Elite EL
.NET Enveloper

Develop guide v1.0.7.0

SENSESHIELD

Copyrights and Trademarks

The Elite EL with its technical documentation is copyrighted (C) 2017 to present by Senseshield Technology Co., Ltd (Senseshield). All rights reserved.

All products referenced throughout this document are trademarks of their respective owners.

All attempts have been made to make the information in this document complete and accurate. Senseshield is not responsible for any direct or indirect damages or loss of business resulting from inaccuracies or omissions. The specifications contained in this document are subject to change without notice.

Contact

Company Name: Beijing Senseshield Technology Co., Ltd

Address: Suite 510,Block C,Internet Innovation Center,Building 5,
No.10,Xibeiwang East Road,Haidian District,
Beijing China

Tel.: +86-10-56730936
Fax: +86-10-56730936-8007

Sales: sense@sense.com.cn
Website: www.sense.com.cn

License Agreement

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE CONTENTS THEREOF AND/OR BEFORE DOWNLOADING OR INSTALLING THE SOFTWARE PROGRAM. ALL ORDERS FOR AND USE OF THE Elite AND/OR EL FAMILY PRODUCTS (including but not limited to the Kit, libraries, utilities, diskettes, disc, Senseshield® and/or Senseshield® keys, the software component of Senseshield and/or EL and the EL License Guide) (hereinafter “Product”) SUPPLIED BY Senseshield Technology Co., Ltd (hereinafter "Senseshield") ARE AND SHALL BE, SUBJECT TO THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT.

This document is a legally binding agreement between you (either an individual or an entity) and Senseshield®. If you are not willing to be bound by the terms of this agreement, you should promptly (and at least within 3 days from the date you received this package) return the unused developer's kit and the programmer's guide to Senseshield. Use of the software indicates your acceptance of these terms.

- **GRANT OF LICENSE**

  The software of the Product is being licensed to you, which means you have the right to use the software only in accordance with this License Agreement. You may (a) copy the software for internal use, (b) modify the software for the purpose of integrating with your application and (c) merge the software with other programs.

- **NON-PERMITTED USES**

  Except explicitly permitted in this License Agreement, you may not (a) copy, modify, reverse engineering, decompose, assemble the Product in whole or in part, or (b) sell, lease, license, transfer, distribute all or part of the Product or rights granted in this License Agreement.

- **LIMITED WARRANTY**

  After the date of purchase, Senseshield provides 24-month warranty that the Senseshield EL key has no material and manufacturing defects substantially. All the responsibilities of Senseshield Software Technology Co., Ltd and all the compensation you can get under warranty are: you can require replace/repair the Product or accept other remedial measures.

- **LIMITATION OF LIABILITY**

  Under any circumstances, Senseshield will NOT be liable for any damages arising out of usage or inability of the Product, including but not limited to: loss of data, loss of profits, and other special, incidental, joint, secondary or indirect loss.

  Except for the limited warranty offered to the original buyer, Senseshield is not responsible for providing any insurance to anyone on the product, performance and service including merchantability and fitness for a particular purpose.

  The entire product, including Senseshield EL, the software, the document, other material shipped as accessories, and backups made by you are copyrighted by Senseshield Security GmbH.

- **TERMINATION**

  Your failure to comply with the terms of this License Agreement shall terminate your license and this License Agreement.

Contents

Copyrights and Trademarks ..... **I**
Contact ..... **II**
License Agreement ..... **III**
Contents ..... **IV**
Overview **1**
About the Guide 1
What is EL .NET Enveloper? 1
Mechanism **2**
Instruction **3**
Enveloper Instruction 3
Add Files into File List 3
Parameters Settings 5
Input Passwords 9
Envelope Software 10
Save Project 11
Remote Update 12
Supplement **17**
Device Not Found Error 17
Invalid Device Error 17
Background Check Errors 17
Device Locked 18
Module Expiration Notice 18
Module Expired Error 19
Executable File Error 19
Appendix **20**
Supported Versions for Developer Tools 20
Supported Operation Systems 20

Overview 1

About the Guide

Mode	Model	Version	Releasing Date
Elite EL .NET Enveloper	STD, Genii, RTC, RTCC	v1.0.7.0	2017.02.22

- **Conventions Used**

The following conventions are used throughout this document:

Italic	Words in italic represent file names and directory names.
Bold	Words in boldface represent keystrokes, menu items, and window names and fields.
	The caution icon flags some content you should be careful of.

- **Document Improvement**

Document Writing Team dedicates to insure the accuracy and completeness of context. Your feedback will assist them to make continuous improvement on EL document. Please do not hesitate to email us, sense@sense.com.cn

What is EL .NET Enveloper?

EL .NET Enveloper helps you reduce time and effort that you spend on software protection. Using .NET Enveloper you can simply add the encryption code to compiled binaries code to prevent software license abuse, without modify the source code of software. .NET Enveloper supports encryption for each method and binding authorization function at the same time, thereby limiting the using frequency and using times for methods.

However, **EL .NET Enveloper** cannot provide protection as strong as encrypt with API calls, which is comprehensive and more secure protection method. Insight of that, it is recommended to use **EL .NET Enveloper** as a temporary solution.

Mechanism 2

How is the **EL .NET Enveloper** different from any other ordinary enveloper tools? **EL .NET Enveloper** binds the envelope with EL device, so that software runs properly only if the corresponding EL device is plugged in. To increase the difficulty of de-shelling and protection strength, **EL .NET Enveloper** also offers anti-debugging, device detection and symbol obfuscation while software running.



EL .NET Enveloper supports EXE and DLL assemblies, which are complied as X86.

