

E4Coder Install instructions for Scilab

Installation and requirements

Version: 1.1

June 8, 2015

Scilab version



About Evidence S.r.l.

Evidence is a company operating in the field of software for embedded real-time systems. It started in 2002 as a spin-off company of the Real-Time Systems (ReTiS) Lab of the Scuola Superiore Sant'Anna (Pisa, Italy). Today, Evidence is a dynamic company having collaborations in the field of electronics, telecommunications, automotives, and industrial automation.

People at Evidence are experts in the domain of embedded and real-time systems, with a deep knowledge on the design and specification flow of embedded software, especially for the embedded market.

Besides providing consultancy services, Evidence also provides: BSPs based on Linux for embedded devices, evaluation boards featuring most innovative 8, 16 and 32-bit microcontrollers for the embedded market, development tools for making embedded software development easier, and tools for the schedulability analysis of real-time tasks running on your final product.

For more information see: <http://www.evidence.eu.com>

Contact Info

Evidence Srl,
Via Carducci 56
Località Ghezzano
56010 S.Giuliano Terme
PISA Italy

Tel: +39 050 99 11 224

Fax: +39 050 99 10 812

For more information about Evidence products, please send an e-mail to the following address: info@evidence.eu.com. Other information about the Evidence product line can be found at the Evidence web site at: <http://www.evidence.eu.com>.

This document is Copyright 2011-2015 Evidence S.r.l.

Information and images contained within this document are copyright and the property of Evidence S.r.l. All trademarks are hereby acknowledged to be the properties of their respective owners. The information, text and graphics contained in this document are provided for information purposes only by Evidence S.r.l. Evidence S.r.l. does not warrant the accuracy, or completeness of the information, text, and other items contained in this document. Matlab, Simulink, Mathworks are registered trademarks of Matworks Inc. Microsoft, Windows are registered trademarks of Microsoft Inc. Java is a registered trademark of Sun Microsystems. in the USA. and other countries, and are used under license. All other trademarks used are properties of their respective owners. This document has been written using L^AT_EX and L^AX.

Contents

| | |
|--|-----------|
| 1. Introduction | 6 |
| 1.1. Quick feature list | 6 |
| 1.2. Scilab support | 7 |
| 1.3. Requirements | 7 |
| 1.4. Licensing | 8 |
| 1.5. Feedback, bugs, and additional examples | 8 |
| 1.6. Technical support | 8 |
| 2. Installing E4Coder | 9 |
| 2.1. E4Coder Prerequisites | 9 |
| 2.2. E4Coder installation | 9 |
| A. License Manager | 18 |
| A.1. E4Coder licensing types | 18 |
| A.2. The License File | 19 |
| A.3. E4Coder License Manager | 22 |
| A.4. License errors | 24 |
| B. Qt Libraries Licensing | 27 |

List of Figures

| | |
|---|----|
| 2.1. Starting Scilab. | 10 |
| 2.2. Open the ATOMS Module Manager. | 10 |
| 2.3. Update the List of Packages. | 11 |
| 2.4. The ATOMS list of packages. Note the category “Finite State Machine (FSM) and GUI Prototyping. | 11 |
| 2.5. The E4Coder Plugin in the ATOMS Repository. | 12 |
| 2.6. The E4Coder Plugin has been installed. You will need to restart Scilab. | 13 |
| 2.8. Example of valid license. | 13 |
| 2.7. The E4Coder License Manager. | 14 |
| 2.9. Select one of the available compilers. | 15 |
| 2.10. The E4Coder Banner, which indicates that E4Coder has been installed correctly. | 15 |
| 2.11. This message will appear if you started Scilab without the USB Dongle connected | 16 |
| | |
| A.1. Dialog box showing instructions on how to setup a node-locked license. | 18 |
| A.2. Dialog box showing instructions on how to setup a Dongle license. | 19 |
| A.3. The About menu in SMCube. A similar one is available in E4CoderGUI. | 20 |
| A.4. The About panel in SMCube. | 20 |
| A.5. The About panel in E4CoderGUI. | 21 |
| A.6. The License panel in E4CoderGUI. | 22 |
| A.7. The E4Coder License Manager. | 23 |
| A.8. License Path Management. | 24 |
| A.9. E4Coder did not find the dongle matching the current license. | 25 |
| A.10. E4Coder found that the installed license has expired. | 26 |
| A.11. E4Coder could not find the license file. | 26 |

About this document

This document describes the installation procedure of E4Coder on Scilab. E4Coder is a set of tools enabling simulation and automatic code generation of complex designs for small microcontrollers.

Function of the document

The function of this document is to provide information about the installation procedure of E4Coder on top of Scilab.

Document history

| Version | Date | Author | Company | Change Description |
|---------|----------|-----------|--------------|---|
| 1.0 | Mar 2015 | Paolo Gai | Evidence Srl | Initial version. |
| 1.1 | Jun 2015 | Paolo Gai | Evidence Srl | E4Coder now on the standard ATOMS repository. |

Acronyms

| Acronym | Meaning |
|-------------|--|
| E4Coder | E4Coder is a factorized acronym meaning “Evidence Erika Enterprise Embedded Coder”. Initially meant as a code generation tool supporting the ERIKA Enterprise RTOS, it is now a general purpose code generation toolset. In that sense, it can be now also interpreted as “Evidence For Coder”, as a set of tools for simulation and code generation of complex designs. |
| E4Coder GUI | E4Coder GUI is the GUI editor for fast prototyping of graphical user interfaces. |
| FSM | Finite State Machine |
| HIL | Hardware In the Loop |
| SMCube | SMCube, or SM^3 , is a recursive factorized acronym which means “SMCube is a State Machine System Modeler”. |

1. Introduction

E4Coder is a set of tools to simulate and generate code for embedded controllers in a standalone configuration or running a real-time operating system.

E4Coder is composed by the following tools:

- *Scilab*, free software for simulation of complex designs [1];
- *E4Coder Code Generator*, an efficient embedded code generator;
- *SMCube*, a finite state machine modeler;
- *E4Coder GUI*, a prototyping tool for Graphical User Interfaces.

The main usages of E4Coder are:

- E4Coder is meant to be a complete tool suite allowing companies to start the development of a *model-based development flow* that starts from the modeling of a complex system down to its implementation on one or more microcontroller systems.
- E4Coder can be used as a *tool for prototyping embedded graphical user interfaces*, thanks to the possibility of simulating a complex model including SMCube state machines that can be shown using E4Coder GUI panels.
- E4Coder can be used as a tool for *Processor In the Loop (PIL)*, *Hardware In the Loop (HIL)* and calibration. Thanks to its communication blocks, E4Coder can be used to log data from the target, post-process it, and for sending control information to the target in real-time.

