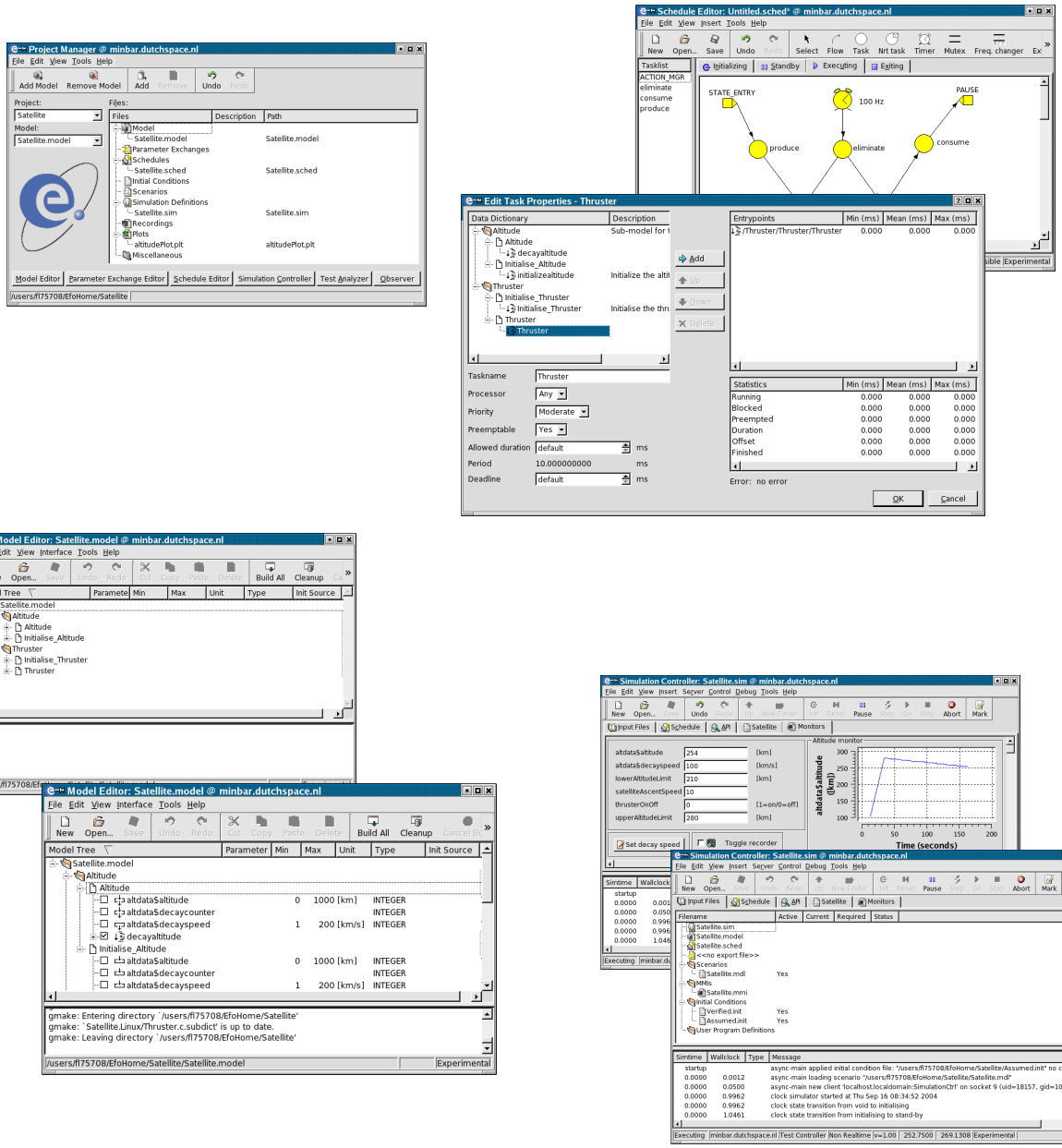
EuroSim Mk5.2
Owner's Manual



National Aerospace Laboratory NLR

Dutch Space
an EADS Astrium company

NSPYRE
making technology matter

Summary

EuroSim is an engineering simulator to support the design, development and verification of (sub) systems defined by programmes of various scales. The facility provides a reconfigurable real-time execution environment with the possibility of man-in-the-loop and/or hardware-in-the-loop additions.
This document specifies the install procedure and hints to maintain the EuroSim installation.

Contents

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to specify the procedure to install EuroSim on a target system and to provide guidelines that help the EuroSim facility manager to maintain EuroSim in an operational state.

1.2 Audience

This document is written for the EuroSim facility manager. The facility manager will be considered the owner of the EuroSim installation, and hence will also be referred to—for brevity—as *owner*. It is assumed that the facility manager is also the (computer) system administrator and thus has administrator (root) privileges.

1.3 Scope

This document applies to EuroSim Mk5.2.1. It contains information on how to install EuroSim and to keep it in an operational state. For information on how to use EuroSim, the reader is referred to the *EuroSim Software User Manual* [SUM12]. For information pertaining to the latest release of EuroSim, please consult the *EuroSim Software Release Notes* [SRN11].

1.4 Document Structure

This document is laid-out as follows:

Introduction
: Purpose of present document and references to other documents.

EuroSim Overview
: Short overview of EuroSim in terms of environment variables, processes, files and directories that might help the owner to maintain the installation. It is not intended to provide a full file list, but only a description of the more important files and their use.

System Requirements
: Prerequisites for the EuroSim installation: machine types, disk space, other software required, saving user data.

Installation
: Procedure on how to install and customize EuroSim.

Verification
: Procedure on how to verify correct installation.

EuroSim projects and users
: Some example strategies will be provided how to set up EuroSim projects, users and repositories.

Trouble Shooting
: Suggestions for solutions in case of problems (log files, FAQ, SPR, helpdesk).

1.5 Notation

Document references are given with a mnemonic between square brackets, like [SUM12]; these mnemonics are listed in the bibliography on page 49.
EuroSim versions are generically stated as M.n.p where M refers to the Major version, m to the minor version and p to the patchlevel version.
For the procedures described in this document, the following notation is used:

Type `string`	Type 'string' at UNIX shell prompt using keyboard.
Choose *a:b:c*	Choose menu item 'c' from sub-menu 'b' from menubar option 'a', using mouse.
Select **item**	Select 'item' (usually an icon) using mouse.
Press **button**	Press button called 'button' with mouse.
`Literal text`	Literal text is written in Courier (monospaced) font.

1.6 Abbreviations

Abbreviation	Description
ANSI	American National Standardization Institute
API	Application Programming Interface
APT	Advanced Package Tool
ASAP	As Soon As Possible
BV	Besloten Vennootschap
EFO	EuroSim Follow-On
ESA	European Space Agency
ESTEC	European Space Technology Centre
F77	Fortran 77
FAQ	Frequently Asked Questions
FORTRAN	Formula Translator
DS	Dutch Space
GNAT	GNU Ada Translator
GNU	Not UNIX
GUI	Graphical User Interface
HTML	Hyper Text Mark-up Language
IVP	Installation Verification Procedure
MB	Mouse Button
MB1	Left Mouse Button

Abbreviation	Description
MB2	Middle Mouse Button
MB3	Right Mouse Button
MIF	Maker Interchange Format
NIVR	Nederlands Instituut voor Vliegtuig en Ruimtevaartontwikkeling
NLR	Nationaal Lucht- en Ruimtevaart Laboratorium
PSS05	ESA's Software Engineering Standard
RAM	Random Access Memory
RCS	Revision Control System
ROM	Rough Order of Magnitude
SGI	Silicon Graphics Inc.
SPR	Software Problem Report
SR	System Requirement
SRN	Software Release Note
SUM	Software User Manual
SWRB	SoftWare Review Board
TBW	To Be Written
TC	Tele Command
TM	Telemetry
TN	Technical Note

Chapter 2

EuroSim Overview

The purpose of this section is to provide some background information on EuroSim, that might help you in maintaining the installed product.

2.1 EuroSim tools, models and libraries

The EuroSim product consists of a number of software *tools* and one or more software *libraries*. A software tool is an application, that runs on a host computer and assists the user in performing some task. Typical examples of EuroSim tools are the *Model Editor* and the *Simulation Controller*. All EuroSim tools are equipped with a modern Graphical User Interface.
The purpose of EuroSim is to assist the user in preparing simulators, running (real-time) simulations and analyzing the results. A simulator typically consists of a model, that makes the simulator specific for some application and an invariant part, that is the same for each simulator. The invariant part consists of the EuroSim tools, and one or more EuroSim libraries. These libraries provide a whole suite of functionality.

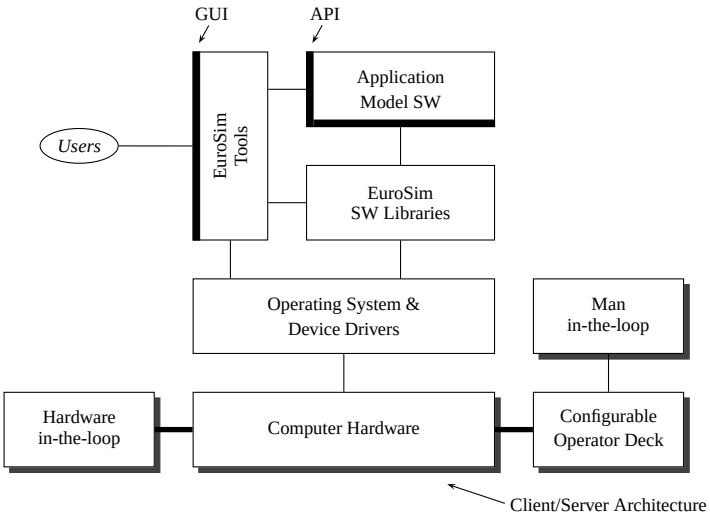


Figure 2.1: General lay-out of EuroSim: tools, model, libraries

There is a standard library called `esim`, that provides basic functionality, for example to look up the simulation time. Other libraries provide functionality for e.g. RS232, MIL1553, TM/TC interfaces.

2.2 EuroSim users and projects

The EuroSim *users* are the people on your system, using the EuroSim tooling to prepare, run and analyze simulations. Users can cooperate in a EuroSim simulation *project*. A EuroSim project consists of:

- A description.
- A directory where the files reside (also called the project root).
- A repository where the versioned files reside (for configuration control).
- A default model.

Project definitions can be kept in two places:

1. In a local EuroSim project description file. By setting the environment variable `EFO_HOME`, a user can override the default EuroSim project file in `$HOME/.eurosim`. A local project file (and there can be many of them) and the associated projects are managed by the user (see section 6.2). He/she can define projects at a local level. Projects and users are protected against each other by the normal OS protection mechanisms, which are respected by the EuroSim tooling.
2. In a system wide EuroSim project description file. This file is managed by you, the EuroSim owner (also called facility manager). Users must set the `EFO_HOME` environment variable to the directory with the system wide project file. You can use this file to create EuroSim projects and allocate a working directory (i.e. disk storage) to them.

2.3 EuroSim licenses and daemon `esimd`

EuroSim simulators are started via the EuroSim daemon[1] `esimd`. Through this daemon it is possible to run a simulation not only on the local machine, but also on any other machine in the network that has a EuroSim daemon running[2]. To prepare a host computer for the purpose of real-time simulations, root privileges are required: hence the EuroSim daemon will normally run with root privileges. Although the daemon is started as root, and the simulation processes starts off being owned by root when running real-time, this only takes as long as is needed to set things up and will never run user code (models) with root priviliges. Non-real-time simulations are never run as root at all. The root privileges are needed to isolate processors (daemon) and to run on these isolated processors (simulator).
The EuroSim daemon also serves as a *license manager*. The EuroSim daemon can only run on hosts listed in the EuroSim *license file*. All EuroSim tools will contact the EuroSim daemon to check whether their host is listed in the license file.
Four different license types are available for EuroSim:

Server License **(Server)**
: This is a realtime license, that allows multiple users to work concurrently with the EuroSim functionality for model development, test preparation and execution (i.e. simulation) and test analysis on a single computer with EuroSim installed.