Instruction 3

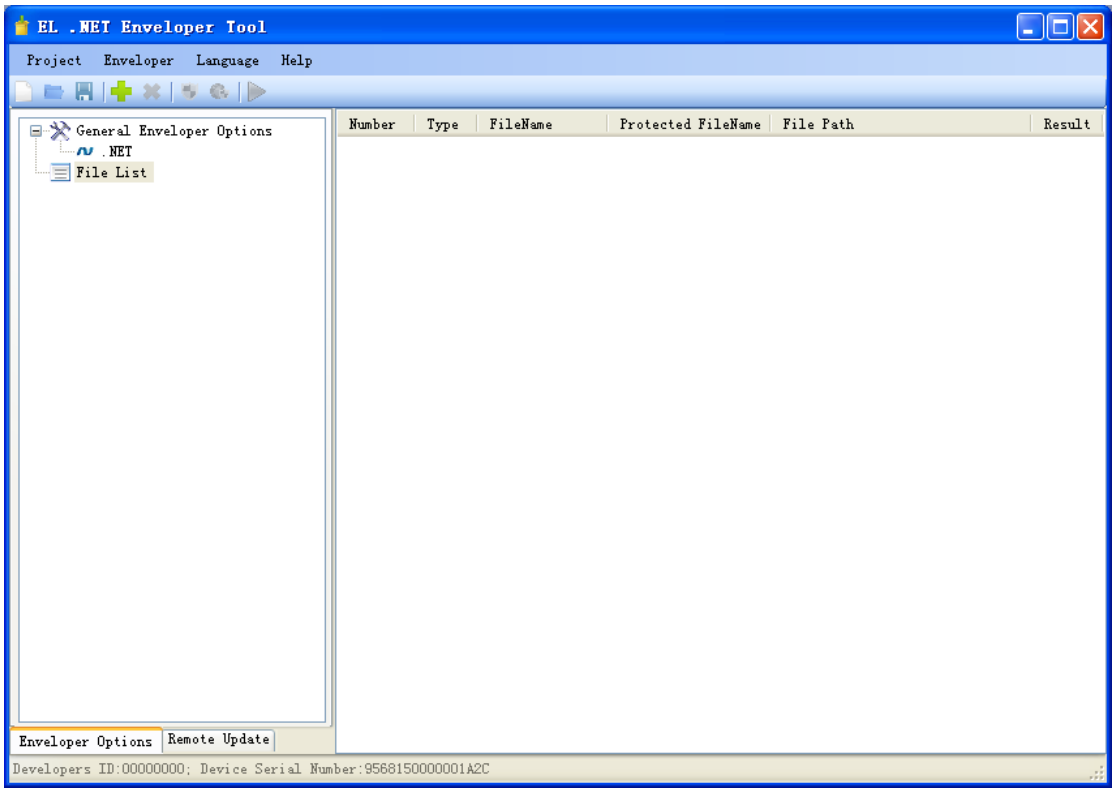
This Chapter gives a brief guide for how to use **EL .NET Enveloper**:

Enveloper Instruction

With **EL .NET Enveloper**, there are three simple steps to complete files encryption and protection:
Step 1. Add files into *File List*.
Step 2. Set envelope parameters (envelop options, symbol obfuscation, strong-name signing, anti-debugging, bind serial number, background check and custom error messages).
Step 3. Input PIN codes and passwords: Developer PIN, User PIN and EL internal programs encryption password, which will be verified before envelope complete.
Following are detail instructions for how to use **EL .NET Enveloper** to envelope files.

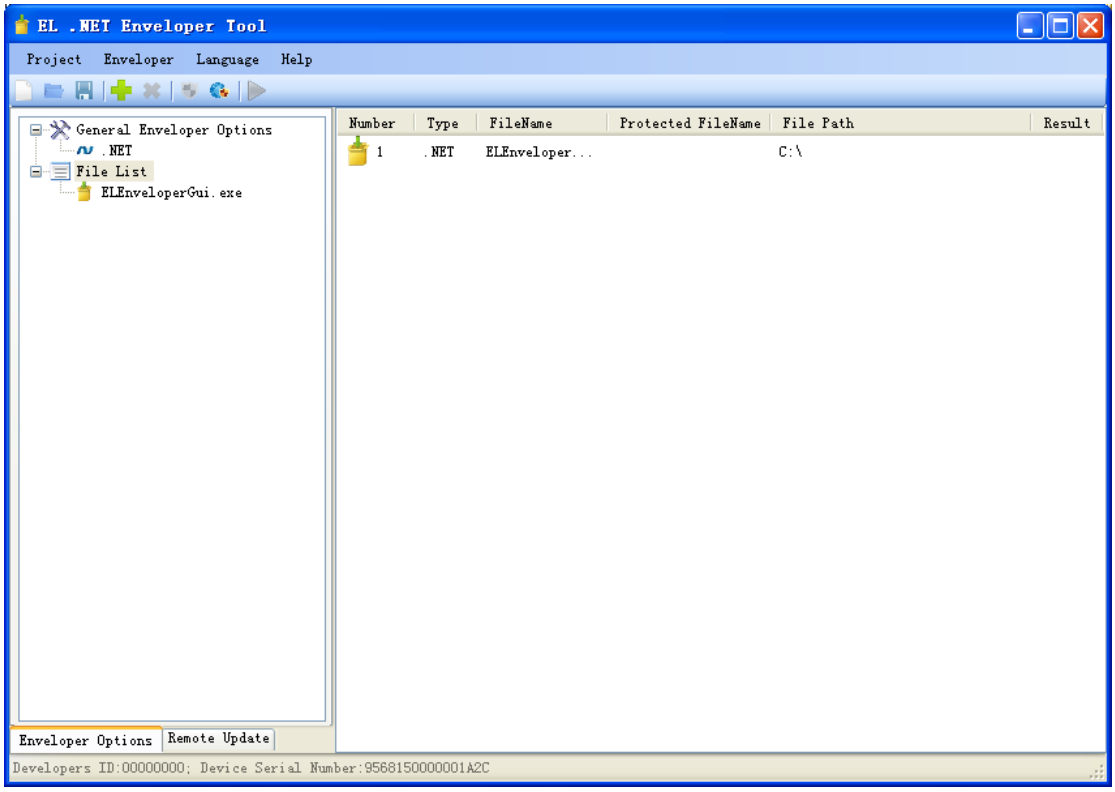
Add Files into File List

1、Plug in corresponding EL device.
2、Launch EL .NET Enveloper by double click ELEnveloperGui.exe. Select operation language under 'Language' in menu bar. EL .NET Enveloper v1.0.7.0 supports Chinese and English operation languages. Developer ID and Device Serial Number will show at the bottom of the control panel (Figure 3-1).



(Figure 3-1)

1、Click  in tool bar to add files into File List or you can drag files into the File List directly. Added files will show in navigation pane, under the File List icon (Figure 3-2).



