server of RT-LAB 10.5.x+ is totally independant from Acresto's FLEXIm software previously used for the license management. It is now purely an Opal-RT Technologies solution.

4.5 Setting Up the Command Station for Windows Target Configurations

4.5.1 Installing Visual C++

As the model is going to be compiled on the command station with NT/2000/XP/Vista/Seven target type, you must install Microsoft Visual C++. 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.

4.5.2 Registering Variables

If Visual C++ 6.0 is already installed, there is an issue with the installer setting the environment variables for the user performing the installation instead of the whole system. To correct the situation, you need to do the following:

1. Go to **Start>Settings>Control Panel** and click on **System**. The **System Properties** window is displayed.
2. Click on the **Advanced** tab.
3. Click on **Environment Variables**. The **Environment Variables** window is displayed.
4. Add the following environment variables (append to existing values) in the **System variables** section.

Note: Please note that these values are for a default installation of Visual C++ 6.0. You may modify them according to your installation. For Visual C++ 2005, 2008 and 2010 no registering variables should be modified.

- **include** - C:\Program Files\Microsoft Visual Studio\VC98\atl\include;C:\Program Files\Microsoft Visual Studio\VC98\mfc\include;C:\Program Files\Microsoft Visual Studio\VC98\include
- **lib** - C:\Program Files\Microsoft Visual Studio\VC98\mfc\lib;C:\Program Files\Microsoft Visual Studio\VC98\lib
- **MSDevDir** - C:\Program Files\Microsoft Visual Studio\Common\MSDev98
- **Path** - C:\Program Files\Microsoft Visual Studio\Common\Tools\WinNT;C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin;C:\Program Files\Microsoft Visual Studio\Common\Tools;C:\Program Files\Microsoft Visual Studio\VC98\bin;

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

4.6 Installing updates

After the installation is completed, install the updates (if any) from the "Updates CD" or by checking on the Opal-RT website.

1. Start the executable for the update and follow the instructions. RT-LAB folder and version will be detected automatically.
2. Some updates have to be installed on target also. Refer to QNX6 or Linux chapter to read the installation procedure.

4.7 Installing the LabVIEW API

The LabVIEW API is provided as a LabVIEW library (.lib file). It contains one VI (Virtual Instrument) per API function.

This API must be installed for each version of LabVIEW you want to use and LabVIEW must be installed first.

To install the API, open the VI located at <RT-LAB installation dir>\labview\Install_API.vi with LabVIEW. The VI should run automatically. If not, press <Ctrl> + R.

Verify that the correct version is used for both LabVIEW and RT-LAB. Select the components you want to install and click OK. Note that the library is required to install the palette.

The installer will do the following (it can be done manually if needed):

1. Copy the OpalApi.llb file to <LabVIEW installation dir>\vi.lib\OpalApi\.
2. Mass compile the entire library for the current version of LabVIEW. This has no effect if LabVIEW 8.5 or 8.5.1 is used.
3. (Optional) Copy the palette to <LabVIEW installation dir>\menus\Categories\OpalApi\.

Remove the destination folders to completely uninstall the library and/or the palette.

Warning: It is strongly discourage to use the original library located at <RT-LAB installation dir>\common\bin\OpalApi.llb. Doing so would create hard-links to this file that will not be updated when you install a newer version of RT-LAB.

4.8 Installing for TestStand

During the installation process, RT-LAB will add certain files to TestStand if TestStand was installed prior to installing RT-LAB. RT-LAB does all the necessary adjustment so that most of the RT-LAB API functions may be readily used in TestStand. The supported versions of TestStand are 3.0, 3.1 and 4.1.

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 in 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.

4.9 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.

Installing RT-LAB Components on QNX 6

5.1 Installing QNX 6.3.2 or QNX 6.5

5.1.1 Installing the QNX RTOS

Insert the CD labeled "OPAL-RT - QNX 6.3.2 runtime CD" or "OPAL-RT - QNX 6.5 runtime CD" in the CD-ROM drive of the target.

Follow the QNX standard installation procedure (see Installation notes – QNX 6.3 Neutrino hosts, Installation notes – QNX 6.5 Neutrino hosts).

Note: During the installation, select the standard installation when prompted. All configuration required for using RT-LAB will be performed automatically during this installation. Once the installation has finished and the target has been rebooted, configure your network.

5.1.2 Configuring the Network

To configure the network:

1. To set your network options, click on **Network**. Now you are ready to set the following (if you do not know what to enter, consult your network administrator):
 - **Device** tab: IP address and Netmask
 - **Network** tab: Default Gateway and Name Server
2. When done, click on **OK** to apply your changes and close the window.
3. QNX is now installed on your computation node. To make this station functional you must install the RT-LAB target component on an operational QNX node.