Workstation license **(Workstation)**
: This is a realtime license that allows a single user to use the EuroSim functionality for model development, test preparation and execution and test analysis on a computer with EuroSim installed. The license is granted to the first user requesting the license and will only be released if the EuroSim daemon is restarted.

Desktop license **(Desktop)**
: This is a non-realtime license that allows a single user to use the EuroSim functionality for model development, test preparation and execution (i.e non-real-time simulation) and test analysis on a computer with EuroSim installed. The license is granted to the first user requesting the license and will only be released if the EuroSim daemon is restarted.

[1] Under Windows, esimd is run as a service.
[2] Provided—of course—that that machine has equal access to your simulator files as the local machine, e.g. via NFS.

Runtime license **(Runtime)**
This is a deployment license. It allows the user to execute hard-real-time simulations, but does not permite the use of the development tools to build simulators.

Capability	Server	Workstation	Desktop	Runtime
Users working concurrently	√			
Hard Real-time executions	√	√		√
Model development	√	√	√	
Test preparation	√	√	√	√
Test analysis	√	√	√	

Table 2.1: EuroSim licenses overview

Your type of license is indicated on the EuroSim license agreement.

2.4 Lay-out of EuroSim installation

The directory structure under `/usr/EuroSim` (Windows: `C:\EuroSim`) is laid-out as follows.

Directory[3]	Contents	Typical/important file	Description
`adainclude`	Package specifications for Ada.	`esim.ads`	Ada package specification for `esim` library.
`bin`	All the EuroSim executables.	`esim`	EuroSim entry point.
`etc`	Miscellaneous files.	`user.sh`	Script that defines the environment variables needed.
		`EuroSim.licenses`	License file.
		`SoftwareReleaseNote`	Software release note [SRN11].
`include`	The include files for C and Fortran.	`esim.h`	C include for basic EuroSim functions.
`lib`	All EuroSim tcl, perl, python and java libraries. EuroSim shared object libraries (Linux).	`tcl, perl, python, java`	Tcl, perl, python and java EuroSim support.
`libes.so`	Library file of basic EuroSim functions.		
`doc`[4]	EuroSim html and pdf documentation.	`OM.pdf, SUM.pdf`	Owner's Manual and Software User's Manual.

Table 2.2: Lay-out of EuroSim installation (`/usr/EuroSim` directory)

[3] Starting at `/usr/EuroSim`
[4] Only on Linux, Windows documentation online

Directory[3]	Contents	Typical/important file	Description
`man`	The EuroSim manual pages.	`man1/esim.1`	Manual page for `esim` tool.
		`man3/esim.3`	Manual page for `esim` library.

Table 2.2: Lay-out of EuroSim installation (`/usr/EuroSim` directory)

2.5 EuroSim environment variables

For its correct operation, EuroSim depends on a number of environment variables. The most important one is `$EFOROOT`, which has to point to where the EuroSim files are installed (i.e. `/usr/EuroSim`). `EFOROOT` and related environment variables are set automatically by EuroSim when using Linux and Windows.

Name	Purpose
`EFO_HOME`	Optional. Points to the location of the EuroSim project file. If not set, then `$HOME/.eurosim` is used.
`EFO_DOC`	Windows Only. Points to the location on the EuroSim website where the documentation is stored that is opened on help. It is possible to make a local version if internet access is not available, please contact the helpdesk.
`MANPATH`	This variable is extended with `$EFOROOT/man`, so that the EuroSim manual pages can be found.
`PATH`	This variable is extended with `$EFOROOT/bin`, so that the EuroSim executables can be found.

Table 2.3: Environment variables needed before starting EuroSim

When the above environment variables are set, EuroSim can be started by typing `esim` on the command line. After selecting a project and role, some more environment variables are set automatically by EuroSim for that session. A complete list is given below for information.

Name	Purpose/meaning
`PROJECTHOME`	Home directory of the current project.
`EFO_XTERM`	The terminal emulation program (if not set `xterm` is used).
`EFO_VC`	The current configuration control system. Possible values are `cadese` or `cvs`. If not set, then no configuration control is used.
`EFO_SHAREDMEMSIZE`	[Deprecated] The shared memory size used by the simulator. By default 4 MB is reserved. If more is needed then this environment variable should be set with the required memory size in bytes. This option is now controlled through settings in the model editor (Configuration tab of Tools:Set Build Options... dialog box).
`EFO_STACKSIZE`	[Deprecated] The stack size reserved for each thread of the simulator. By default 16k (Linux) is reserved. If more is needed then this environment variable should be set with the required stack size in bytes. This option is now controlled through settings in the model editor (Configuration tab of Tools:Set Build Options... dialog box).

Table 2.4: Environment variables used within EuroSim

2.6 EuroSim user's files

EuroSim will create and/or look for certain files in the user's home and/or current working directory; details are listed below.

Directory	Filename	Purpose
`$HOME/.eurosim`	`projectmanagerrc`	Contains the user preferences and GUI settings for the Project Manager.
	`modeleditorrc`	Contains the user preferences and GUI settings for the Model Editor.
	`scheduleeditorrc`	Contains the user preferences and GUI settings for the Schedule Editor.
	`simulationctrlrc`	Contains the user preferences and GUI settings for the Simulation Controller.
	`testanalyzerrc`	Contains the user preferences and GUI settings for the Test Analyzer.
	`modeldescriptioneditorrc`	Contains the user preferences and GUI settings for the Model Description Editor.
	`parameterexchangeeditorrc`	Contains the user preferences and GUI settings for the Parameter Exchange Editor.
	`calibrationeditorrc`	Contains the user preferences and GUI settings for the Calibration Editor.
	`smp2editorrc`	Contains the user preferences and GUI settings for the SMP2 Editor.
	`modelmakerc`	Contains the user preferences for the ModelMake tool.
`$EFO_HOME`	`projects.db`	By setting `$EFO_HOME` to the directory where `projects.sdb` resides, one can override EuroSim's default project description file.
`$PROJECTHOME`	`.cadeserc`	This file defines the location of the project's repository for the Cadese tooling.

Table 2.5: EuroSim user files

Chapter 3

System requirements

This section lists the requirements on software and computer that must be satisfied for EuroSim installation. Some requirements are mandatory, while other are optional, depending on the user's requirements for the related EuroSim functionality. Optional requirements are clearly marked as such.

3.1 Processor Architecture

Starting with EuroSim 5.2, EuroSim is available for both 64-bit and 32-bit Intel architecture (`x64_86`). For Linux platforms EuroSim is delivered as a 64 bit application for the 64-bit OS. For Windows 7, EuroSim is is delivered as a 32-bit application that can run on either 32-bit or 64-bit OS releases.

3.2 EuroSim for Linux

3.2.1 Supported distributions

3.2.1.1 Full installation

For the following distributions the full EuroSim tool suite is available.

Distribution	Abbreviation	Architecture
RedHawk Linux 5.4	redhawk54	i686
RedHawk Linux 6.0	redhawk60	i686
RedHawk Linux 6.3.5	redhawk63	i686 and `x86_64`
RedHat Enterprise Linux 6.3 (and rebuilds from Scientific Linux or Centos)	rhel63	i686 + `x86_64`

Table 3.1: Supported Linux distributions

The End of Support date indicates at which date the EuroSim consortium plans on stopping actively supporting a particular distribution. Customers with a maintenance contract will keep support for their selected operating system as long as their maintenance contract is running and the distribution is maintained by its supplier.However this is in pricinple restricted to the Minor release version of EuroSim (e.g. Mk4.4) that is agreed at the start of the agreement. Upgrades to newer minor and major releases with support for an older distribution may be possible, but is at the discretion of the EuroSim helpdesk.
As shown in above table the currently support linux distributions have a limited end date. The reason is that the /esimmk release will be followed very short term with a next minor release on which an upgrade is provided to Redhat/Redhawk 6.3 including 64 bit release which will be our new long term support baseline.

3.2.1.2 Client only installation

For the regular and some older distributions client only installations are provided via the EuroSim website Support pages. Please check www.eurosim.nl for the available client software.
In case of a tool from a different vendor that is interoperable with EuroSim, the vendor has most likely already packaged the client library with its product. The vendor will then specify the versions of the EuroSim server that his product supports.
If you are a tool vendor and want to package the client with your tool then the webpage with the client libraries states the legal basis on which the consortium supports that. If the client library that you seek is not there, then please contact the EuroSim helpdesk.

3.2.2 Requirements

All requirements on other software packages are specified in the EuroSim RPM packages.

ReqID	Requirement	Remarks
IR–1	Host computer shall be an Intel Pentium compatible PC or better. For guanteed hard realtime usage we recommend using quadcore processors. For hard realtime usage we recommend procuring a an integrated system via the EuroSim consortium.	
IR–2	Linux distribution shall be one of the distributions mentioned in subsection 3.2.1	
IR–3	Hyperthreading shall be turned off in the BIOS when hard realtime simulation is required.	Needed to eliminate severe cause of clock jitter when running real-time.
IR–4	Host RAM shall be 512MB or more. At least 4GB recommended	
IR–5	Free disk space shall be at least 100 MB, At least 1GB recommended.	Needed for EuroSim installation.
IR–6	Basic development system (C compiler, linker and make) shall be available.	Not required for run-time license.
IR–7	C++ compiler shall be available.	Part of Linux distribution.
IR–8	Fortran compiler shall be available.	Part of Linux distribution. Only required when using Fortran sub-models.
IR–9	Ada compiler shall be available.	Part of Linux distribution. Only required when using Ada sub-models.
IR–10	RCS shall be available.	Part of Linux distribution. Only required when using configuration control with the cadese backend.
IR–11	CVS shall be available.	Part of Linux distribution. Only required when using configuration control with the cvs backend.
IR–12	Perl 5 shall be available.	Part of Linux distribution. Only required when using the batch utility (in package perl-EuroSim).

ReqID	Requirement	Remarks
IR–13	Readline shall be available.	Part of Linux distribution. Only required when using the batch utility.
IR–14	PV-Wave, version 6.x shall be available.	Only required when using EuroSim's test analysis functionality with the PV-Wave back end.
IR–15	Gnuplot shall be available.	Part of Linux distribution. Only required when using EuroSim's test analysis functionality with the gnuplot back end.
IR–16	Firefox shall be available.	Part of Linux distribution. Only required for on-line availability of the EuroSimSUM.

3.2.3 Limitations

Hard realtime execution is presently only guaranteed on the Concurrent iHawk computer with the Redhawk Linux distrution and on Rehard Enterprise Linux with the MRG kernel installed (64-bit only). On standard Redhat Enterprise distributions, EuroSim will make a best effort, but the realtime quality can not be guaranteed.

3.3 EuroSim for Windows

3.3.1 List of supported Windows distributions

3.3.1.1 Full installation

For the following distributions the full EuroSim for Windows tool suite is available.