(Figure 3-2)

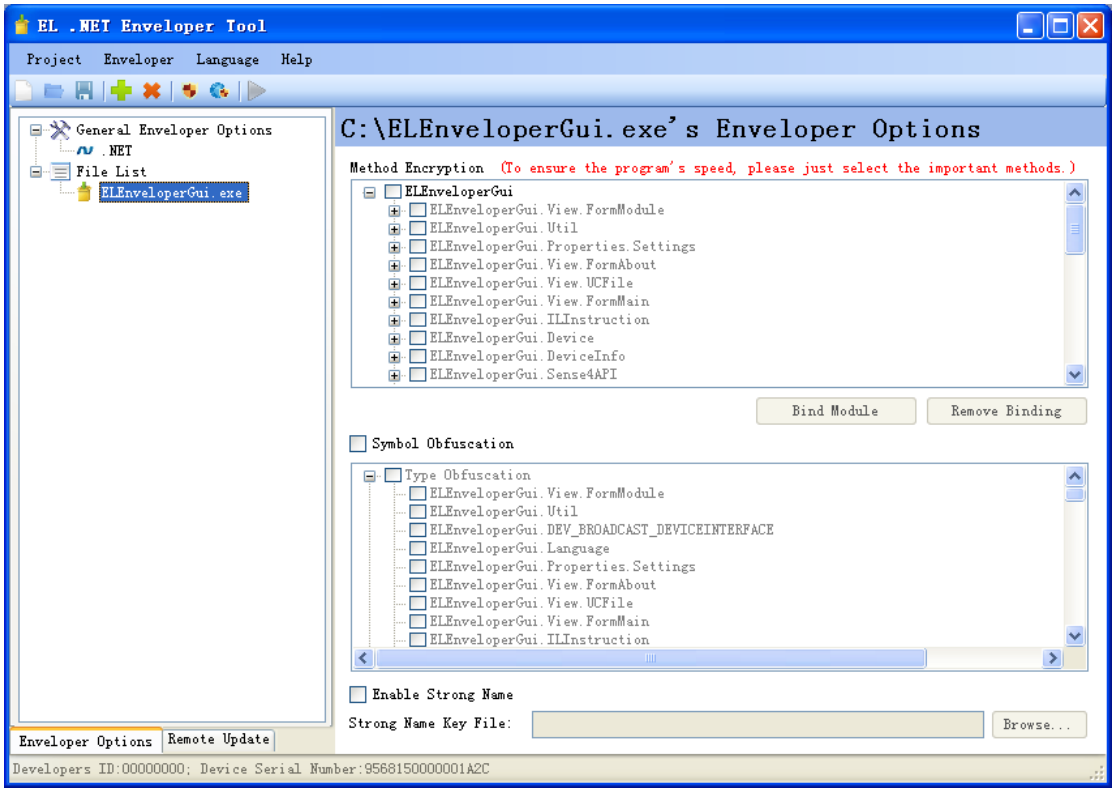
1、Click  in tool bar to envelope all files in File List one-time.

Parameters Settings

1. Envelop Options:

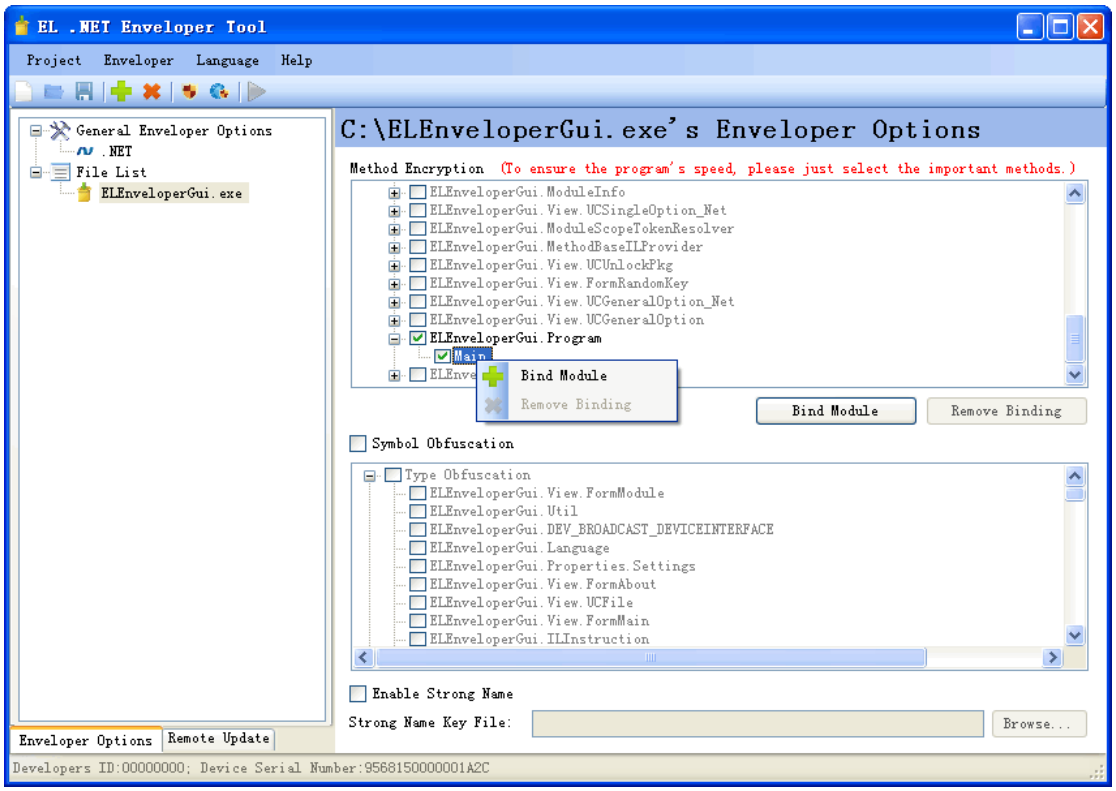
[Mandatory]
Select a file under the *File List*, Envelope Options will show in the details pane (Figure 3-3). Tick methods which you want to encrypt from 'Enveloper Options' list. For EXE assemblies, entrance methods are selected as default. For DLL assemblies, no method is selected as default, encryption methods require to select manually.

> Select too many encryption methods will delay program running speed, it is recommended to encrypt important methods.



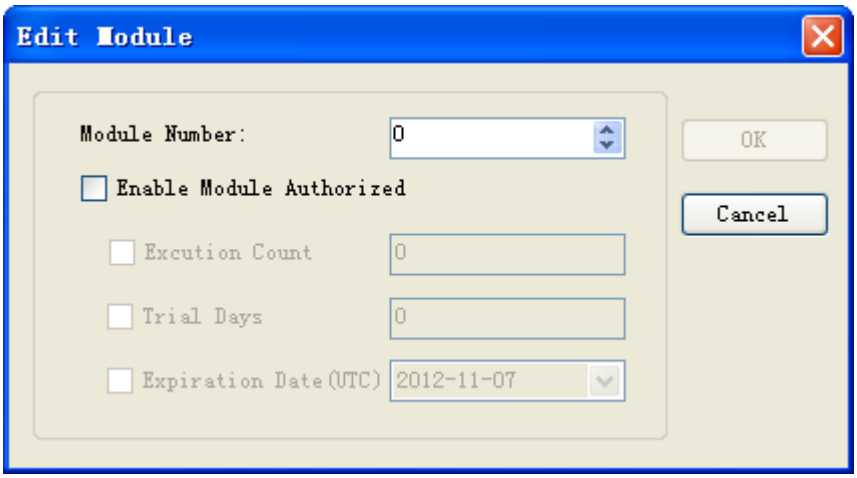
(Figure 3-3)

Encryption method can be bound with specific module, set limitations for using frequency and using times of the method. To bind an encryption method with specified module, select a method and click 'Bind Module' or right click the method, select 'Bind Module' in the context menu (Figure 3-4). To remove the binding, select a method and click 'Remove Binding' button.