The idea driving the development of E4Coder was the need for a *lightweight* set of tools, based on open-source libraries and using open-source software on the target microcontroller, which could help designing complex systems at reasonable license costs.

1.1. Quick feature list

Using the E4Coder toolset, you can easily:

- Simulate continuous time and discrete time designs.
- Edit, simulate and generate code for Finite State Machines (FSM) using SMCube. Hierarchical state machines with parallel states are supported, with additional support for junction points.

- Simulate and generate code from the same XCos diagram without changes. In other words, the diagram contains both the superblocks to be generated part as well as a simulation of the environment where they will run.
- Generate compact and readable code for your embedded single and multirate algorithm, that can be easily integrated in your mass-production project.
- Generate pure library code for small microcontrollers without an RTOS.
- Generate a complete Real-Time / Soft Real-Time project for a number of operating systems, including OSEK/VDX based kernels (with the open-source OSEK/VDX RTOS Erika Enterprise [2]), Linux pthreads, Windows
- Customize the variable names used in the generated code using link names.
- Create custom blocks to integrate existing legacy applications, functions, and data in your XCos project.
- Edit and simulate embedded panels using E4CoderGUI.

1.2. Scilab support

E4Coder is currently available for Scilab 5.5.2 for Windows, 32 and 64 bits.

The Scilab porting currently includes:

- SMCube (editor and simulation);
- E4CoderGUI (editor and simulation).

The porting currently does not include the code generator and the integration with target boards. The porting of the code generator is planned for the next release.

1.3. Requirements

E4Coder is currently distributed as a customized ATOMS repository, and has the following requirements:

- Windows hosts;
- Scilab 5.5.2 32 or 64 bits;
- A supported compiler¹ among the following ones:
 - Microsoft Visual C++ (supported versions: Express edition 2008, 2010);
 - MinGW on Windows, installed in C:\MinGW, from the following package:

<http://www.e4coder.com/download/mingw-get-inst-20120426.exe>

¹A working compiler is needed in order to be able to simulate an SMCube diagram.

1.4. Licensing

Scilab is distributed under the CeCILL License².

E4Coder, E4Coder GUI, and SMCube are packaged as independent toolboxes, and are distributed under a commercial license.

SMCube and E4Coder GUI uses the Qt Libraries, which are dynamically linked to allow you to change the library with new versions at your choice. Licensing information about the Qt Libraries is available in Appendix B.

1.5. Feedback, bugs, and additional examples

We hope that you will enjoy E4Coder and its features for your embedded projects. If you have any suggestions on how to improve E4Coder and the surrounding tools, please do not hesitate contact us by writing to info@evidence.eu.com.

Please also periodically check the E4Coder website (see below) for additional information, demos, and examples on the E4Coder website:

<http://www.e4coder.com/>

in particular on the E4Coder Documentations and Videos page.

1.6. Technical support

Technical support for E4Coder is available directly from Evidence. Many information can be found directly on the E4Coder website, at the following address:

<http://www.e4coder.com>

For sales, pricing, order status, license renewals and general information, please contact Evidence Srl directly at the address and phone numbers available at the following web page:

<http://www.evidence.eu.com/en/contact-us.html>

²See <https://www.scilab.org/scilab/license> for more information.

2. Installing E4Coder

E4Coder is packaged as a customized ATOMS repository. To install E4Coder, you need to perform the following steps:

2.1. E4Coder Prerequisites

In the following section are listed required programs/libraries needed by E4Coder.

1. One of the following compilers supported by E4Coder:

- a) Microsoft Visual Studio Express edition version 2008 or 2010.

Visual Studio is required by XCos and by SMCube. Other versions of Visual Studio Express edition *will not work*.

The location to start the download is:

<http://www.microsoft.com/visualstudio/eng/downloads>.

- b) MinGW, installed from the following URL:

<http://www.e4coder.com/download/mingw-get-inst-20120426.exe>

2. Scilab 5.5.2.

You need to install the Windows version (32 or 64 bit) as downloaded from:

<http://www.scilab.org>.

2.2. E4Coder installation

1. Before starting the installation, be sure to have received a License file provided by Evidence. The license file is typically called `license.xml`. It will be needed later at the first load of the E4Coder toolbox¹
2. If you have a USB Dongle provided by Evidence, connect it in an empty USB port of your PC.
3. Start Scilab *with Administrator privileges*, as shown in Figure 2.1.

¹It will not be required during the installation procedure itself, because the installation is done with administrator privileges, whereas the License file is copied in the user home.

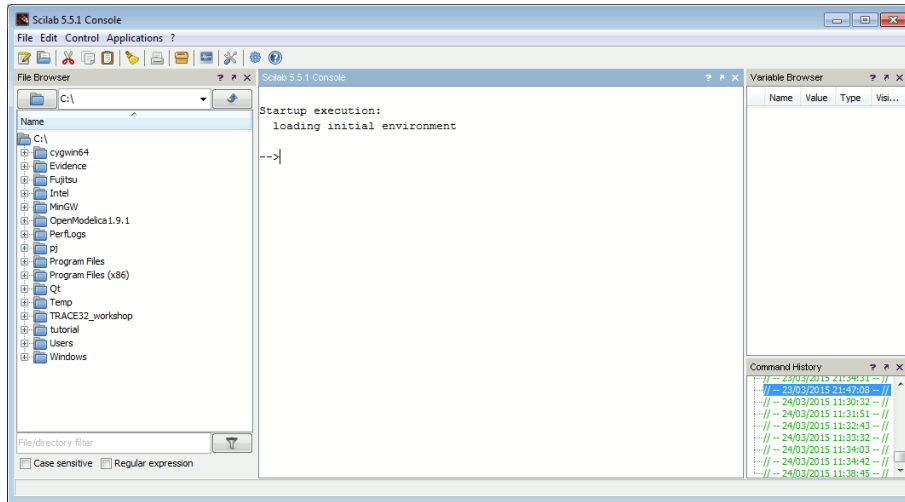


Figure 2.1.: Starting Scilab.

4. Please note again: *you need Administrator Privileges to install E4Coder*. To gain Administrator Privileges, you need to right-click on the Scilab program icon, and select “Run as Administrator”.
5. E4Coder is distributed as an online ATOMS repository. In order to be able to install it, you need to open the ATOMS Module Manager, as shown in Figure 2.2.