Distribution	Abbreviation	Architecture	End Of Support
Windows 7	win7	i686 and `x86_64`	MicroSoft defined

Table 3.3: Supported distributions

The End of Support date indicates at which date the EuroSim consortium plans on stopping actively supporting a particular distribution. Customers with a maintenance contract will keep support for their selected operating system as long as their maintenance contract is running and the distribution is maintained by its supplier.However this is in pricinple restricted to the Minor release version of EuroSim (e.g. Mk5.x) that is agreed at the start of the agreement. Upgrades to newer minor and major releases with support for an older distribution may be possible, but is at the discretion of the EuroSim helpdesk.
As shown in above table the support for Windows XP has expired, which is a result of the expiration of support by Microsoft for Windows XP. Customers are advised to move to the new Windows7 distribution. The support for Windows 7 will remain as long as Microsoft supports the Windows7 operating system. The EuroSim application on Windows 7 is a 32-bit application but has been verified to work on 64 bit releases as well without any special measured needed.

3.3.1.2 Client only installation

Client libraries for windows to operate over a network with linux distributions are available and downloadable from the EuroSim website Support pages. Please check www.eurosim.nl for the available client software.
In case of a tool from a different vendor that is interoperable with EuroSim, the vendor has most likely already packaged the client library with its product. The vendor will then specify the versions of the

EuroSim server that his product supports.
If you are a tool vendor and want to package the client with your tool then the webpage with the client libraries states the legal basis on which the consortium supports that. If the client library that you seek is not there, then please contact the EuroSim helpdesk.

3.3.2 Requirements

The EuroSim for Windows release is provides as an installer. Before this can be installed, the user must however first install the MinGW compiler suite version 4.8.1 or higher with MSYS support, as well as GnuPlot. For the MinGW/MSYS compiler suite and utilities, a convenience installer is provided which will install a duplicate of what has been used by the EuroSim Developers environment. This also includes utilities like Tcl/Tk, Perl, Vim, Bash etc If the user prefers it is also possible to install the MinGW environment manually. More information is provided in the installation section. For GnuPlot the installer as avilable on the internet is provided as convenience.

ReqID	Requirement	Remarks
IR–1	Host computer shall be an Intel Core 2 compatible PC or better.	
IR–2	Operating System shall be Windows 7, either 32 bits or 64 bits	
IR–3	Host RAM shall be 2GB MB or more.	
IR–4	2Gb diskspace shall be avialable	Needed for EuroSim and MinGW installation.

3.3.3 Limitations

Note that the capabilities of EuroSim for Windows are limited with respect to the Linux release. In particular there is no support for Perl, and at most soft realtime execution can be achieved, hence for Windows 7 releases, only desktop licenses are available

Chapter 4

Installation & Customization

This section lists—step by step—the procedure to install EuroSim on a computer system. This procedure can be used for a first time installation, as well as for any subsequent installations, e.g. installing a new version.

IMPORTANT NOTICE:

You might want to save the license file `/usr/EuroSim/etc/EuroSim.licenses` from an existing EuroSim installation before installing a new version.

4.1 Installation procedure for Linux

4.1.1 Linux installation steps

This installation procedure is for the supported red hat based distributions, which use the rpm solution for package installation.
In this section the notation `M.m.p-x.rhel6.i686.rpm` identifies a packages for EuroSim Major version M, minor version m, patchlevel p, with x being a generation sequence number (irrelevant to the user) and rhel6 defining the RedHat Enterprise Linux 6 release for processor architecture i686.
Instead of rhel6 you may also encounter redhawk6 for the concurrent realtime version of red hat.

Step	Description
4.1–1	Copy the download section URL provided to you by the EuroSim helpdesk in your favorite internet browser and press enter. The link will look like http://www.eurosim.nl/software-download/¡YOUR SECTION¿.
4.1–2	Enter your username and password in the authorization dialog to log in to your download section on the EuroSim website
4.1–3	You will see EuroSim mark versions with major and minor version numbers of the EuroSim release(s) that are made available to you. Enter the version that you want to install, select the patch level directory and then go to the rhel or redhawk directory in line with the desired distribution.

4.1–4	download the following packages to you computer.
	
	`required:`
	`- EuroSim-M.m.p-x.distro.arch.rpm`
	
	`optional:`
	`- EuroSim-TSP-M.m.p-x.distro.arch.rpm`
	`- EuroSim-client-M.m.p-x.distro.arch.rpm`
	`- EuroSim-WebInterface-Monitor-M.m.p-x.distro.arch.rpm`
	`- EuroSim-WebInterface-Server-M.m.p-x.distro.arch.rpm`
	`- EuroSim-client-M.m.p-x.distro.arch.rpm`
	`- EuroSim-java-M.m.p-x.distro.arch.rpm`
	`- EuroSim-java-client-M.m.p-x.distro.arch.rpm`
	`- EuroSim-python-client-M.m.p-x.distro.arch.rpm`
	`- EuroSim-tcl-client-M.m.p-x.distro.arch.rpm`
	`- perl-EuroSim-python-M.m.p-x.distro.arch.rpm`
	
	Depending on what is already installed on your system, you may also need other packages on which EuroSim depends. These can be found in the default repositories that come with your Linux distribution supplemented with the EPEL (Extra Packages for Enterprise Linux). This repository can be activated by installing a small RPM: epel-release. This RPM will add the epel.repo file to /etc/yum.repos.d/. The latest version of the epel-release RPM can be downloaded from http://fedoraproject.org/wiki/EPEL.
	
	`rpm --install epel-release-6-8.noarch.rpm`
	`yum repolist`
	
	Yum should list the epel repository, the default repositories of your Linux distribution and, optionally, the Concurrent repositories.

Step	Description
4.1–5	Install EuroSim `yum install EuroSim-M.m-p-x.rhel6.i686.rpm` If you only need the client libraries, then instead of installing the `EuroSim-M.m-p-x.rhel6.i686.rpm` you can install `EuroSim-client-M.m-p-x.rhel6.i686.rpm`. This EuroSim client package includes all client libraries provided by the EuroSim-client package so you cannot install both.
4.1–6	Optionally install EuroSim perl batch utility `yum install perl-*.rpm` This package requires the EuroSim client libraries.
4.1–7	Optionally install EuroSim python batch utility `yum install EuroSim-python-client-M.m-p.rhel6.i686.rpm` This package requires the EuroSim client libraries.

4.1–8	Optionally install EuroSim java batch utility
	`yum install EuroSim-java-client-M.m-p.rhel6.i686.rpm`
	This package requires the EuroSim client libraries and the Java Runtime Environment. At the time of writing, this package requires java-1.6.0-openjdk but in fact any JRE with version 1.5 or higher will do, for example java-1.5.0-sun. Therefore, yum may report an error about an unresolved dependency. In that case, make sure that a recent JRE is installed on your system by typing
	`yum list installed 'java*'`
	and then install the EuroSim Java package with rpm instead of yum:
	`rpm -Uvh --nodeps EuroSim-java-client-M.m-p.rhel6.i686.rpm`

4.1–9	Optionally install EuroSim tcl batch utility `yum install EuroSim-tcl-client-M.m-p.rhel6.i686.rpm` This package requires the EuroSim client libraries.
4.1–10	Optionally install Java model support `yum install EuroSim-java-M.m-p.rhel6.i686.rpm` This package requires the Java Development Kit. See also the instructions for the EuroSim-java-client package.

Step	Description
4.1–11	Optionally install TSP provider support `yum install EuroSim-TSP-M.m-p.rhel6.i686.rpm` TSP is the Transport Sample Protocol. The package tsp is required by EuroSim-TSP, which is not available in any public repositories. If you do not have tsp, do not install EuroSim-TSP.
4.1–12	Optionally install the Web Interface Server. `yum install EuroSim-WebInterface-Server-M.m-p.rhel6.i686.rpm`
4.1–13	Optionally install the Web Interface Monitor. `yum install EuroSim-WebInterface-Monitor-M.m-p.rhel6.i686.rpm`

Your installation for linux is now completed.

4.1.2 Starting the EuroSim daemon under Linux

A license key is required to start the EuroSim daemon. Normally, the key for your system will already be included in the license file `/usr/EuroSim/etc/EuroSim.licenses`.
If you need a new licenses key, you can mail that request to esim-info@dutchspace.nl, providing the following info:

1. The reference number of your EuroSim license agreement.
2. Your computer's name, type and OS version.
3. The first line of the output of the `sysinfo` command. The information produced uses the MAC address of the first Ethernet card present in the system. If there is no Ethernet card present in your system, you cannot get a license for EuroSim.

Now, proceed as follows (still as user 'root', and assuming this is a first-time installation).

Step	Description
4.1–14	Append your license key (a string starting with `EuroSim`) to the license file: `cat >> /usr/EuroSim/etc/EuroSim.licenses` Usually this step is not needed.

4.1–15	Start the daemon
	`# /etc/init.d/esim start`
	To stop the daemon again you should do:
	`# /etc/init.d/esim stop`
	Restarting the EuroSim daemon is automatically done during the installation of EuroSim.

When only an EuroSim update has been installed, it is wise to stop and start the EuroSim daemon (step 4.1–15), to make sure that the running `esimd` is compatible with the new version of EuroSim. On installation or upgrade the EuroSim daemon is automatically restarted.

As delivered, the EuroSim daemon will write simulator messages[1] to `esimd.log` in directory `/var/log`. You can configure that by editing the file `/etc/init.d/esim`. By leaving out the `-l` switch altogether, the EuroSim daemon will write its messages to the system log file `/var/log/messages`. Not all messages are logged however. If you want to see all messages you must make a minor modification to the file `/etc/syslog.conf`. Just add `;local0.*` to the list of filters which write to `/var/log/messages`. The line would then look something like:

```
*.info;mail.none;authpriv.none;cron.none;local0.* /var/log/messages
```

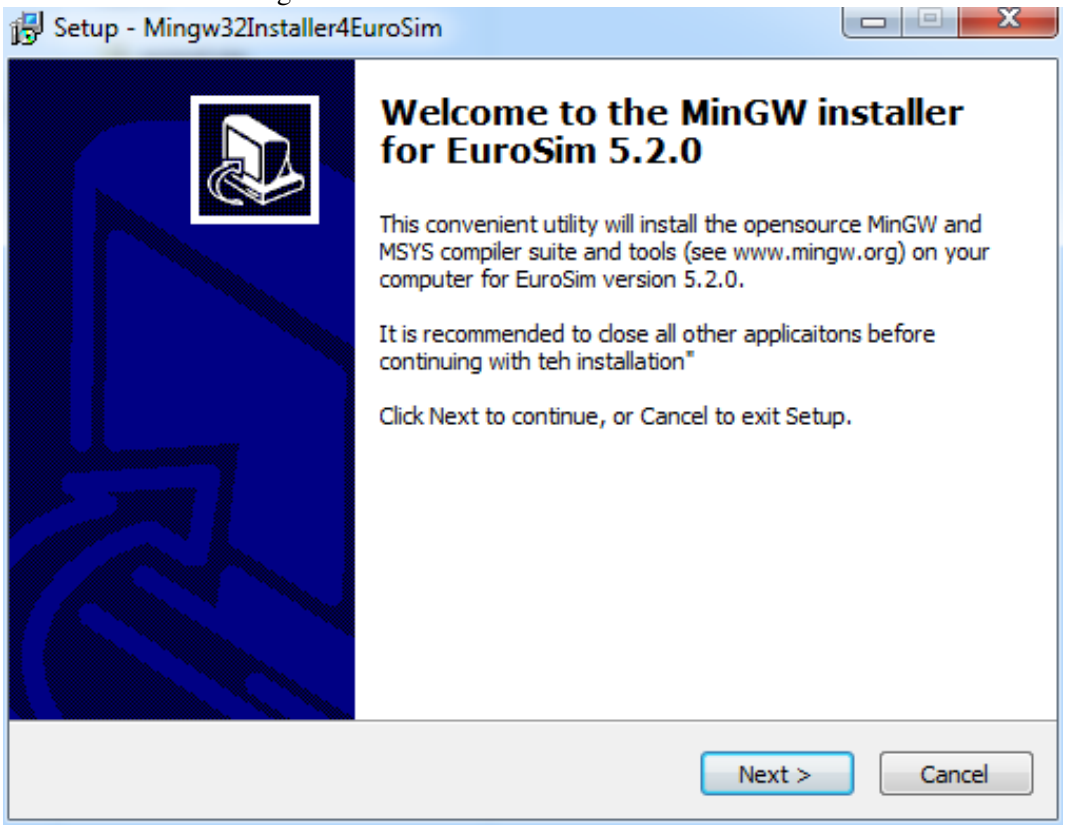
The EuroSim daemon log file is automatically cycled every week or when the size grows bigger than 10 MB. The configuration file can be found in `/etc/logrotate.d/esimd`. The script will keep 5 older log files.
For more information on the EuroSim daemon, see `man esimd`.