5.1.3 Installing the SMP kernel (optional)

If the target where QNX has been installed is a multi-cpu or multi-core computer, the SMP kernel has to be installed to enable the multi-core feature of QNX. An Opal-RT SMP kernel has been installed during the CD installation.

To install the OPAL-RT SMP kernel:

1. copy the opal-rt smp kernel and replace the boot file:
 - **cp /boot/fs/smp_opal.ifs /.boot**
2. reboot the target

5.2 Installing RT-LAB on QNX 6.x Targets

For QNX 6.x, you must install RT-LAB on each target node.

Refer to [5.2.2, Installing from the RT-LAB GUI](#). The RT-LAB daemon (OpalD) has been installed automatically during the CD installation.

5.2.1 Installing manually

To install from the network:

1. On the **Command Station** double click on **My Computer**. The **My Computer** window is displayed.
2. Go to **C:\OPAL-RT\RT-LAB\vX.X.X\target\qnx**
3. Open another **My Computer** window.
4. Enter **ftp://ntuser@xxx.xxx.xxx.xxx/** in the user field, where **user** is a valid username for your qnx system (default is **ntuser**), **pass** is the corresponding password (default is **ntuser**) and **xxx.xxx.xxx.xxx** is the compilation node's IP address.
5. You are now connected by FTP to your compilation node.
6. Copy the RT-LAB installer script **opinstall** by dragging and dropping from one window to the other.
7. Copy the RT-LAB archive file **rtlab-qnx6-[version_name].tgz** by dragging and dropping from one window to the other.
8. Connect to the QNX node by telnet (i.e. **Start>Run> telnet xxx.xxx.xxx.xxx**) and login as **root**.
9. Execute the following commands to install the archive:

```
cd /  
  
# textto -l /home/ntuser/opinstall  
  
# chmod 755 /home/ntuser/opinstall  
  
# /home/ntuser/opinstall /home/ntuser/rtlab-qnx6-[version_name].tgz  
  
# rm -f /home/ntuser/rtlab-qnx6-[version_name].tgz
```

5.2.2 Installing from the RT-LAB GUI

To install from the RT-LAB GUI:

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

5.3 Installing RT-LAB updates on target

After the installation is completed, install the updates (if any).

1. Follow the same instruction described on paragraph [5.2.2, Installing from the RT-LAB GUI](#) but select the tgz file from Install/Patch popup menu for the update instead of the **rtlab-qnx6-vx.y.tgz** file.

Installing RT-LAB Components on Redhat Linux Machines

6

This section describes the installation of RT-LAB host and target components on a Redhat Linux machine.

6.1 Installing the Operating System

Follow the Redhat standard installation procedure (*See Redhat Linux Releases Notes* for installation procedures). Once the Redhat Linux OS is installed, you must configure it for using RT-LAB.

6.2 Installing RT-LAB Target Node

6.2.1 Installing from the Network

To install from the network:

2. Open My Computer window
3. Go to **C:\OPAL-RT\RT-LAB\vX.X.X.X\target\redhawk**
4. Open another **My Computer** window.
5. Enter **ftp://ntuser@xxx.xxx.xxx.xxx/** in the user field, where **user** is a valid username for your redhat system (default is **ntuser**), **pass** is the corresponding password (default is **ntuser**) and **xxx.xxx.xxx.xxx** is the compilation node's IP address.
6. You are now connected by FTP to your compilation node.
7. Copy the RT-LAB archive file **rtlab-redhawk-[version_name].rpm** by dragging and dropping from one window to the other.
8. Connect to the Redhat node by telnet (i.e. **Start>Run> telnet xxx.xxx.xxx.xxx**) and login as **root**, password **redhat**
9. Type the following command to install RT-LAB:
rpm -ivh /home/ntuser/rtlab-redhawk-version_name.rpm.

6.2.2 Installing from the RT-LAB GUI

To install from the RT-LAB GUI:

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

6.3 Installing RT-LAB updates

After the installation is completed, install the updates (if any).

1. Follow the same instruction described on paragraph [6.2.2, Installing from the RT-LAB GUI](#) but select the rpm file from Install/Patch popup menu for the update instead of the **rtlab-redhawk-vx.y.rpm** file.

6.4 Uninstalling RT-LAB

To uninstall RT-LAB:

1. Open a shell terminal.
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 in a shell terminal: **# rpm -qa | grep rtlab.**