(Figure 3-4) Bind Module

Click 'Bind Module' button to bring out the *Edit Module dialog box* (Figure 3-5). Input the Module Number (0~255), tick 'Enable Module Authorized', setup 'Execution Count' (0~65535), 'Trial Days' (0~65535) and 'Expiration Date'. Click 'OK' to finish.



(Figure 3-5) Edit Module

Note:

EL STD, **EL RTC** and **EL RTCC** support 'Execution Count' function; only **EL RTC** and **EL RTCC** supports 'Trail Days' and 'Expiration Date' function.

For **EL RTC** and **EL RTCC** 'Execution Count' and 'Trail Days' can function together, 'Execution Count' and 'Expiration Date' can function together, while 'Trail Days' and

'Expiration Date' CANNOT function together. The function combinations show as following table:

Function Combinations	Execution Count	Trail Days	Expiration Date
Execution Count	-	√	√
Trail Days	√	-	X
Expiration Date	√	X	-

EL RTC/RTCC Functions Combinations

Furthermore, Trail Days and Expiration Date are calculated by EL RTC's inner clock which is adjusted to G.M.T. (Greenwich Mean Time), please take time difference into consideration when setting these two parameters.

2. Symbol Obfuscation:

[Optional]
Applying Symbol Obfuscation to EXE assemblies requires you have a deep understanding at your software. Symbol obfuscation may cause software running failure sometimes, such as the software contains reflection code.



DLL assemblies do NOT support Code Obfuscation.

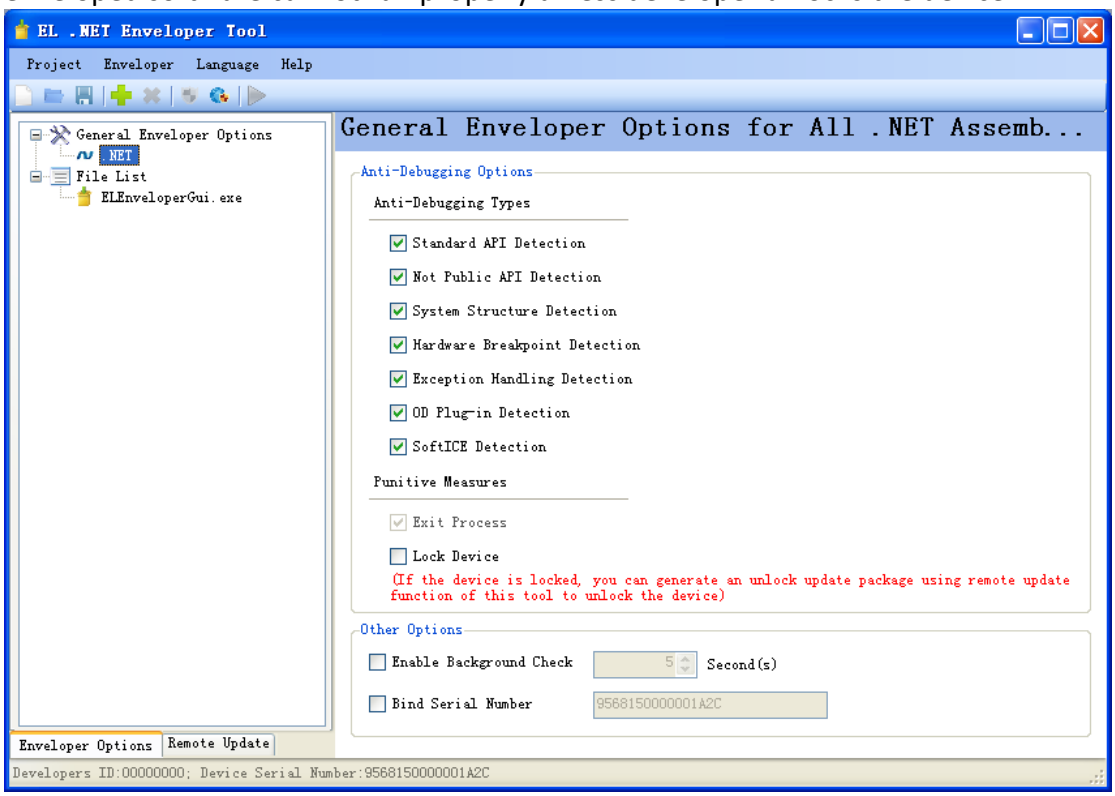
3. Strong-Name Signing

[Optional]
Strong-Name Signing, or strong-naming, gives a software component a globally unique identity that cannot be spoofed by someone else. Strong names are used to guarantee that component dependencies and configuration statements map to exactly the right component and component version.
To give a enveloped software a Strong-Name signature, tick the box before 'Strong-Name Signing' and select the Strong Name Key File which can be generated by *SN.exe* (Microsoft ® .NET Strong Name Utility).

4. Anti-Debugging

[Optional]
Click '' icon under 'General Enveloper Options' in the navigation pane (Figure 3-6), general enveloper options for all .NET assemblies will show in the detail pane. It is recommended that select all Anti-Debugging Methods in the list. Tick the 'Lock

Device' option, device will be locked when debugging enveloped software and the enveloped software cannot run properly unless developer unlocks the device.



(Figure 3-6)

5. Background check

[Optional]
Tick 'Enable Background check' to enable the background device detection function. With this option, background system will detect device regularly to ensure device is plugged in along with software running all time. Time interval for device detection unit per second, available value is an integer from 1 to 9999, the default frequency setting is checking every 5 seconds.