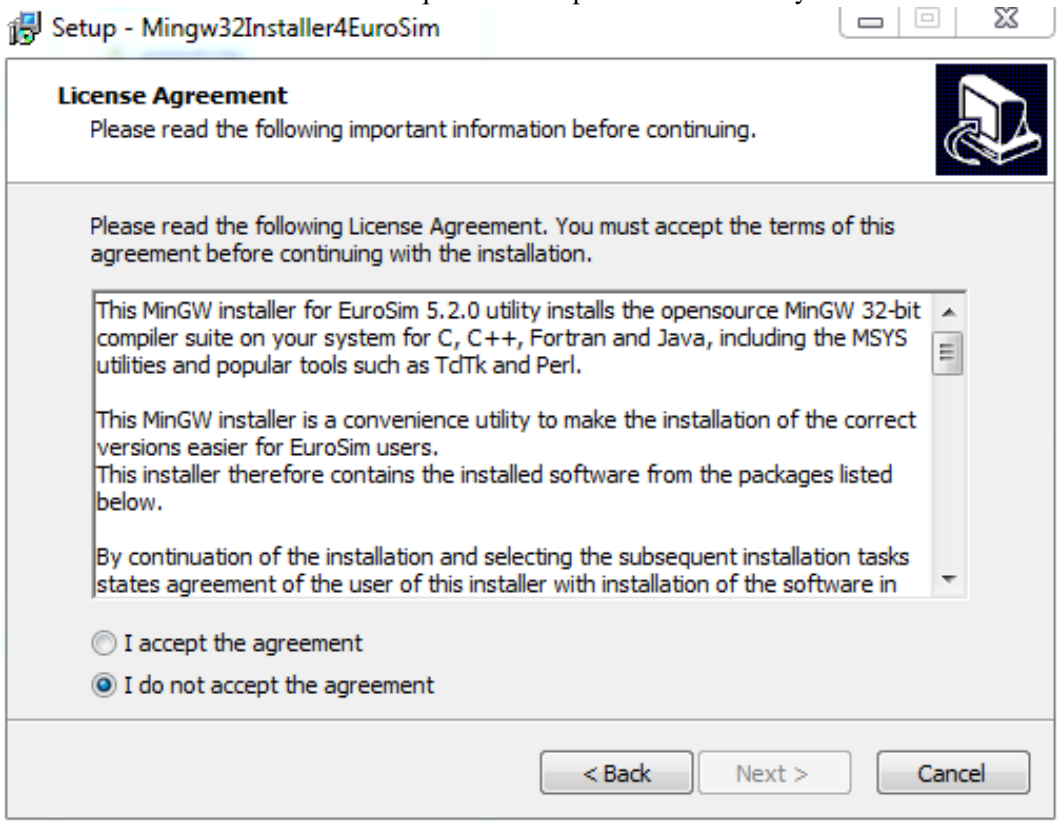
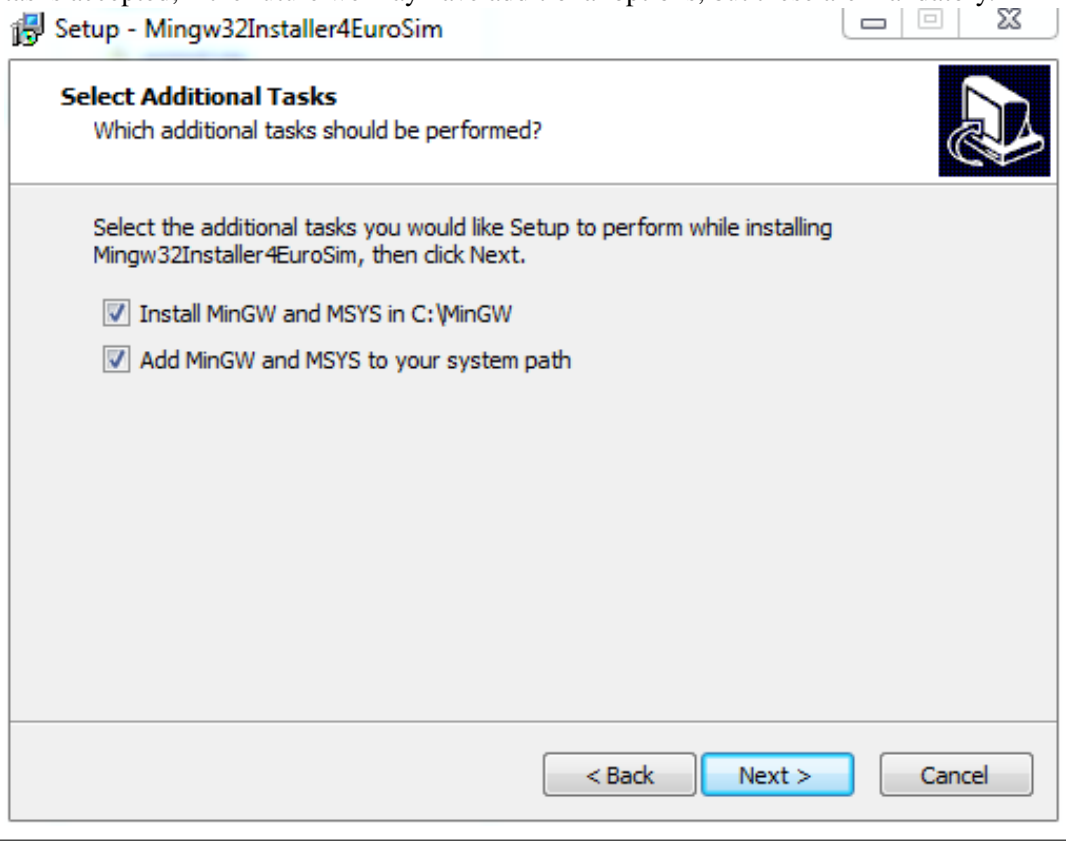
[1] The daemon writes its own messages to syslog, and routes the messages from started simulators to the indicated log file. So, what is in the log file, is in fact, output from EuroSim simulators.

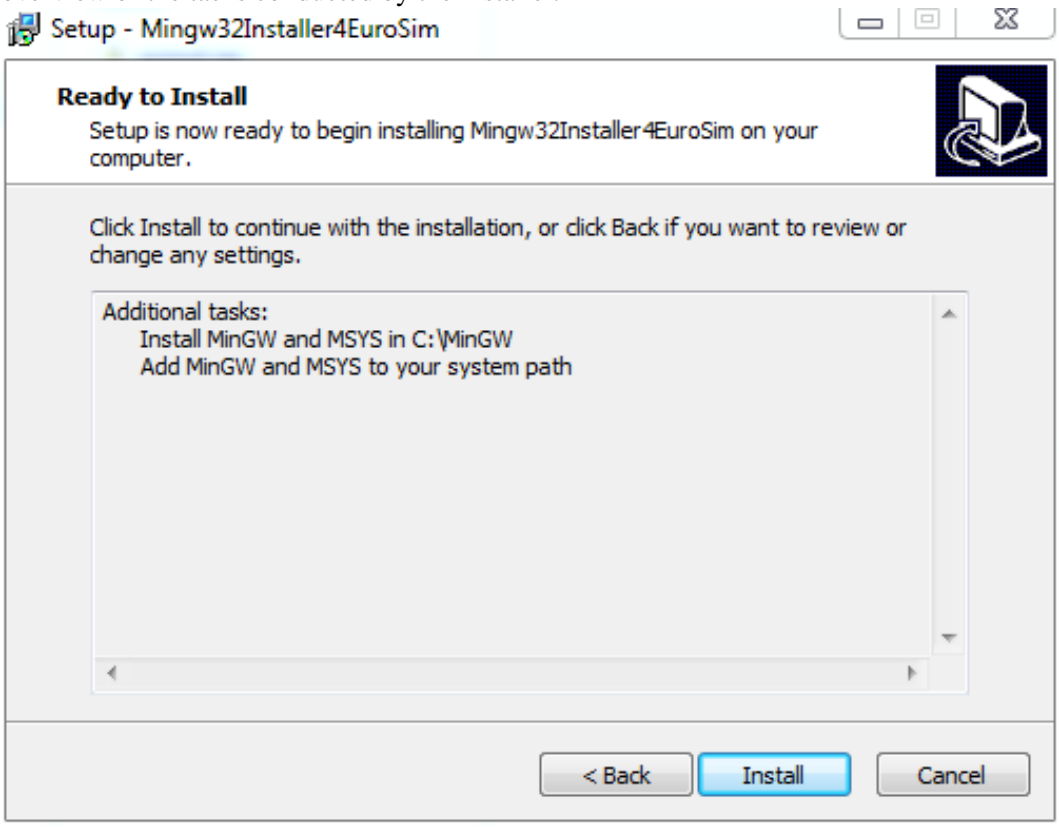
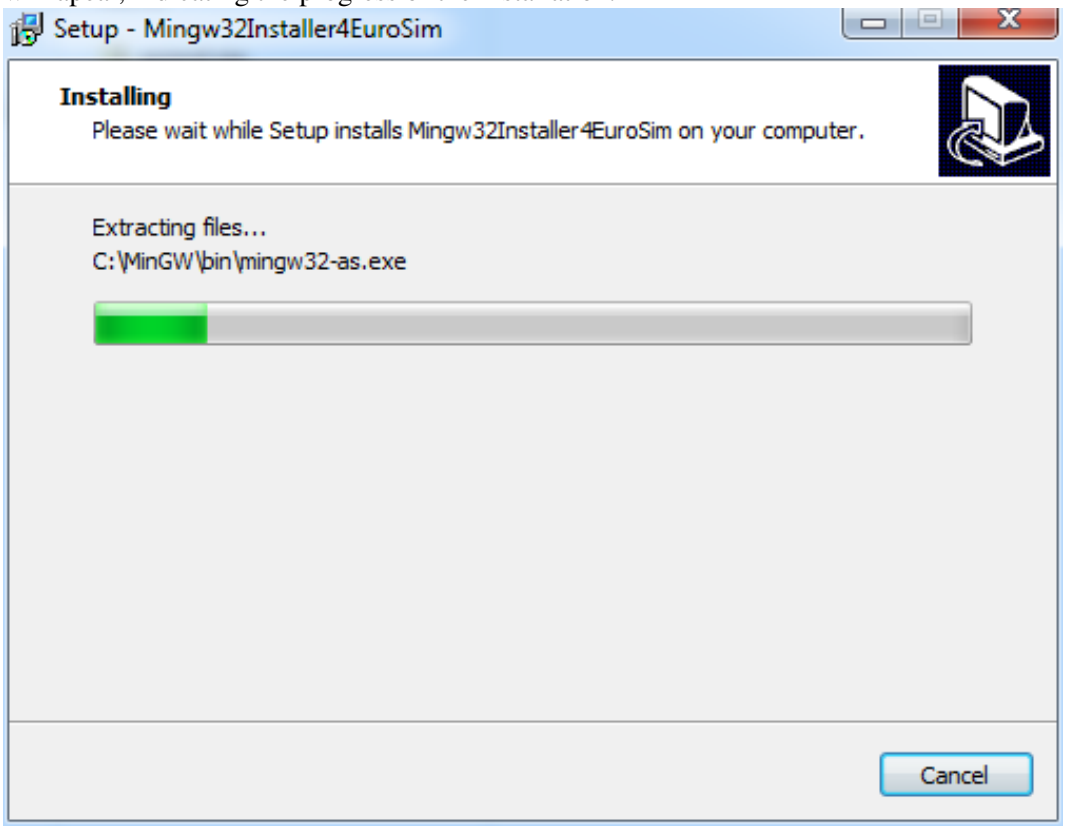
4.2 Installation procedure for Windows