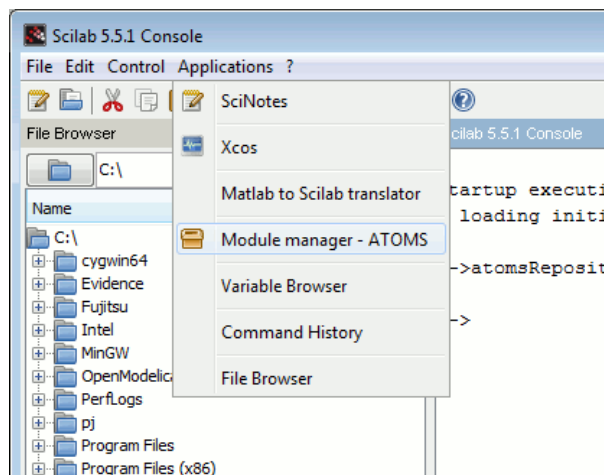


Figure 2.2.: Open the ATOMS Module Manager.

6. Update the list of available packages (see Figure 2.3) to be sure to have all the latest repositories up to date (it may take a few minutes).
7. Select the “Finite State Machines (FSM)” Section under the list of categories (see Figure 2.4). Then select the “E4Coder Plugin” as shown in Figure 2.5.

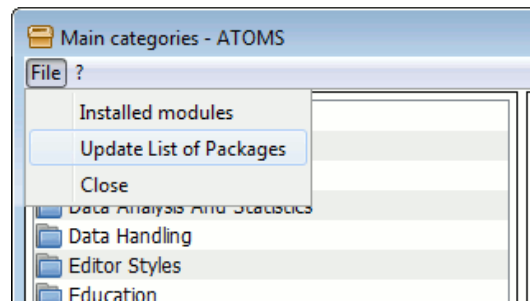


Figure 2.3.: Update the List of Packages.

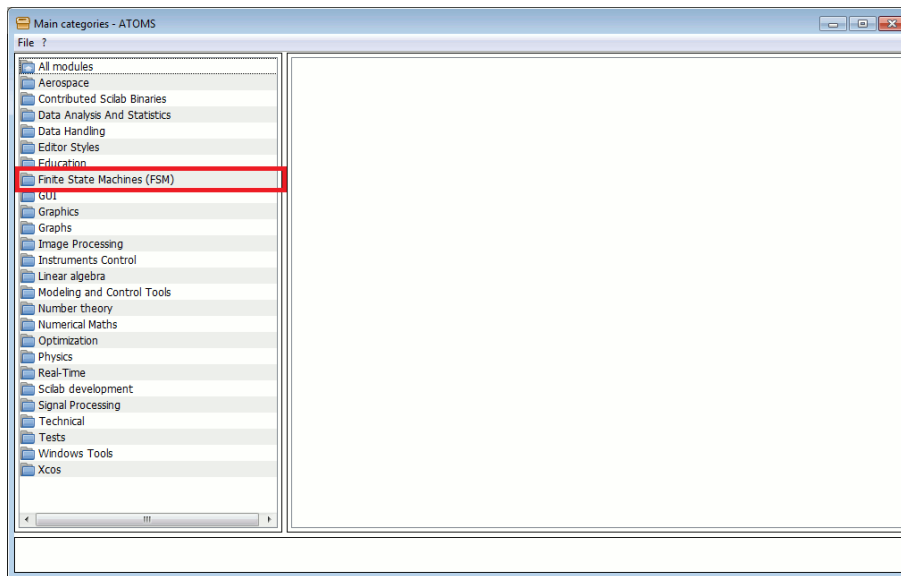


Figure 2.4.: The ATOMS list of packages. Note the category “Finite State Machine (FSM) and GUI Prototyping.

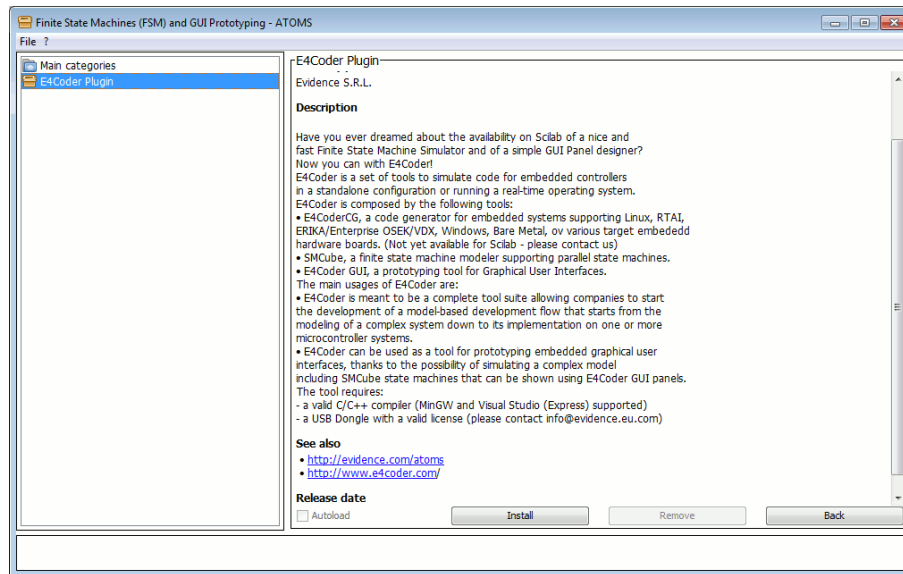


Figure 2.5.: The E4Coder Plugin in the ATOMS Repository.

8. Click on the Install Button. As a result, you will be required to restart Scilab *with Administrator Privileges* (see Figure 2.6).
9. Please restart Scilab *with Administrator Privileges*.
10. At the E4Coder load, the loader will check for a valid license file. In case of license check fail, the E4Coder License Manager window will appear (see Figure 2.7). Please refer to Section A for a detailed description of License Manager.

At the first launch, the License Manager will show the license file has not been found in the default directory. In order to load a new license file, you have to press the “Choose” button. After that, the License Manager will show a dialog window that enables to traverse the file system in order to select license file. Then you have to press the “Load” button so that the license file will be **copied** into the in the default directory. Finally, the License Manager will check the license validity and will show the corresponding result:

- in case of a valid license (see Figure 2.8), the License Manager will end and E4Coder will continue the loading operation.
- in case of no valid license, the License Manager will request a new valid license. Please refer to Section A.4 for more information on license errors. Notice that by pressing the “Exit” button with no valid licenses, the E4Coder load will fail.