6. Bind Serial Number

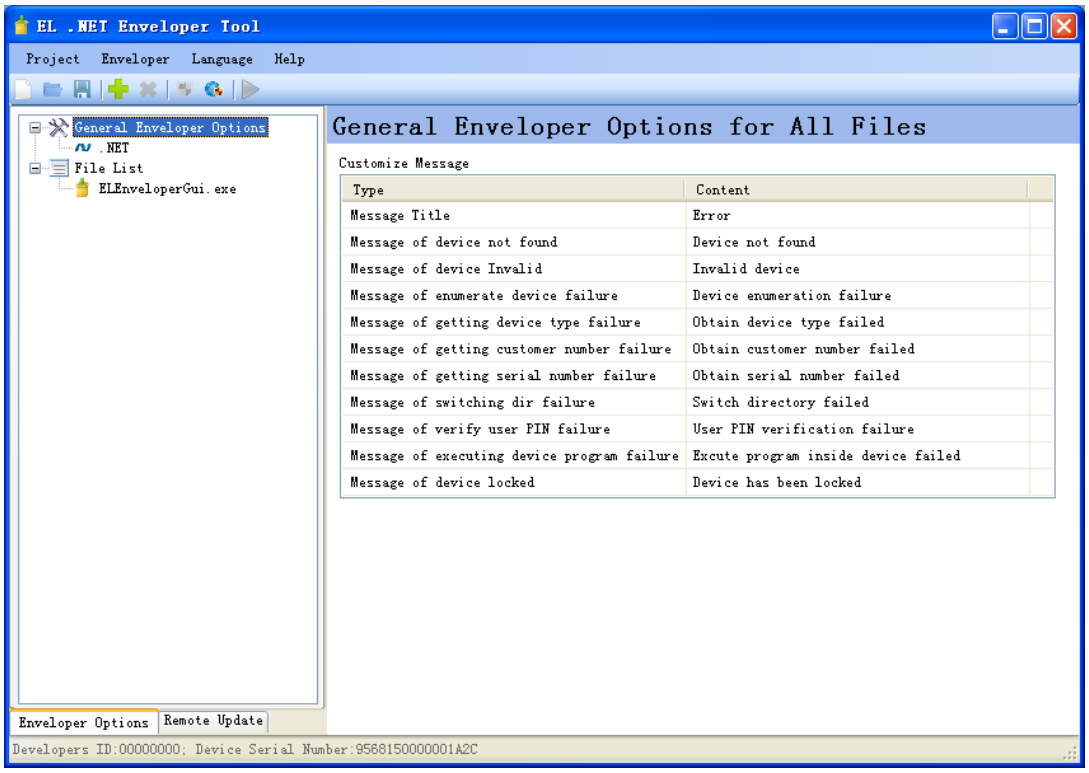
[Optional]
'Bind Serial Number' function will bind enveloped software with the Global Unique Serial Number inside device, which means the enveloped software can run properly only when the specific device which has the bound serial number is plugged in. Tick the box before 'Bind Serial Number' and set the Global Unique Serial Number, Serial Number of current connected device is showing as default.

Example:
Software S has three sets of copies SoftwareA, SoftwareB and SoftwareC, whose Corresponding devices are KeyA, KeyB and KeyC. KeyA, KeyB and KeyC share the same Developer ID but have different Global Unique Serial Numbers.
When use KeyA to envelope executable file (.exe file) of SoftwareA, if:
(i) the enveloped softwareA executable file is bounded with Serial Number of KeyA, enveloped softwareA can only run properly when KeyA is plugged in;

(ii) the enveloped softwareA .exe is *NOT* bounded with any Serial Number, either KeyA, KeyB can allow softwareA .exe run properly.

7. Custom Error Messages

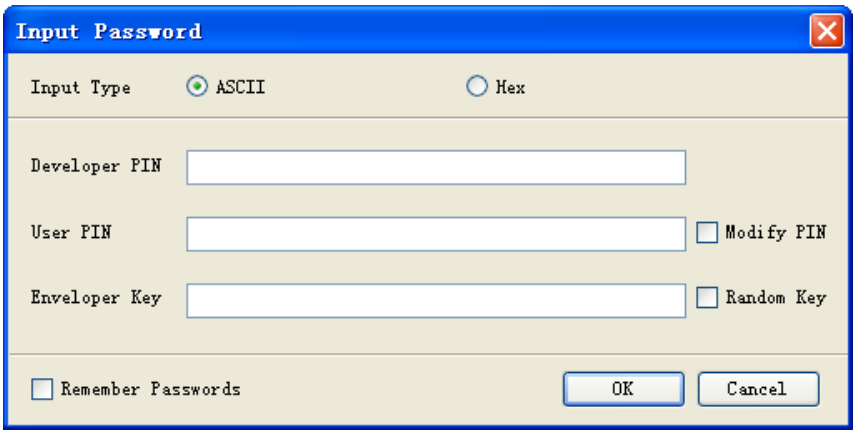
[Optional]
Click 'General Enveloper Options' in navigation pane (Figure 3-7), general enveloper options will show in the detail pane. You can custom notice messages by edit message content in the 'Content' row. Customized error messages will display when EL .NET Enveloper is running.



(Figure 3-7) Customize Message

Input Passwords

When finish parameters settings, click  in tool bar to envelope the current software. To complete software envelope, you will be required to input *Developer PIN*, *User PIN* and *Enveloper Key*. (Figure 3-8)



(Figure 3-8) Input Passwords

First of all, choose an input type for PINs and Password.
Developer PIN will be verified when download program into EL device during envelope process. Since Developer PIN will not be recorded into the enveloped software, you can input the default Developer PIN here.
User PIN is used to call programs inside device when the enveloped software running. To launch program in EL device, user PIN will be recorded into software when envelope, so it is recommended ***NOT*** use default User PIN. If you tick 'Modify User PIN', the old User PIN will be rewritten with the new User PIN.