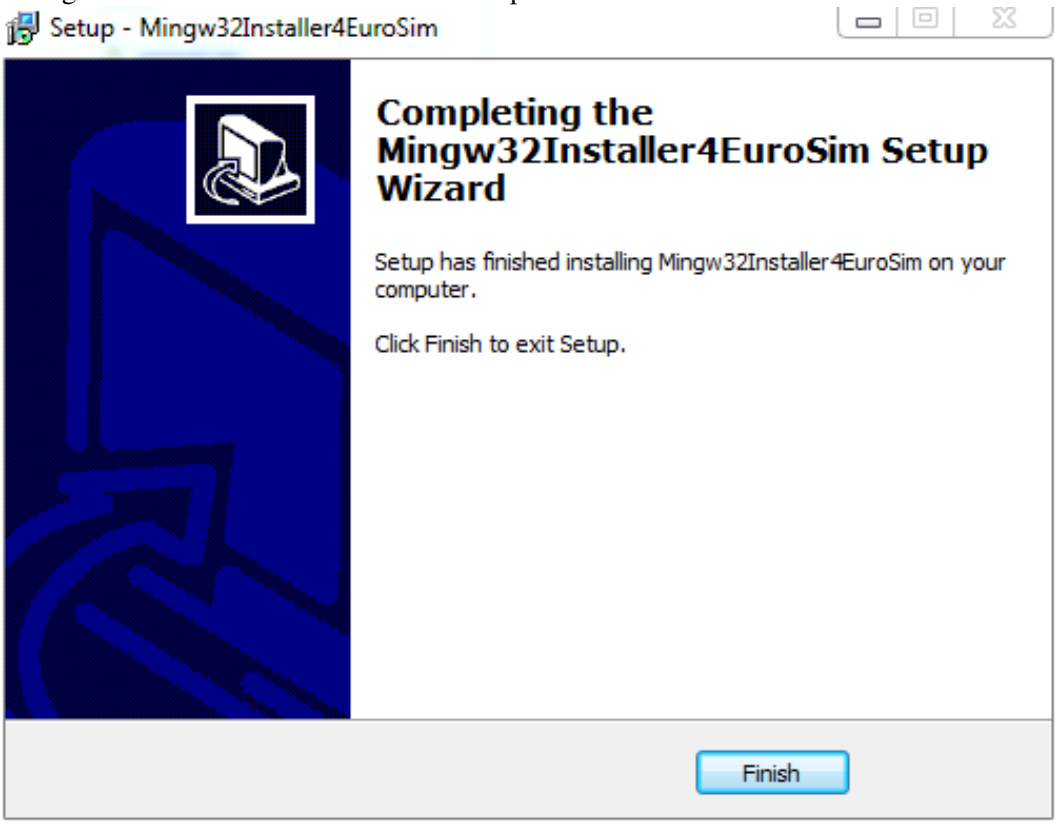
4.2.1 Windows installation steps

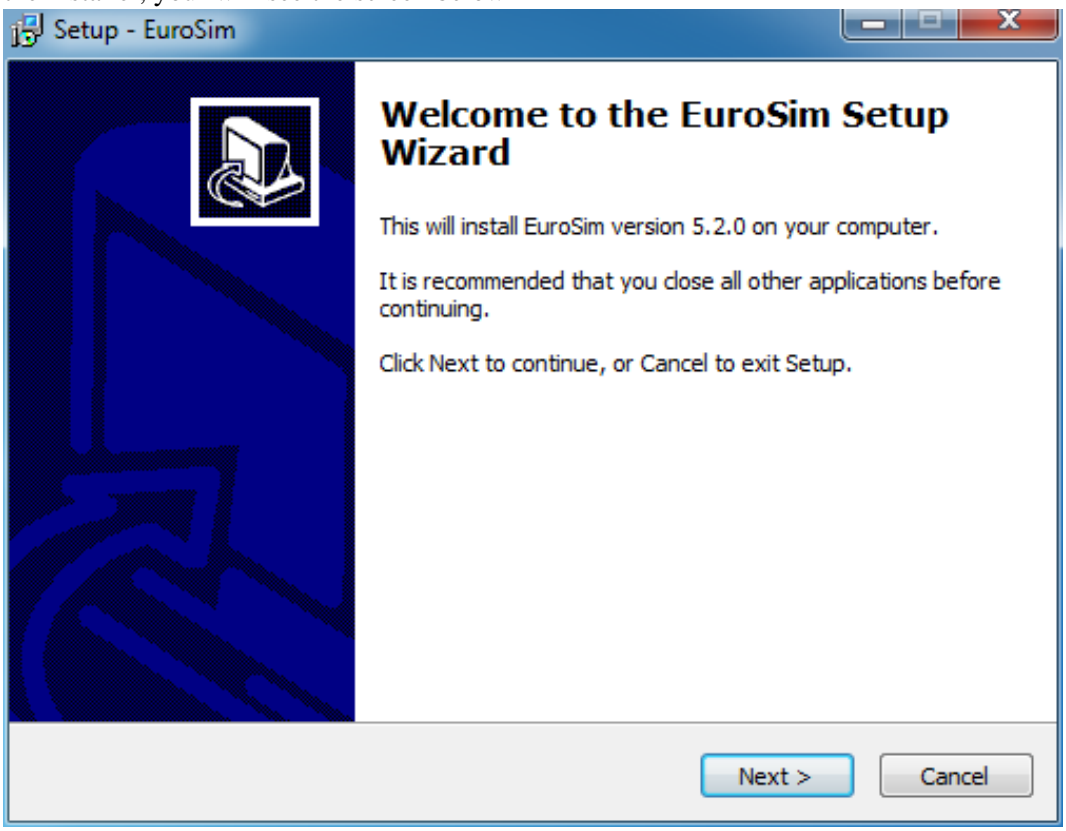
Use a user account with administrator priviliges or log in to the system as ‘Administrator’, and perform the steps in this section.

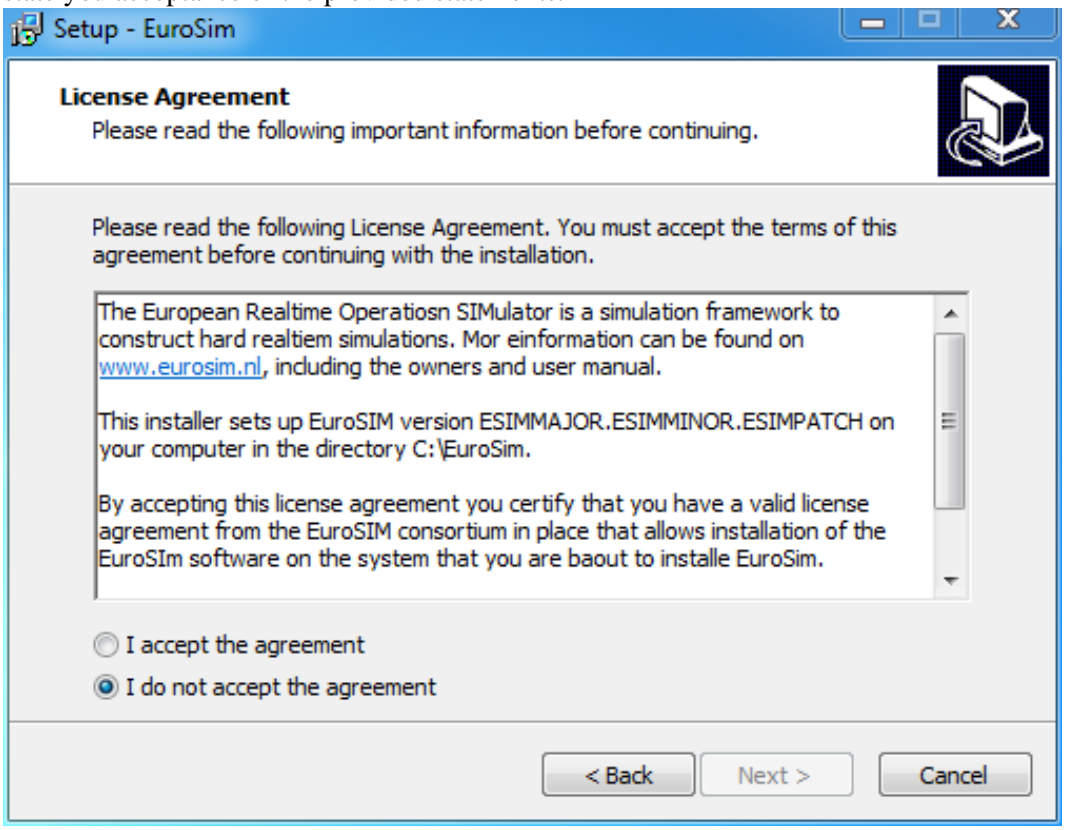
Step	Description
4.2–1	Copy the EuroSim website download URL provided to you by the EuroSim helpdesk in your favorite internet browser and press enter. The provided link will look like `http://www.eurosim.nl/software-download/<assigned name>`.
4.2–2	Enter your username and password in the authorization dialog to log in to your download section on the EuroSim website
4.2–3	You will see EuroSim directories with major and minor version numbers. Enter the directory for the version that you want to install, and subsequently for the patch release that you want to install and then go to the win32 directory.
4.2–4	If MinGW is not on your system alread and you would like to install MinGW/MSYS via the provided convenience installer then double click the file MinGW4EuroSimSetup-5.x.y to initiate the download and installation process.Following figure shows the opening screen of the MinGW installer. Note that due to the size of this file it may take some time before it is download and running. 