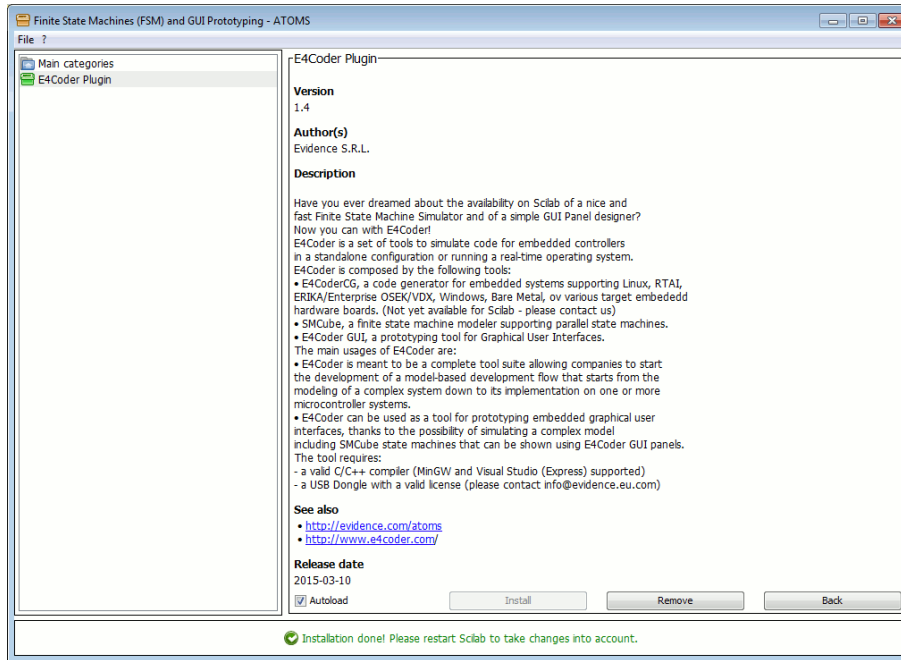


Figure 2.6.: The E4Coder Plugin has been installed. You will need to restart Scilab.

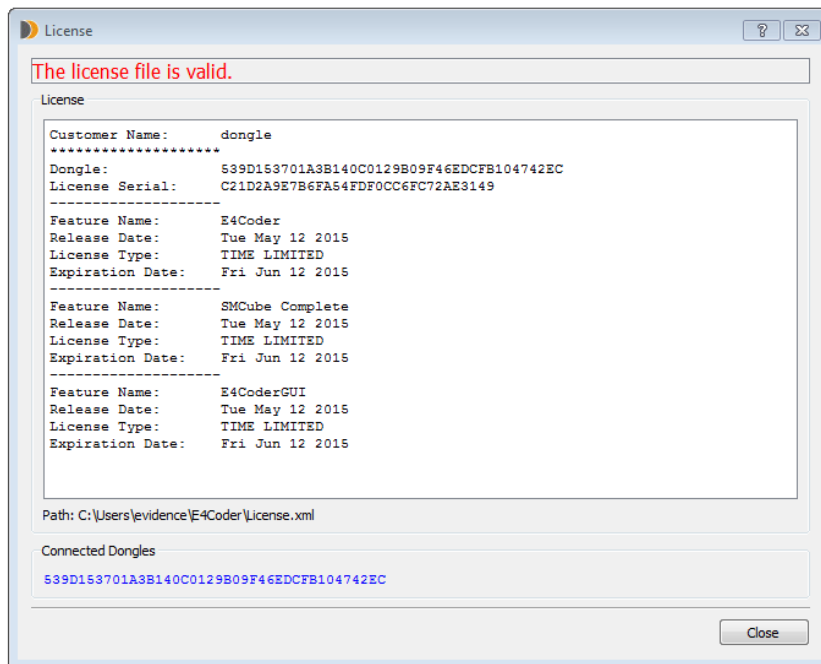


Figure 2.8.: Example of valid license.

WARNING: Please do not put the license file `License.xml` file inside the E4coder installation directory because it could be deleted the next time you'll update E4Coder. We suggest to maintain the default path.

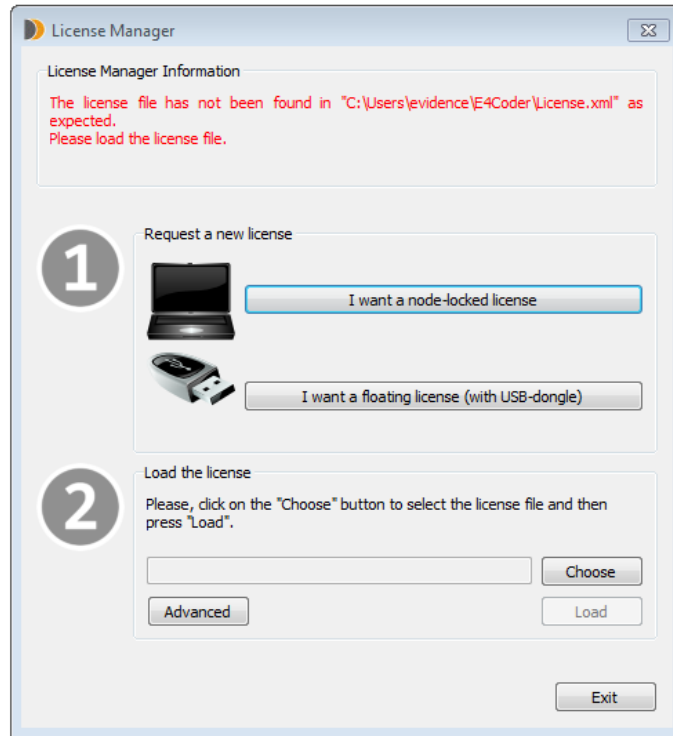


Figure 2.7.: The E4Coder License Manager.

11. In case of successful license check, we need to inform SMCube about the right compiler toolchain to use (see Figure 2.9). The tool lists the compiler toolchains automatically found. You can select one of them, or you can specify the location of the compiler if you did not install it in the default directory.
12. Finally, E4Coder prints a nice banner, with the indication of the current build (see Figure 2.10).
13. Unless you specified differently when installing the repository (see the Autoload checkbox in Figure 2.6), E4Coder will start automatically at each Scilab startup. Please note that if the USB Dongle is not connected at startup, you will get a warning message such as the one in Figure 2.11.