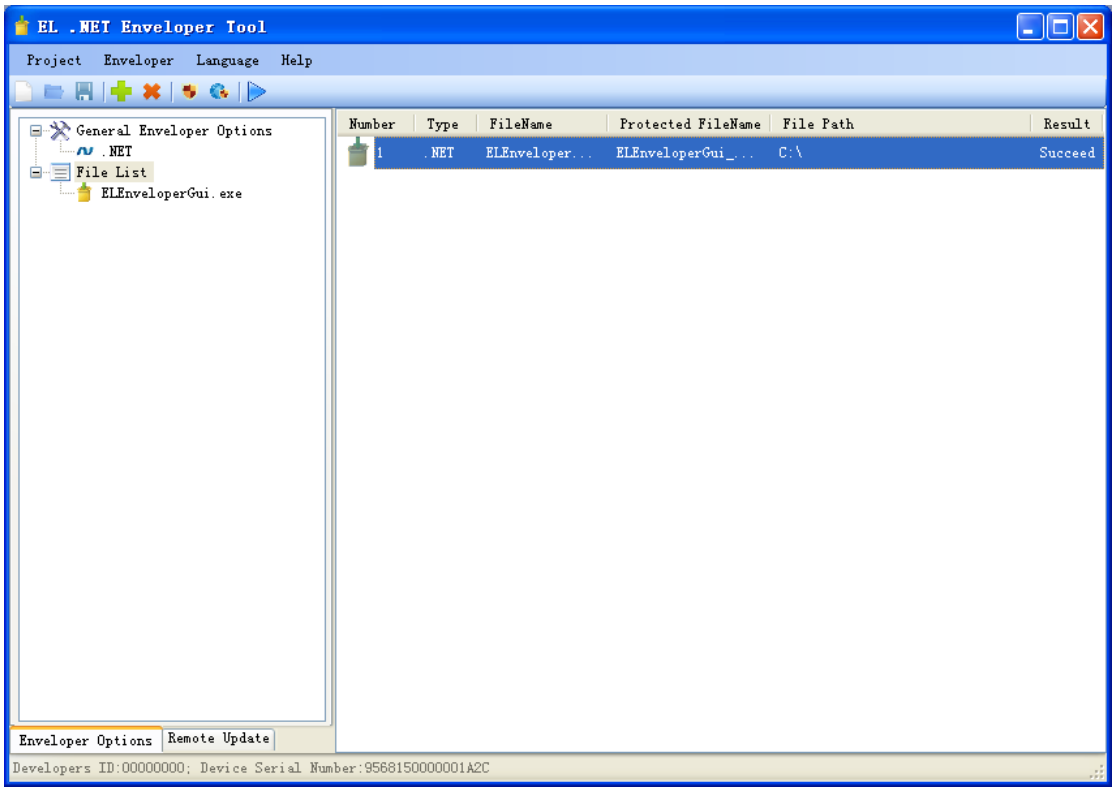


Enveloper Key generates key of encrypt/decrypt codes inside device. Different Enveloper Key generates different encryption codes, so that when two EL devices have different Developer Keys, they cannot be applied to same software, even without Serial Number Binding.

Example:
Envelope Software S with two EL devices, KeyA and KeyB, enveloped software as S_A and S_B, which are not binding Serial Number. Then, S_A can run normaly when either KeyA or KeyB is plugged in, the same as S_B.
However, when S_A and S_B are enveloped with different *Enveloper Key* or the Random Developer Key, S_A run properly only when KeyA is plugged in, S_B can run properly only when KeyB is plugged in.

Envelope Software

After input *Developer PIN*, *User PIN* and *Enveloper Key*, click 'OK' to envelope software. When complete software envelope, click 'File List' in the left navigation pane, enveloped files will show in the right detail pane (Figure 3-9). An executable file, which is the enveloped file, name as '***_shell_timestamp.exe' or '***_shell_timestamp.dll' will be generated under the path of origin file.
Click  in tool bar to test whether the enveloped software runs properly.



(Figure 3-9) Envelope Complete

Execute enveloped software require executable file 'EC01' which located in the root directory of device ('EC01'will be download automatically into current device when envelope software). This ensured that software run properly only when device carried the matched executable file 'EC01' is plugged in.
For developer devices convenience, a '.bin' file will be generated under the same path with the *EL .NET Enveloper* when software envelope complete. Download this '.bin' file into devices and renamed it as 'EC01' to finish setting. However, when software is enveloped more than one time, several enveloped files and '.bin' files will be generated, under this situation, files will match according to timestamp.

Example:
Enveloped file of Executable file 'ABC.exe' named as 'abc_shell_20120807134911.exe' matched bin file named as 'EC01_20120807134911.bin'. In this situation, timestamp '20120807134911' is for matching convincing, *EL .NET Enveloper* will not check or limit files name. Therefore, without binding Serial Number, devices downloaded same .bin file could be applied to the same enveloped software.
Additional, if an 'EC01' file already exists in device, it will be rewritten when envelope. Thence when use one device to envelop software more than one time, the final 'EC01' file in device will be the one have latest timestamp.

Save Project

After envelope finish, enveloped files and settings for all parameters can be saved into project file, when envelope the same file next time, you can open the saved project.

Remote Update

Remote Update function realized module update and unlock device with Module Update Packages and Unlock Update Packages.

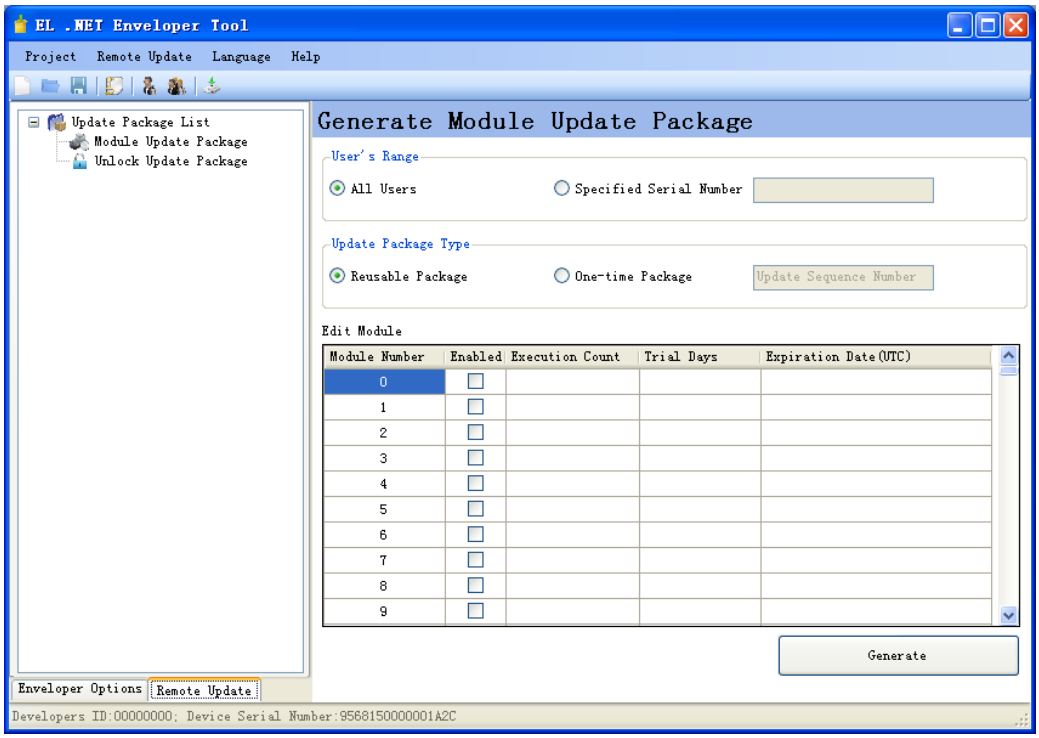
Initialize Device

Remote Update function is based on both *Issuer Device* and *User Device*. Use *Issuer Device* to issue update packages and use *User Device* to update device program with update packages.