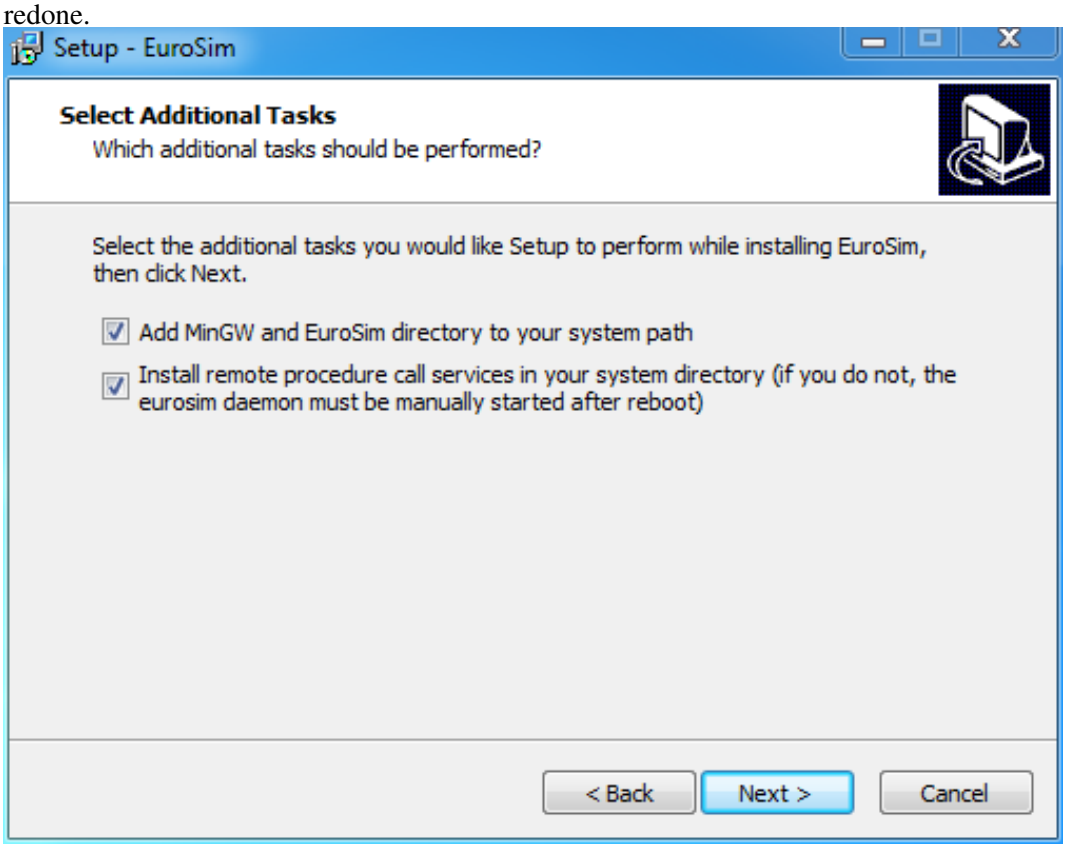
Step	Description
4.2–5	Press the Next button, which will lead to the License Agreement dialog. Please Review and state you acceptance of the provided statements.This dialog also lists all the content that is installed and which actions that is equal to when performed manually. 
4.2–6	Press the Next button to continue to the Task selections dialog as shown below. Keep both tasks accepted,in the future we may have additional options, but these are mandatory. 

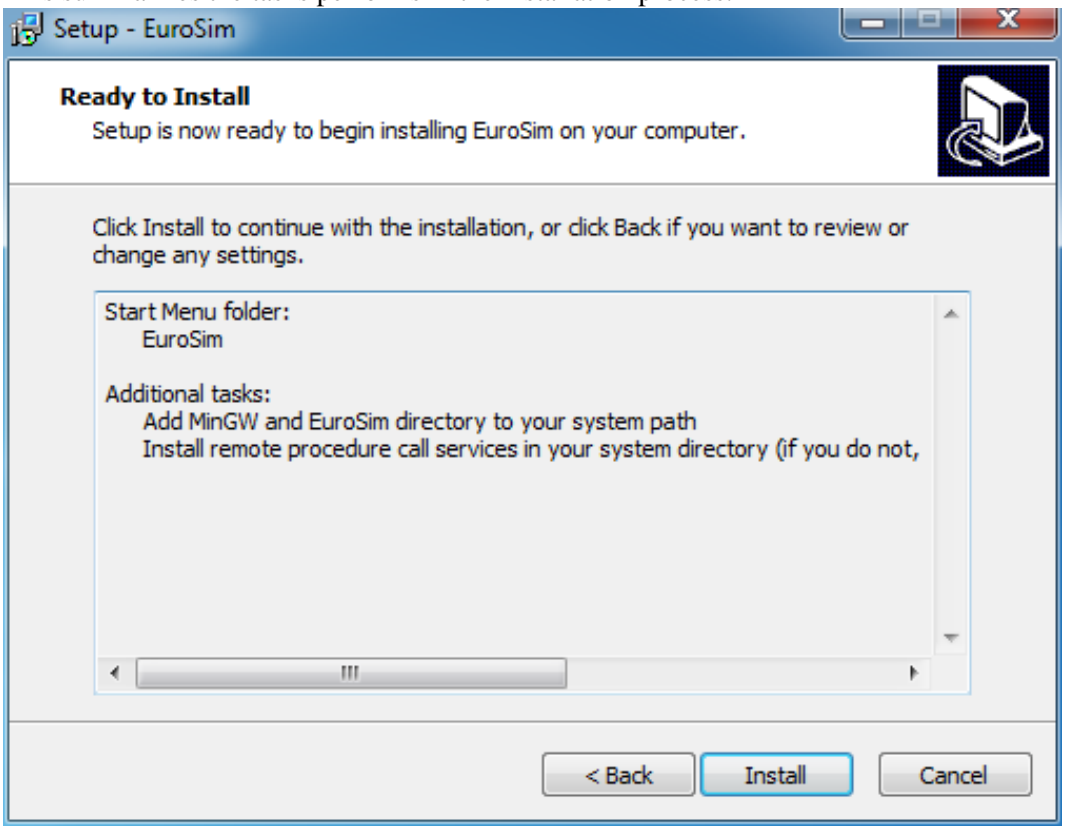
Step	Description
4.2–7	Press the Next button to continue to the Task review dialog as shown below. This shows an overview of the tasks conducted by the installer. 
4.2–8	Press the Install button to perform the actual installation. The progress dialog shown below will apear, indicating the progress of the installation. 

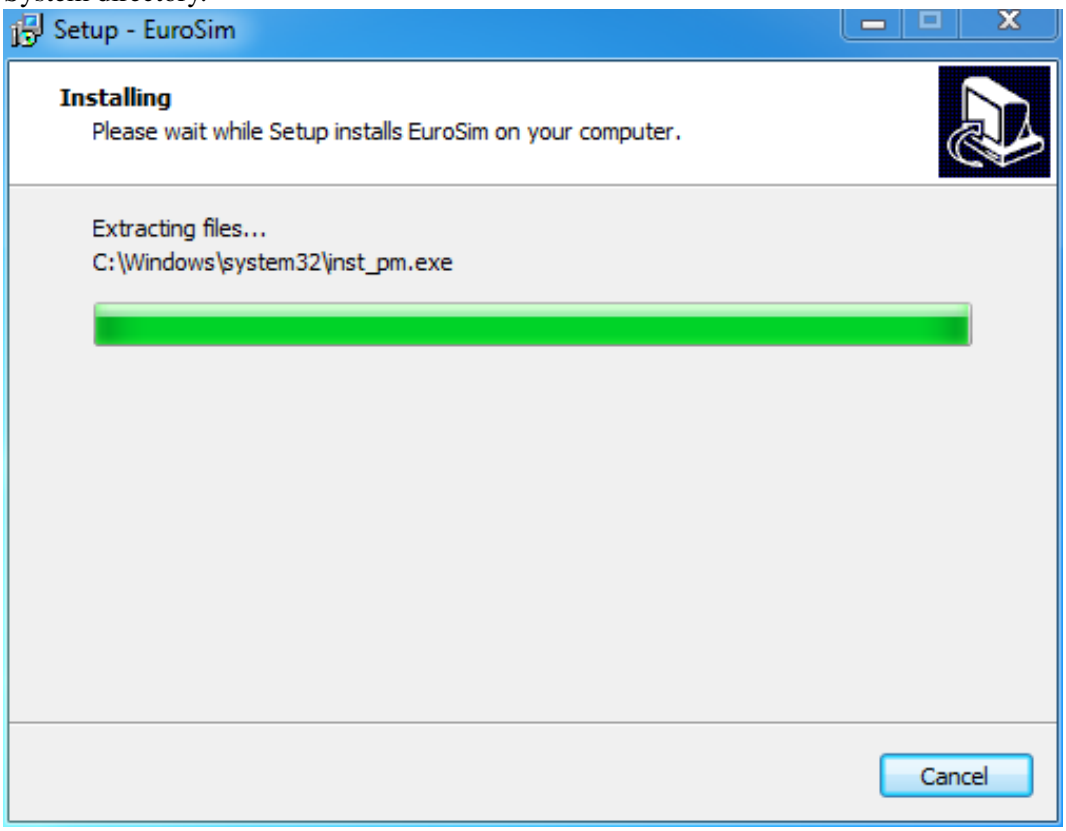
Step	Description
4.2–9	When completed the dialog below is shown. You now have all the open source tools and utilities required by EuroSim on your computer, including the proper PATH variable settings as needed to install EuroSim itself. press Finish to exit. 

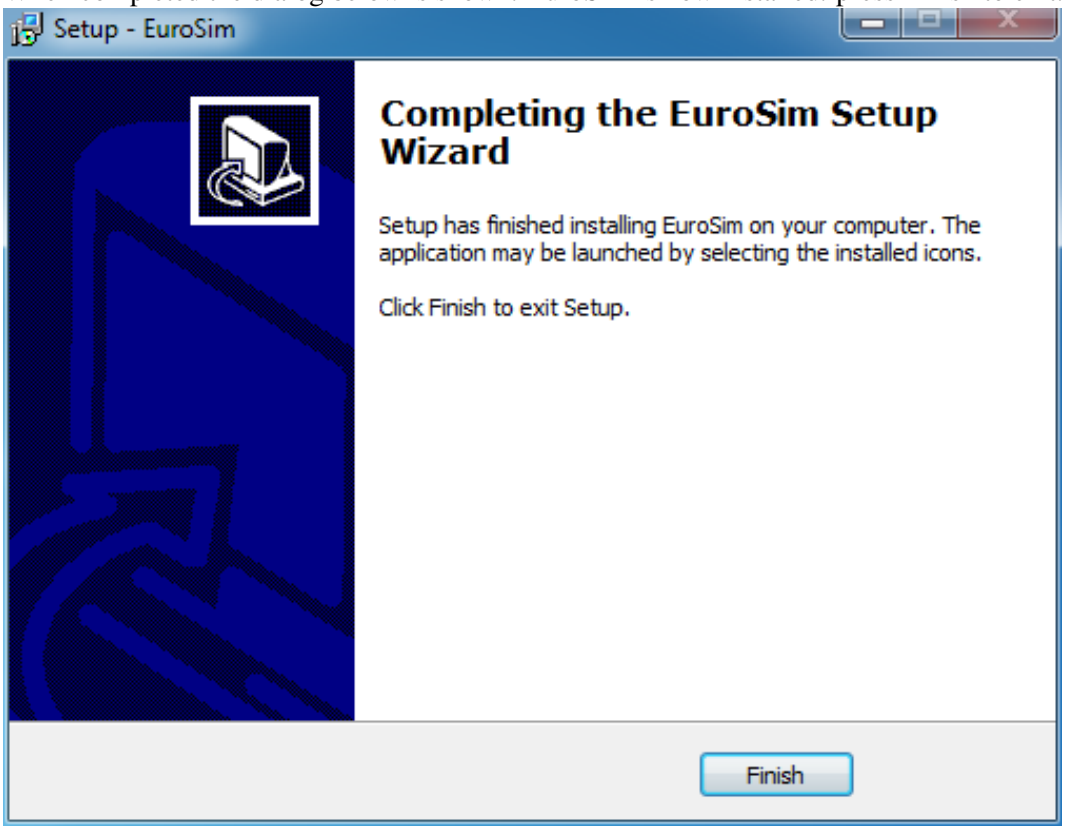
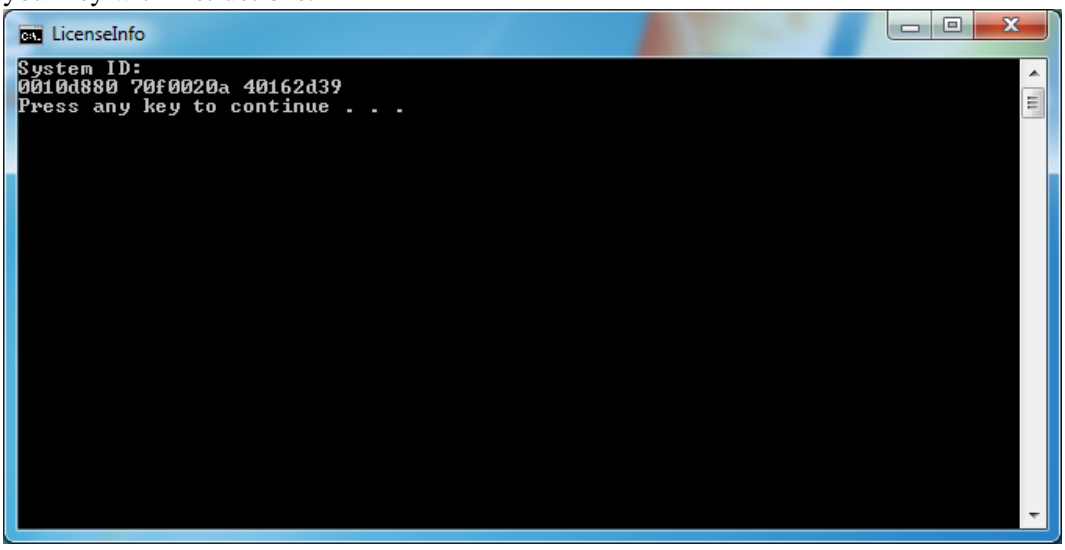
Step	Description
4.2–10	When MinGW/MSYS is available on your system, EuroSim itself can be installed. Please double click in your download section on the EuroSim website the file EuroSimSetup-x.y.z.exe where x is the Major, y the minor and z the path level of the EuroSim version you would like to install. After your system has downloaded and started the installer, your will see the screen below 