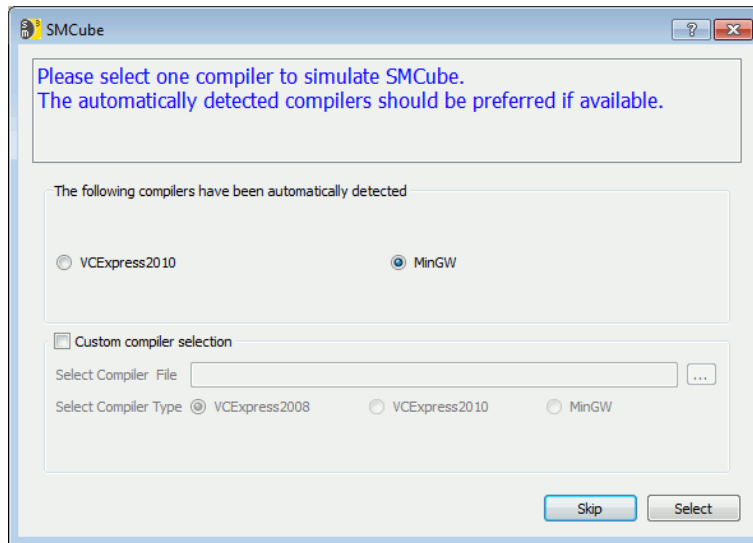


Figure 2.9.: Select one of the available compilers.

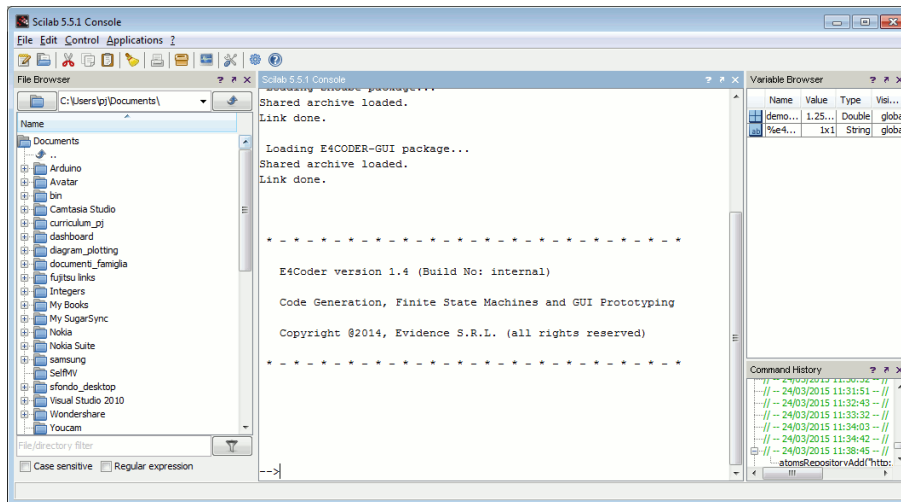


Figure 2.10.: The E4Coder Banner, which indicates that E4Coder has been installed correctly.

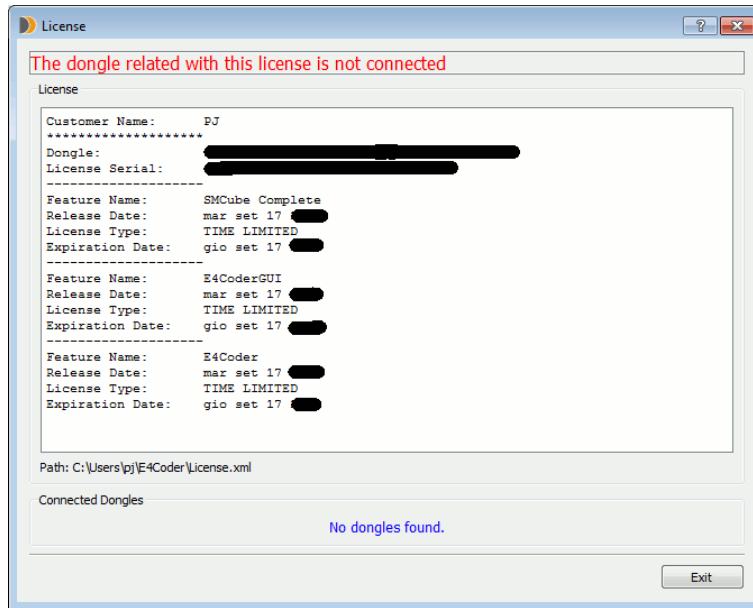


Figure 2.11.: This message will appear if you started Scilab without the USB Dongle connected

Bibliography

- [1] Scilab Enterprises. Scilab. open source software for numerical computation. <http://www.scilab.org/>.
- [2] Evidence Srl and many other contributors. Erika enterprise and rt-druid. <http://erika.tuxfamily.org>.

A. License Manager

E4Coder is available with two kind of licensing schemes described in Section A.1. In order to run the E4Coder toolset needs a License file `License.xml` provided by Evidence (see details in Section A.2).

Section A.3 gives an overview of the License Manager operations, whereas Section A.4 details the possible license errors shown by the License Manager.

A.1. E4Coder licensing types

E4Coder is available with two kind of licensing schemes:

Node-locked license A node-locked license is an encrypted “key” that is locked to a specific machine. The node-locked license authorizes E4Coder running only on the related PC. In order to obtain the node-locked license the user has to contact Evidence s.r.l. and send the serial code that will appear in the License Manager when loading E4Coder for the first time (see Figure A.1, which appears after pressing the button “*I want a node locked license*” described in Section A.3).



Figure A.1.: Dialog box showing instructions on how to setup a node-locked license.

Dongle license A dongle license is an encrypted “key” that is locked to a specific USB Dongle that will be provided by Evidence s.r.l.. This floating license authorizes running E4Coder only with the related USB-dongle connected. In order to obtain the floating license the user has to contact Evidence s.r.l. to obtain the license file and the related USB-dongle. As the first thing before loading E4Coder, please

remember to connect the USB Dongle to your PC (see Figure A.2, which appears after pressing the button “*I want a floating license*” described in Section A.3).

In case of USB Dongle license Evidence will provide a matching USB dongle. E4Coder can be installed on multiple systems with the same valid License file, but at any one time it can be run only on the system where the USB dongle is connected.