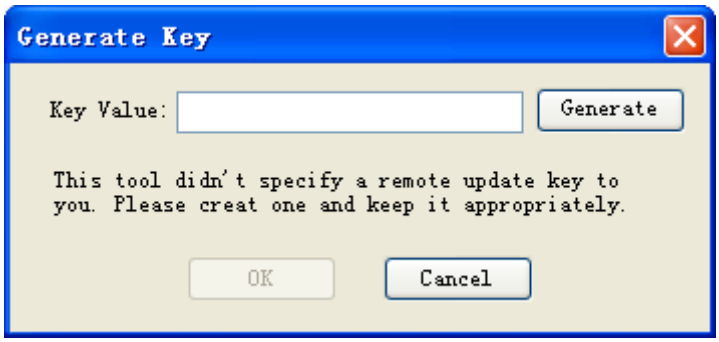
> If you want to use Remote Update function, please create *User Devices* for end users in advance.

1. Create Key Files

EL .NET Enveloper can be used to create Issuer Device, User Device and release files. Click *Remote Update* sheet under navigation pane to switch to remote update control panel (Figure 3-10). A Key Value, which is saved in 'InitTool.ini' under same path with the *Tool*, is required for creating Issuer Device, User Device and release files. You can create the Key Value by clicking  in tool bar (Figure 3-11).



(Figure 3-9) Remote Update

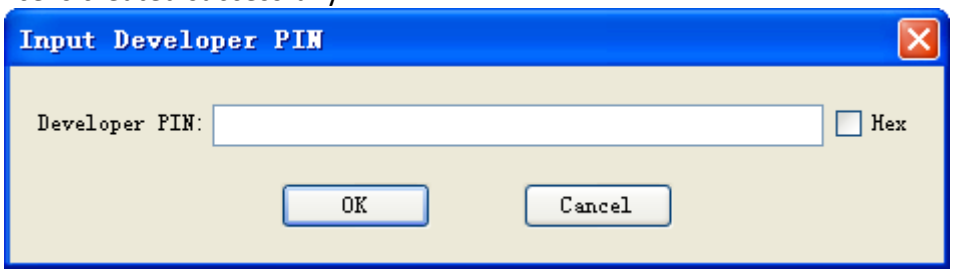


(Figure 3-10) Create Key Value

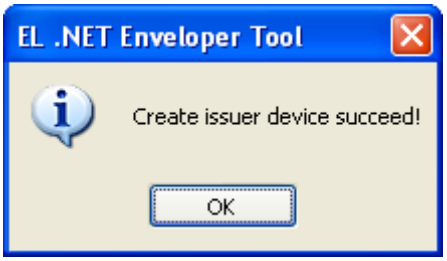
Input Key Value (16 characters) in this box or click 'Generate' to generate a random Key Value. Click 'OK', an 'InitTool.ini' file will be created.

2. Create Issuer Device

Click  in tool bar to create Issuer Device (Figure 3-11), input *Developer PIN.* Tick the 'Hex' to input Developer PIN in hexadecimal, or input in normal string as default. Click 'OK' to create Issuer Device, following dialog box (Figure 3-12) will pop-out after Issuer Device is created successfully.



(Figure 3-11) Input Developer PIN



(Figure 3-12) Issuer Device is created successfully

3. Create User Device

Click  in tool bar to create User Device. Creating process please refer to create Issuer Device.

4. Release File

Click in tool bar to release information files for creating Issuer Device and User Device. These files will be saved to the same directory of the Envelope Tool, default files name are 'FB01.bin', 'FB11.bin' and 'FB12.dat'. Executable file name for Issuer Device and User Device are 'FB01' and 'FB11', the information file name for User Device is 'FB12'. Developer can download these information files to EL, under the root directory to set devices.

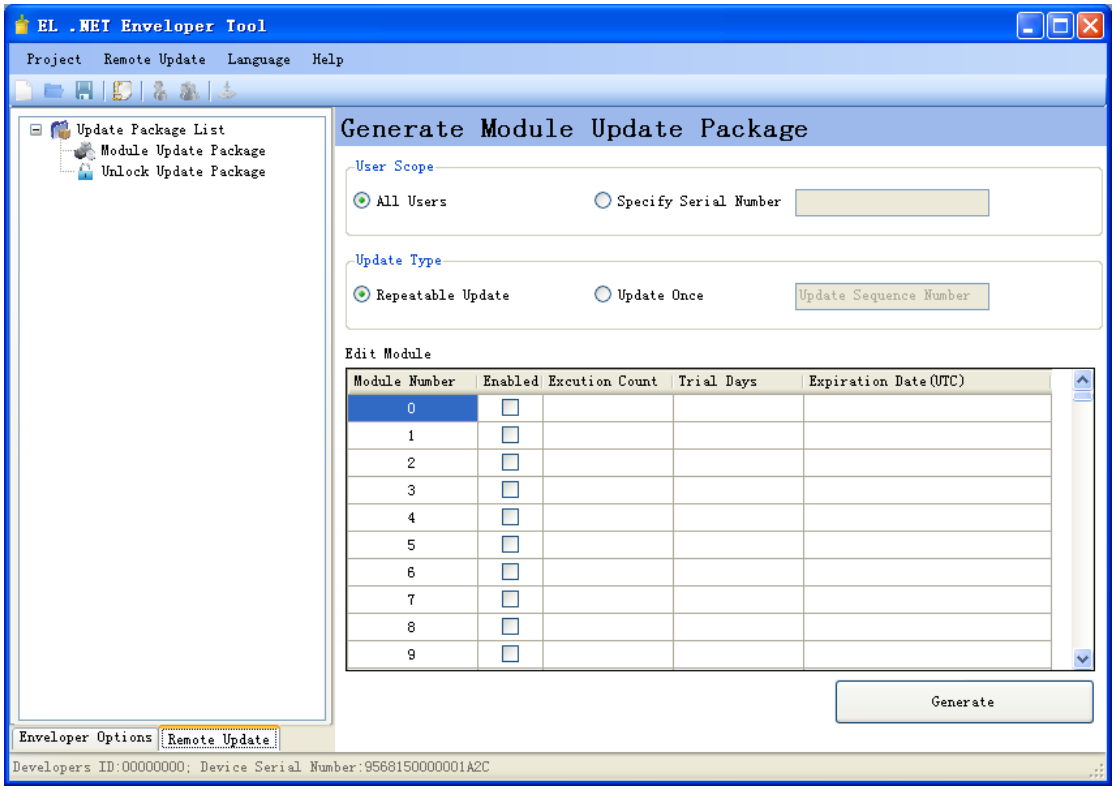
> Please tick '.bin file read-writable in device (It's read-writeable for executable file)' when use **Elite EL Development and Test Tool** download FB11 file.