Step	Description
4.2–11	Press the Next button, which will lead to the License Agreement dialog. Please Review and state you acceptance of the provided statements. 

Step	Description
4.2–12	Press the Next button, which will present the tasks dialog shown in the figure below. The adding to the system path task is mandatory. The remote procedure call service may be skipped if already running from a previous installation, though it will not be a problem if redone. 

Step	Description
4.2–13	Press the Next button, which will present the task review dialog shown in the figure below. This summarizes the tasks performs in the installation process. 

Step	Description
4.2–14	Press the Install button. Progress of the installation is shown as in the dialog below. If the installer encounters files that already existed, a warning may pop up allowing the user to ignore or overwrite or cancel. Usually this is due to EuroSim files from a previous installation, and it is best to select overwrite to replace them with the latest version. This may also happen on the portmapper software which is installed with EuroSim in the System directory. 

Step	Description
4.2–15	When completed the dialog below is shown. EuroSim is now installed. press Finish to exit. 
4.2–16	The final step to perform is to send the EuroSim help desk the license info (also named sysinfo). Please press the windows Start butoon, open the EuroSim group that the installer created and run the LicenseInfo program. This program will present a window with a code as shown in the figure below. Please send this code to the EuroSim helpdesk to receive your key with instructions. 

After adding the license key recieved from the EuroSim helpdesk to the license file you can run EuroSim by double clicking the EuroSim icon on your desktop or launching it from the EuroSim group in the Start Menu. The EuroSim project manager will appear.

If instead of the project manager, an error dialog appears, or if you see a demo mode notification, then first please try a reboot of your system. If that does not resolve the problem, then please contact the

EuroSim helpdesk.

Chapter 5

Installation Verification

In order to write down the procedures as compactly as possible, the procedure steps are often specified at a higher level of abstraction. E.g. instead of saying 'Choose *Create:New:Project* and type project `my_project` and then press **OK**' it is simply written 'Create new project `my_project`'. Consult [SUM12] for details.
Optional steps are indicated by a procedure step number in a grey field; the condition is listed directly above that grey field.

5.1 Linux system

5.1.1 Linux host computer

Purpose: verify that host is compatible with EuroSim.

Step	Procedure	Expected Result	Checked
5.1–1	Note processor type (use `hinv`).	Should be Intel Pentium or better.	
5.1–2	Note memory size (use `hinv`).	Should be $\geq$512 MB.	
5.1–3	Note free disk space (use `df .`).	Free disk space should be $\geq$100 MB.	

5.1.2 Linux host distribution and operating system

Purpose: verify that host's distribution and operating system is compatible with EuroSim.

Step	Procedure	Expected Result	Checked
5.1–4	Inspect distribution (use `cat /etc/redhat-release`).	Should be a supported Redhat release or supported Redhat release (Concurrent computer).	

5.1.3 Third party software

Purpose: verify required third party software; note any exceptions.

Step	Procedure	Remark	Result	Checked
5.1–5	Check basic development system is available (use `rpm -qa` for RedHat Linux).	Not required for EuroSim run-time license.	gcc-4.4.5 or higher installed.	
5.1–6	Check if the F77 compiler is available.	Optional.	gcc-g77-3.4.6 or higher installed.	
5.1–7	Check if the C++ compiler is available.	Optional.	gcc-c++-4.4.5 or higher installed.	
5.1–8	Check if the RCS utilities are available.	Optional.	rcs-5.7-18 or higher installed.	
5.1–9	Check if the CVS utilities are available.	Optional.	cvs-1.11.5-23 or higher installed.	
5.1–10	Check if Firefox is available.	Optional.	Version 3.6.17 or higher installed.	

5.2 Windows system

5.2.1 Windows host computer

Purpose: verify that host is compatible with EuroSim.
Use *Control Panel:System* to get system information.

Step	Procedure	Expected Result	Checked
5.2–1	Note processor type.	Should be Intel Core 2 or better.	
5.2–2	Note memory size.	Should be $\geq$2 GB.	
5.2–3	Note free disk space.	Free disk space should be $\geq$2 GB.	

Step	Procedure	Remark	Result	Checked
5.2–4	Check if the right mingw compiler is available (use `gcc --version`)).	Not required for EuroSim run-time license.	`gcc version 4.8.1` or higher installed.	
5.2–5	Check if the F77 compiler is available.	Optional.	gfortran version 4.8.1 or higher installed.	
5.2–6	Check if the C++ compiler is available.	Optional.	g++ version 4.8.1 or higher installed.	
5.2–7	Check if Internet Explorer is available.	Optional.	Version 8 or higher installed.	

5.2.2 Windows host operating system

Purpose: verify that host's operating system is compatible with EuroSim.

Step	Procedure	Expected Result	Checked
5.2–8	Inspect operating system.	Should be WWindows 7 either 32 or 64 bit.	

5.3 Exercise EuroSim tools/functionalities

Purpose: test proper functioning of EuroSim on (new) host.

Step	Tool	Procedure	Result	Checked
5.3–1		Prepare user environment as described in section 6.1	Environment variables set.	
5.3–2	Project Manager	Linux/UNIX: type `esim`; Windows: double click desktop icon	Project Manager tool appears.	
5.3–3	Project Manager	Choose *Help:About*	About dialog appears with version number.	
5.3–4	Project Manager	Create new project `IVP`.	Project `IVP` appears in the Project Manager.	

Only perform step 5.3–5 if Firefox[1] is needed/installed:

Step	Tool	Procedure	Result	Checked
5.3–5	Firefox	Choose *Help:Contents* in the Project Manager.	Index to all on-line documentation appears on screen.	
5.3–6	Project Manager	Select project `IVP` and press the **Model Editor** button	Model Editor starts with empty canvas.	
5.3–7	Model Editor	Attach file node `ansi.c` and edit file to insert code as per Appendix A. View file (*Edit:View Source*) and save model as `IVP.model`.	Model hierarchy extended with file node; source code editor starts correctly; source code viewer starts correctly.	
5.3–8	Model Editor	Expand node `ansi.c`.	Variables and procedures that are candidates for the API are shown.	
5.3–9	Model Editor	API-fy variables `c_y`, `c_ampl`, `c_freq` and entry point `ansi_c` by checking the corresponding checkboxes in the listview.	Selected variables appear with checkmark.	

Only perform step 5.3–10 if Fortran is needed/installed:

[1] On Windows the viewer associated with .html documents will be started

Step	Tool	Procedure	Result	Checked
5.3–10	Model Editor	Make file node `fortran.f`. Insert code as per Appendix A (be sure to insert 6 spaces at the beginning of each line according to the Fortran 77 syntax rules). Select *Tools:Set Build Options* and select **Linux**[2] **Fortran runtime libraries** in the Support tab. Select **OK**. API-fy variables.	API-fiable variables and procedures are shown. Selected variables appear with checkmark.	

Step	Tool	Procedure	Result	Checked
5.3–11	Model Editor	Add file node `IVP.env`. Choose *Edit:Edit Source*. Press **Get Current Environment** and save.	Environment editor starts up and loads current environment.	
5.3–12	Model Editor	Choose *Edit:View Source*.	Environment viewer starts OK and shows no difference between stored and current environment.	

Step	Tool	Procedure	Result	Checked
5.3–13	Model Editor	Choose *Tools:Set Build Options*. Select **Gnat Ada runtime libraries**.	*Build Options* window appears.	
5.3–14	Model Editor	Choose *Tools:Build All*.	Makefile, `.exe` and `.dict` are successfully generated.	
5.3–15	Project Manager	Press **Schedule Editor**.	Schedule Editor starts with empty schedule.	
5.3–16	Schedule Editor	Select *File:Select Model* and choose the model created earlier. For the executing state (*View:Executing*), schedule the C, F77 and Ada entry points (as appropriate) periodically, e.g. at 20, 10 and 5 Hz respectively. Save and exit.	No problem.	
5.3–17	Project Manager	Press **Simulation Controller**.	Simulation Controller starts OK with empty canvas.	
5.3–18	Simulation Controller	Press *File:New* to choose model and schedule. Choose model `IVP.model` and schedule `IVP.sched`. Select API tab page.	Wizard appears. API tab shows `IVP.model` data dictionary.	

[2]Or Windows, depending on your platform

Step	Tool	Procedure	Result	Checked
5.3–19	Simulation Controller	Select *Insert:New MMI* and call the new MMI `IVP.mmi`. Select *Insert:New Monitor* and select variable `ansi:ansi_c:y`. Change to plot against simulation time and save as `ansi_c_y vs t`.	A new MMI tab appears. Monitor editor starts. Monitor appears on MMI tab.	
5.3–20	Simulation Controller	Select *Insert:New Scenario* and call the new scenario `IVP.mdl`. Choose *Insert:New Recorder*. Select var `ansi:ansi_c:y` and save as `ansi_c_y`.	Action editor starts. Recorder icon appears on canvas.	
5.3–21	Simulation Controller	Choose *Insert:New Initial Condition*. When prompted to save the file, save if as `IVP.init`.	The initial condition editor appears.	
5.3–22		Change initial condition of variable `ampl`. Press **OK**.	The *Input Files* tab shows the new initial condition file.	
5.3–23	Simulation Controller	Choose *Server:Select Server*.	A dialog with a list of available servers is shown.	
5.3–24	Simulation Controller	Press **Init**.	You are asked to save the changes: save the `.sim` file as `IVP.sim`. The simulator initializes and reaches `stand-by` state. Ignore warnings on inconsistent default values in the Fortran model; this is a known SPR.	
5.3–25	Simulation Controller	Press **Go**.	State changes to *executing*. The simulation time is continuously incremented. Recording action on scenario tab has status `EA` (executing and active). Monitor on the MMI tab is continuously updated and shows a sinus. The variables on the API tab are continuously updated.	
5.3–26	Simulation Controller	Create alpha numeric monitor and check that the initial condition (see step 5.3–22) has been set correctly. Enter another value on-line.	Variable's value is OK. Observe effect in time history plot.	
5.3–27	Simulation Controller	Press **Pause**. Press **Stop**. Choose *File:Exit*.	Simulator reaches *unconfigured* state.	