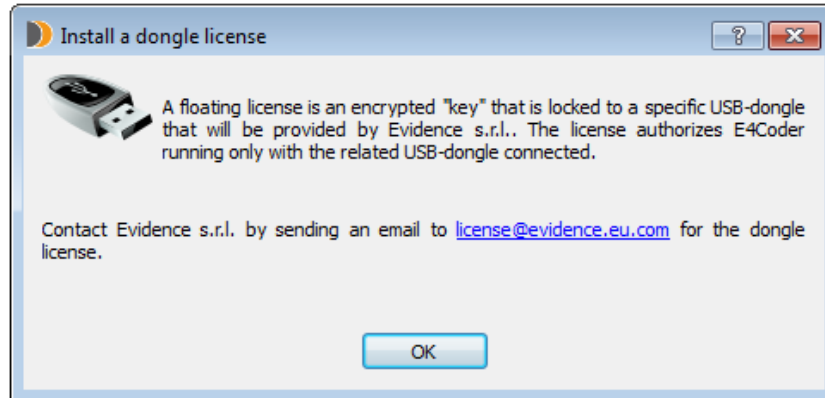


Figure A.2.: Dialog box showing instructions on how to setup a Dongle license.

A.2. The License File

The License file provided by Evidence enable a set of *features* of the toolset, and can set different *licensing* for each of them.

The features specified in a License are:

E4Coder The main component of the toolset, needed to install and load the ScicosLab extension pack.

SMCube Complete The SMCube feature; if not specified in the License, simulation and code generation for state machines is disabled.

E4CoderGUI The user interface simulation feature; if not specified in the License, GUI blocks are disabled.

For each feature, the following types of licensing are available:

TIME LIMITED Enables the specific feature up to a given expiration date.

TIMED Enables the feature indefinitely, allowing to use any version released before the expiration date.

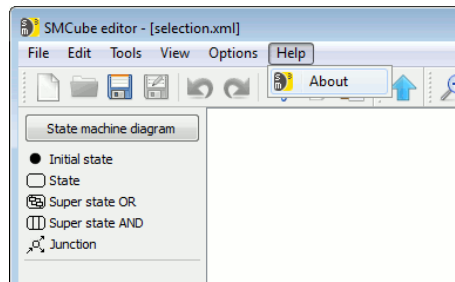


Figure A.3.: The About menu in SMCube. A similar one is available in E4CoderGUI.

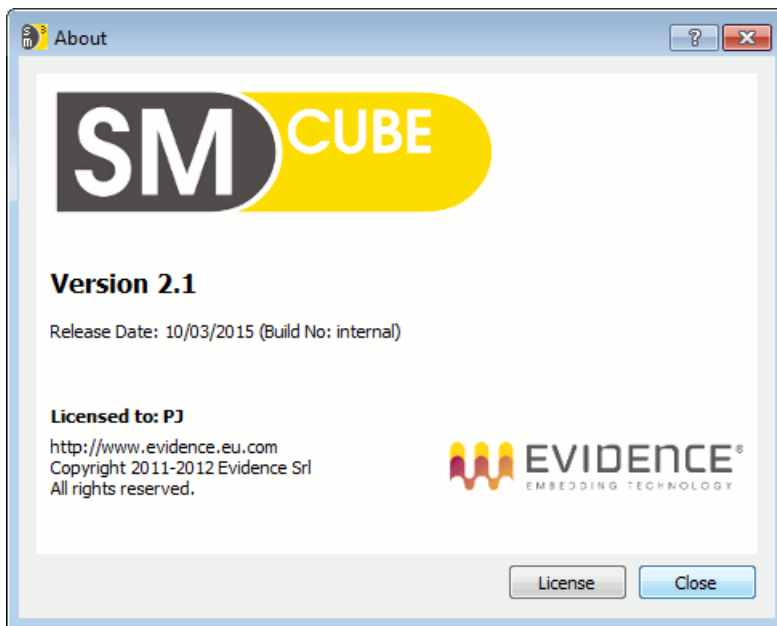


Figure A.4.: The About panel in SMCube.

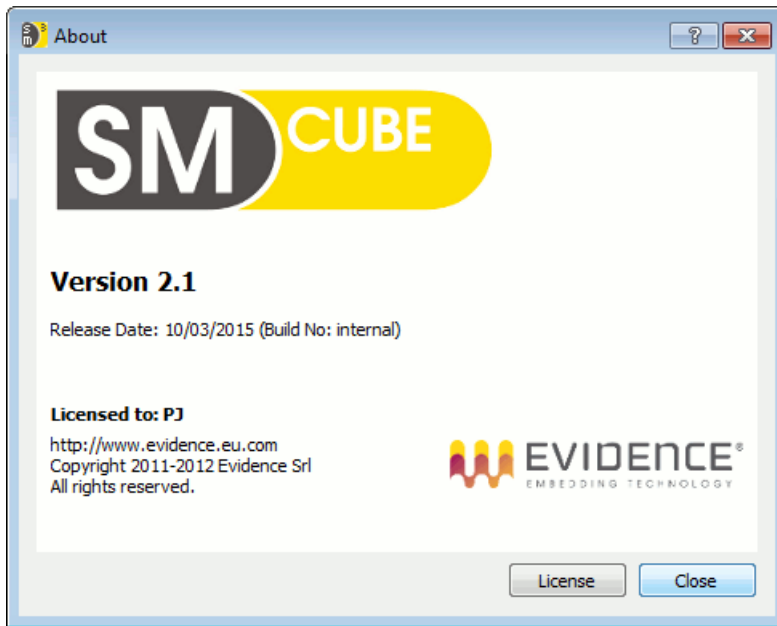


Figure A.5.: The About panel in E4CoderGUI.

The `License.xml` file is installed by default in the `E4Coder` directory under the current user home directory. After the installation, the currently installed license can be examined from the About menu in `SMCube` or `E4CoderGUI` (see Figure A.3 and A.4).

The licensing information is accessed by clicking on the `License...` button (see Figure A.5).

The License panel, shown in Figure A.6, displays all the information about the current license:

- the customer name;
- the serial number linked to the license
 - in case of USB Dongle license, the serial number of the USB dongle linked to the license;
 - in case of node-locked license, the serial number of the machine linked to the license;
- the license serial number;
- for each feature, the release date, the license type, and the expiration date.
- the path of the current license directory.

Moreover, in case of USB Dongle license, the License panel will show the Dongles section (see Figure A.6). Such section will allow to check the serial number of the dongle that is currently connected to the system. If the user connects another dongle, he/she can

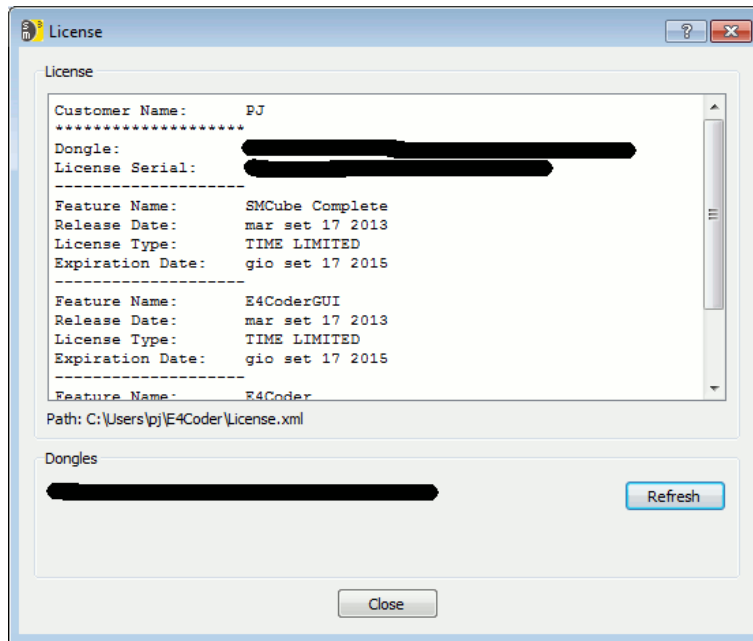


Figure A.6.: The License panel in E4CoderGUI.

force the system to read its serial number by hitting the Refresh button. Obtaining the serial number of the dongles can be useful when contacting Evidence (e.g. for license renewals).

A.3. E4Coder License Manager

The E4Coder License Manager will appear when loading E4Coder in case there is no license installed in the system, or in case of a license error. Furthermore, the License manager will be explicitly called by invoking the function `E4Coder_license_manager()` in ScicosLab in order to:

- examine the currently installed license,
- select a new license file
- define a new license path location.

As shown in Figure A.7, the License Manager will give an overview of license file. Furthermore, in case of license error, the License Manager will give details on the occurred error with the corresponding solution. Please refer to Section A.4 for the description of license errors.

The License Manager window is then organized in two steps:

1. How to require a new license.

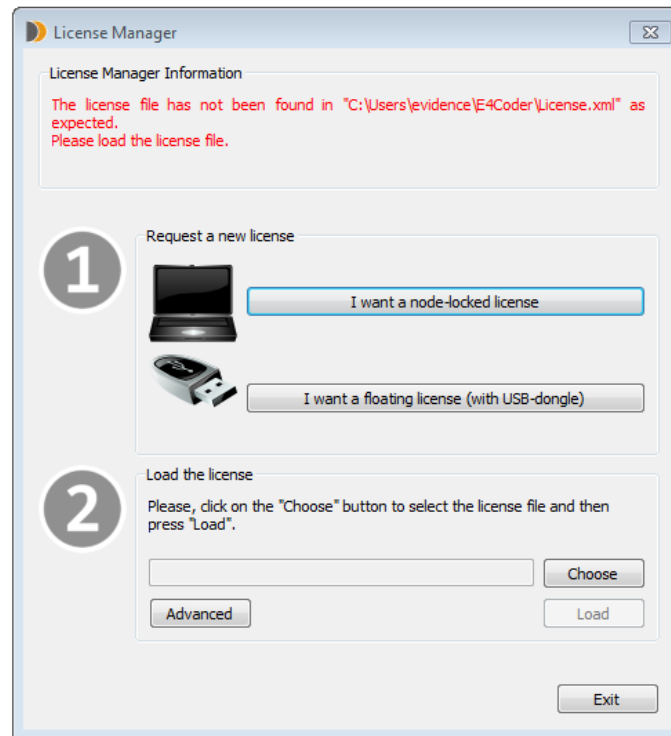


Figure A.7.: The E4Coder License Manager.

The License Manager will give information on how to request a new license (see Figure A.7). By pressing on the corresponding button, the License Manager will show the instructions on how to obtain a node-locked license or a USB Dongle license.

2. How to install the new license.

The License Manager will allow to load a new license file in the default directory and, as advanced setting, to change such default directory (see point 2 in Figure A.7).

In order to load a new license file, you have to press the “Choose” button. After that, the License Manager will show a dialog window that enables to traverse the file system in order to select license file. Then you have to press the “Load” button so that the license file will be copied into the in the default directory. Finally, the License Manager will check the license validity and will show the corresponding result:

The License Manager also allows to change the default directory where the license files are loaded for the check. With reference to Figure 2.7, you can press the button “Advanced” to open the “Advanced settings” dialog (see Figure A.8). By pressing on the “Choose” button, the License Manager will show a dialog window that enables to traverse the file system in order to select new license path. Then you have to press the “OK” button so

that the License Manager will change the default directory that will contain the copy of the license files.

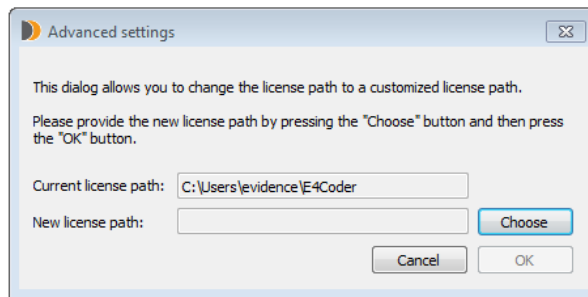


Figure A.8.: License Path Management.

A.4. License errors

When working with E4Coder, a number of errors can occur which are related to licensing issues. Table A.1 gives an overview of the most common license errors that can be shown by the License Manager with the corresponding solution.

In case of USB Dongle license the most common type of errors involves the use of the USB dongle. If E4Coder cannot find the correct dongle, matching the installed license, an error will occur, as shown in Figure A.9. Note that the error dialog shown in the figure reports a case in which E4Coder could not find the valid dongle, but instead found another one with a different serial number, shown in blue in the Connected Dongles section at the bottom.

Another common error results from a license expiration: if the installed license for the feature in use is of type TIME LIMITED and its expiration date has passed, an error like that in Figure A.10 will be shown. Please contact Evidence to renew your license.