Step	Tool	Procedure	Result	Checked
5.3–28	Project Manager	Press **Test Analyzer**	Test Analyzer starts OK with empty canvas.	
5.3–29	Test Analyzer	Select *File:Select test result file*. Select: `IVP.model.tr` (in sub-directory `<date>/<time>/`).	File selector appears. Recording file `ansi_c_y.rec` appears on screen.	
5.3–30	Test Analyzer	Choose *View:Expand all nodes*. Select variable `ansi_c:y`. Drop variable on Test Analyzer's canvas. Select **Next**, **Next** and **Finish**.	Hierarchy is shown. Plot editor appears showing variable. `simulation_time` is listed as X variable. Plot icon appears on canvas and a window with the graph appears.	
5.3–31	Test Analyzer	Select plot icon. Choose *File:Print*. Note that the plot backend print dialog may be hidden behind an application window. Choose *Tools:Plot Backend Interface*.	Plot is printed. Plot backend command log appears, indicating plot is spooled to printer.	
5.3–32	Test Analyzer	Choose *File:Exit* and discard results.	Test Analyzer window disappears.	
5.3–33	Project Manager	Choose *File:Exit*.	The Project Manager disappears.	

Conclusion

Test successfully executed:

EuroSim version	
Hostname	
OS version	
Tester's name	
Date	
Remarks	

Chapter 6

EuroSim Projects & Users

6.1 Adding a EuroSim user

6.1.1 Adding a EuroSim user under Linux

When a user of your system wants to start using EuroSim, he/she should do the following:

Step	Description	Remarks
6.1–1	`export EFO_HOME=xxx`	Optional. By default `~/.eurosim` will be used.

Table 6.1: EuroSim user set-up for `/bin/sh` or `bash`

Step	Description	Remarks
6.1–2	`setenv EFO_HOME xxx`	Optional. By default `~/.eurosim` will be used.

Table 6.2: EuroSim user set-up for `/bin/csh`

6.1.2 Adding a EuroSim user under Windows

When a user of your system wants to start using EuroSim, he/she should set the `EFO_HOME` environment variable to point to the directory where the project files are created. On Windows, open the Control Panel (*Start:Settings:Control Panel*) and double click the *System* icon. Select the *Environment* tab and add the `EFO_HOME` variable. Whether you add the variable to the list of System Variables or User Variables depends on whether you want to share your project directory with other users or keep your projects in a private directory (i.e. not shared). Make sure that the directory name does not contain spaces.

6.2 Adding a EuroSim project

A EuroSim project simply consists of an entry in a EuroSim project database. By setting `$EFO_HOME` to the directory where the project database resides, the user can select a project from the list and execute one or more of the tools. How to add a EuroSim project is described in [SUM12], section 5.
If you use a shared project database, every user can access any project listed there. If you want to avoid this, you should use the Operating System's file protection mechanism. Various possibilities exist[1], and the following is given as a starting point only.

[1] Please consult one of the many books on UNIX or WINDOWS system administration.

When using the system-wide EuroSim project description file

1. The file `$EFO_HOME/project.db` is readable by everybody but writable by `user=root` or `group=sys` only. The normal user can thus not change its contents.

2. When you create a new project (by editing the afore mentioned project description file) also create a new UNIX group and add the users working on the project to this group. Make the project's home directory of this group, and set the group `s`-bit. Make the directory writable to group members only. You may either make it readable to everybody or to the group only, depending on security requirements.

When using local EuroSim project description files

By default every user creates a local project database and can create projects. He/She is only bound by the standard UNIX protection mechanisms you have set up as a system administrator.
As an intermediate between the two extremes sketched above, you might consider the use of a *project librarian*. This librarian is made responsible for the maintenance of a local project description file, of which he/she is made the owner. Make the project description file writable by the owner exclusively, and place it in a new directory. Create a new UNIX group, and make the aforementioned directory read/write by this group and set the group `s`-bit. Assign the people wishing to use this project description file to this new group. By doing so, the project librarian can create projects and assign users to them, without needing your assistance; the project home directories should—of course—be sub-directories of the directory created by you. Projects managed by one project librarian are protected from those by other project librarians by the UNIX group protection. Projects within a group are not protected against each other, except maybe by making file write-only for the owner.

6.3 EuroSim repository

When defining a project, one can set the project's repository, i.e. the directory where versioned files will be stored.
When using Cadese for version control, the default project repository is a sub-directory called RCS located in the project directory. People working on the same project share a single repository. By placing the repository in a higher level directory, multiple projects may share the same repository. This possibility comes in handy, if people want to have their own working space. Then they simply define their own projects (using a local project definition file), but set their repository variable to the same, shared repository.
When using CVS for version control, the project repository can be chosen arbitrarily.

Chapter 7

Troubleshooting

When experiencing problems with EuroSim, there are a number of places you can look for hints as to how to solve them.

1. When something goes wrong, EuroSim often provides feedback directly on the GUI or on `stdout` or `stderr`. Especially important to check are the command logs in the Model Editor (when generating a simulator) and in the Simulation Controller (when running a simulation).
2. When the EuroSim daemon[1] `esimd` has been started with the `-v` flag, a EuroSim simulator will output diagnostic information in the daemon's log file[2] (specified with the `-l` flag) or on the system's console.
3. When started with the `-v` flag, the EuroSim daemon will output information on what it has been doing to the syslog. Check the syslog and search for 'esimd' to find that information.
4. EuroSim is delivered with a so-called 'FAQ'; this is a list of frequently asked questions (and answers!). You can browse this list, which is in HTML format on the CD-ROM, with a standard browser.

When the problem persists, you can contact the EuroSim help desk:

EuroSim Help Desk
Dutch Space BV
PO Box 32070
2303 DB Leiden
The Netherlands
Tel: +31 71 5245 550
Fax: +31 71 5245 498
e-mail: esim-support@dutchspace.nl

In all communications, please provide your EuroSim version and license number.

[1] Implemented as a service on Windows platforms.
[2] Use the event viewer in the adminstrative tools group on Windows platforms.

Appendix A

Source Code Listings

C source of `ansi.c`

```
/*
 * File: ansi.c
 *
 * Contents:
 *
 * $Id: ansi.c,v 2.1 1997-08-18 14:39:55 brandt Exp $
 */
#include <esim.h>
#include <math.h>

static double c_y=0;
static double c_freq=1.5; /* rad/sec */
static double c_ampl=3.14;

void ansi_c(void) {
      double t = esimGetSimtime();
      c_y = c_ampl * sin(c_freq*t);
}
```

Fortran source of `fortran.f`

```
CC File:
C
C Contents:
C
C $Id: fortran.f,v 2.0 1997-03-03 14:33:53 alison Exp $
C

      subroutine F77
      implicit none
      include "esim.inc"
      double precision T, Y, AMPL, FREQ
      common /FORTRAN/ T, Y, AMPL, FREQ
      data Y, AMPL, FREQ /10, 3.14, 0.5/
      T = esimgetsimtime()
      Y = AMPL*sin(FREQ*T)
      return
      end

C     block data
C     double Y, AMPL, FREQ
```

```
C     common /FORTRAN/ Y, AMPL, FREQ
C     data Y, AMPL, FREQ /10, 3.14, 0.5/
C     end
```

Ada source of `mada.ads`

```
--
-- File:
--
-- Contents:
--
-- $Id: mada.ads,v 2.0 1997-03-03 14:33:55 alison Exp $
--
--

package MADA is
      procedure ADA95;
      y:    Long_float;
      ampl: Long_float;
      freq: Long_float;
end MADA;
```

Ada source of `mada.adb`

```
--    'Global_State_Variables
--          LONG_FLOAT MADA.ampl:
--                INIT="3.14",
--          LONG_FLOAT MADA.freq:
--                INIT="0.5"
--
--    'Entry_Point MADA.ADA95
--       'Global_Output_Variables
--                       LONG_FLOAT MADA.y:
--                             INIT="0.0"

--
-- File:
--
-- Contents:
--
-- $Id: mada.adb,v 2.1 1997-08-18 14:39:56 brandt Exp $
--
--
with esim; use esim;
-- with math_h; use math_h;

with Ada.Numerics.Long_Elementary_Functions;
use Ada.Numerics.Long_Elementary_Functions;



package body MADA is

      procedure ADA95 is
            t: Long_float;
            begin
                  t := esimgetsimtime;
                  y := ampl*sin(freq*t);
      --          y := ampl*t;
      end ADA95;
begin
      freq := 0.5;
      ampl := 3.14;
```

```
end MADA;
```


Revision Record

Iss	Rev	Date	Reason for change	Changes
1	0	1997-Apr-14	document creation	
1	1	1997-May-02	update after internal review	
1	2	1997-Jun-02	update for SPR-1721	
1	3	1997-Jun-25	update for SPR-1771, SPR-1783 and SPR-1803	
1	4	1999-Nov-18	update for IRIX 6.5 port	
2	0	2000-Mar-03	official EuroSim Mk2 release	
2	1	2000-May-2	Mk2rev1 release: added Appendix C: *EuroSim RTI: HLA extension*	
2	2	2000-Oct-6	Mk2rev2 release: fixed SPR-1830, add installation dependency `a2ps`, add installation for Linux version	
3	0	2002-May-14	EuroSim Mk3 release	
3	1	2003-Sep-12	EuroSim Mk3.1 release	Updated requirements.
3	2	2004-Oct-08	EuroSim Mk3.2 release	
4	0	2006-Feb-23	EuroSim Mk4.0 release	
4	1	2008-Jan-28	EuroSim Mk4.1 release	
4	2	2010-Aug-8	EuroSim Mk4.2 release added Appendix D: *Embedded EuroSim extension*	
4	3	2011-Jun-15	EuroSim Mk4.3 release	
4	4	2011-Nov-7	EuroSim Mk4.4 release	
5	0	2012-Aug-15	EuroSim Mk5.0 release	modified EuroSim licensing scheme , removed SGI/Irix references and RTI section as these are no longer supported
5	1	2013-Jun-22	EuroSim Mk5.1 release	Major overhaul of the windows section due to the first windows7 release, and removal of the embedded appendix
5	2	2014-Mar-05	EuroSim Mk5.2 release	Major overhaul of the windows section due to the new windows7 and 64 bit release

This document replaces two other documents:

- EuroSim Installation Guide (ING), FSS-EFO-TN-460
- EuroSim Installation Verification Procedure (IVP), FSS-EFO-PLN-513

These documents are consequently discontinued.

Bibliography

[SRN11] *EuroSim Mk5.2 software release notes*, 2011, FSS-EFO-SRN-388. Stored in `$EFOROOT/etc/SoftwareReleaseNote`. Final word from developers before packaging; always contains last and latest information concerning delivered EuroSim release.

[SUM12] Dutch Space BV, *EuroSim Mk5.2 software user's manual*, 2012, NLR-EFO-SUM-002, issue 6 revision 2. Stored in `$EFOROOT/doc/pdf/SUM.pdf`. This file contains the EuroSim Software User Manual in Adobe Acrobat format. Also stored in directory `$EFOROOT/doc/html/SUM`. This directory contains the EuroSim Software User Manual in HTML format.