Finally, if E4Coder could not find the License file in the chosen location, it will show a message like that in Figure A.11. If this happens, please check the path you chose for the license during installation (the default is ‘‘<user home directory>\E4Coder’’): it should contain the file `License.xml`. Otherwise, and if you have a copy of the license file, copy it to that location and restart E4Coder. If the problem persists, try reinstalling E4Coder. If you need to obtain a new copy of your license file, contact Evidence and please be sure to provide the serial number related to the license.

| License Manager errors | How to solve |
|---|---|
| The license file has not been found in the default E4Coder directory as expected. | Load a license file or contact Evidence s.r.l. for the new license file as explained in Section A.1. |
| The node-locked license file does not refer to the current node. | The node-locked license authorizes E4Coder running only on a specific machine. Load the license file related to the machine or contact Evidence s.r.l. for the new license by sending the serial code as explained in Section A.1. |
| The license check failed due to license file corruption. | Try to reload the license file. In case of continuous error due to file corruption, please contact Evidence s.r.l.. |
| The license does not list the feature "E4Coder". | The current license does not authorize the execution of E4Coder. Please load a license file that includes the required feature or contact Evidence s.r.l. for a new license. |
| The dongle related to the license file is not connected. | Please connect the dongle related to the floating license. |

Table A.1.: License Manager errors.

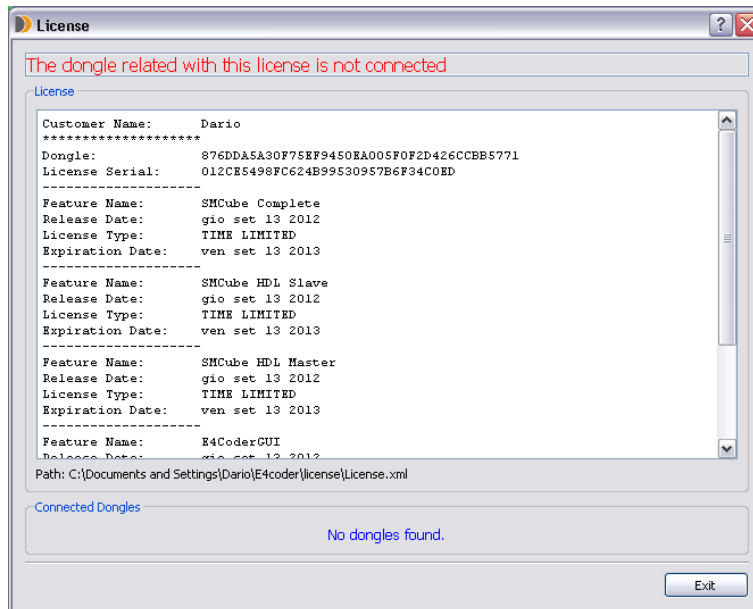


Figure A.9.: E4Coder did not find the dongle matching the current license.

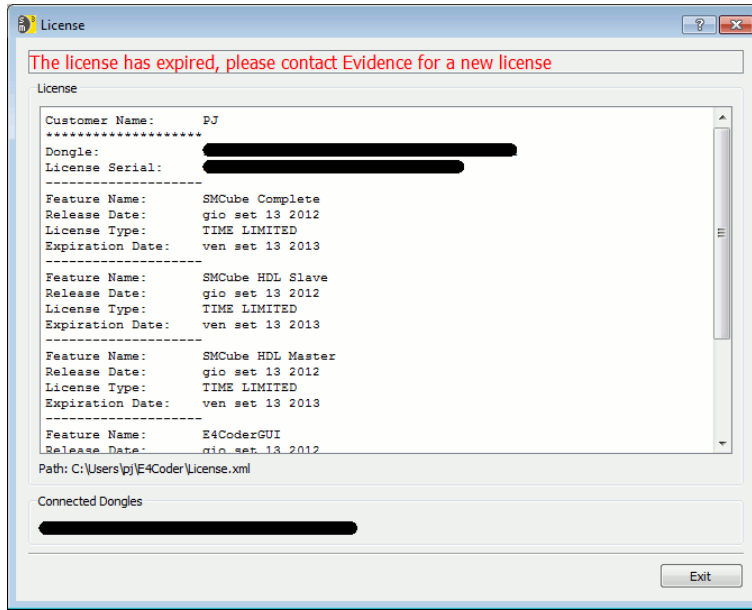


Figure A.10.: E4Coder found that the installed license has expired.

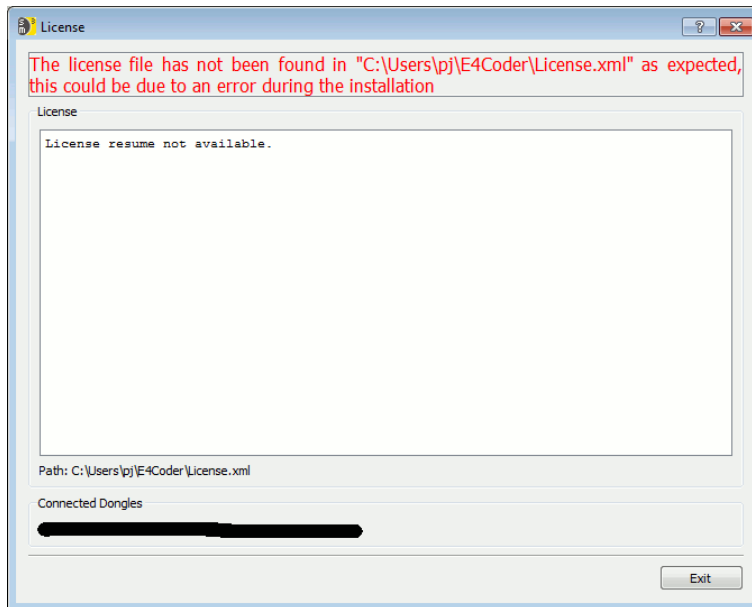


Figure A.11.: E4Coder could not find the license file.

B. Qt Libraries Licensing

Parts of E4Coder are written using the Qt Libraries, which are dynamically linked to the executable. The Qt GUI Toolkit is Copyright (C) 2011 Nokia Corporation and/or its subsidiary(-ies), and the following license applies.

The GNU Lesser General Public License (Version 2.1)

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive

or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the “Lesser” General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under

Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user’s computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this

License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can

do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
};one line to give the library's name and a brief idea of what it does.};  
Copyright (C) };year}; };name of author};
```

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library ‘Frob’ (a library for tweaking knobs) written by James Random Hacker.

```
};signature of Ty Coon};, 1 April 1990  
Ty Coon, President of Vice
```

That’s all there is to it!