Issue Update Packages

Update Packages include Module Update Packages, which can be used to update licenses of modules, and Unlock Update Packages, which can be used to reactive the locked devices.

1 Module Update Package

Click 'Module Update Packages' in the navigation pane to show options for generating module update package (Figure 3-13).



(Figure 3-13) Module Update Package

First, set Users' Range. 'All Users' indicates the update package can be used to update all the user's devices, which have same Developer ID with the device issued

the update package. ‘Specify Serial Number’ indicates the update package can only be used to update the device with specified serial number.
Next, set Update Package Type. ‘Reusable Package’ indicates that the Update Package can be reused. ‘One-time Package’ indicates that the Update Package can be used only one time, and if the ‘Sequence Number’ is set Update Package can be used only when the sequence number is greater than the sequence number of previous Update Package.
Then setup, Execution Count, Trail Days and Expiration Date for each module.
After finish above settings, click ‘Generate Update Package’ to complete.



Note:

EL STD, **EL RTC** and **EL RTCC** support ‘Execution Count’ function; only **EL RTC** and **EL RTCC** supports ‘Trail Days’ and ‘Expiration Date’ function.

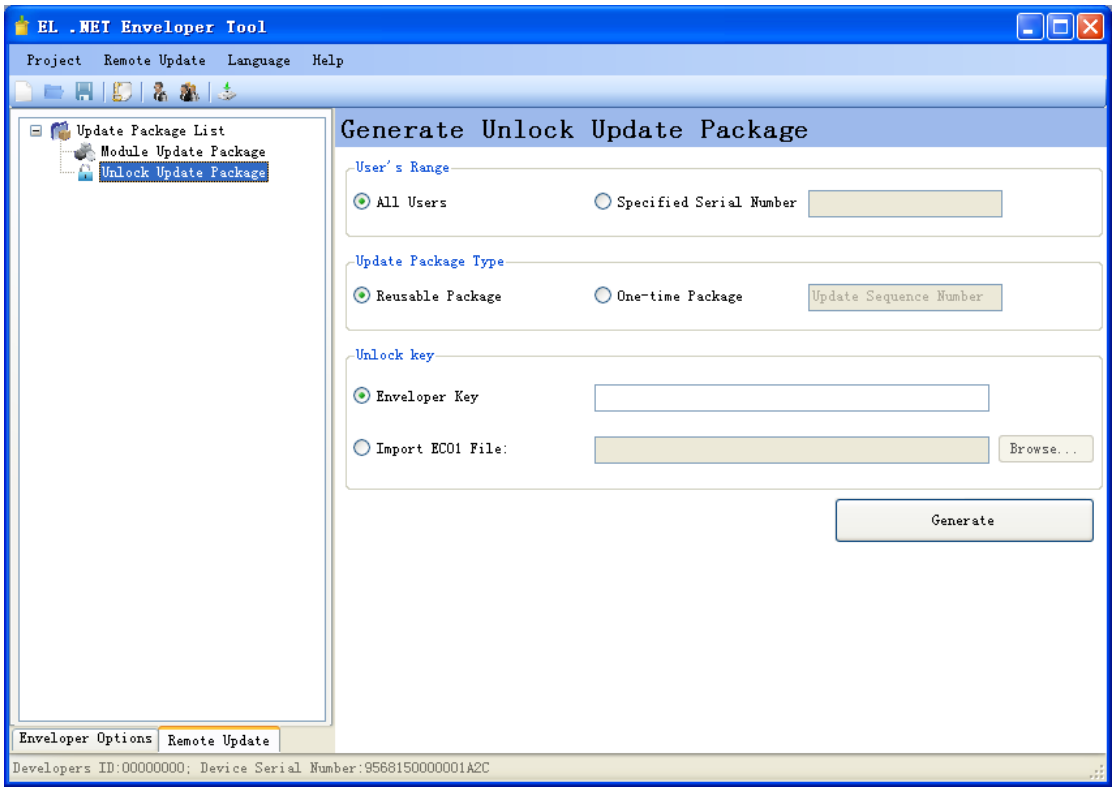
For **EL RTC** and **EL RTCC** ‘Execution Count’ and ‘Trail Days’ can function together, ‘Execution Count’ and ‘Expiration Date’ can function together, while ‘Trail Days’ and ‘Expiration Date’ CANNOT function together. The function combinations show as following table:

Function Combinations	Execution Count	Trail Days	Expiration Date
Execution Count	-	√	√
Trail Days	√	-	X
Expiration Date	√	X	-

EL RTC/RTCC Functions Combinations

2 Unlock Update Package

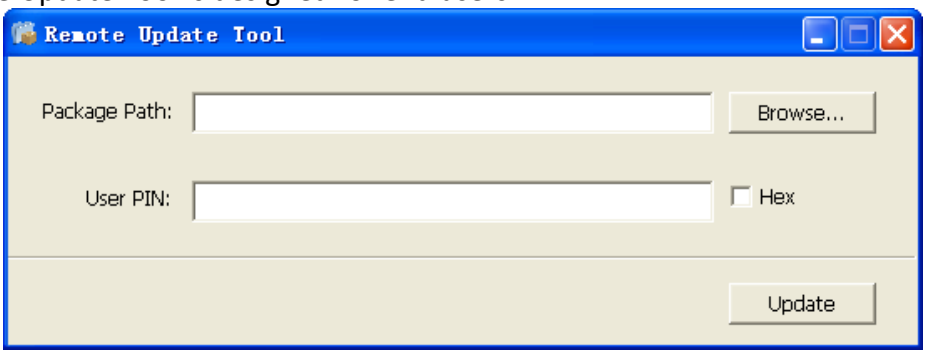
Click ‘Unlock Update Package’ in the navigation pane to show the Generate Unlock Update Package (Figure 3-14).
Settings for Users’ Range and Update Package Type please refer to introduction of ‘**Module Update Package**’ above.
There are two ways to unlock the device: (i) input Key which specified when envelope directly; (ii) import ‘EC01’ file and get the specified Key from ‘EC01’ file.
When complete settings, click ‘Generate Update Package’ to create an Unlock Update Package.



(Figure 3-14) Unlock Update Package

Use Update-Package

Besides of update packages, a Remote Update Tool is also required when end users update their devices. The interface of Remote Update Tool is showing as Figure 3-15. Click 'Browse…' to select an Update Package and input User PIN, tick the 'Hex' to input in Hexadecimal, or input in normal string as default. Click 'Update' to start device update.
Remote Update Tool is designed for end users.



(Figure 3-15) Remote Update Tool

Supplement 4

Following are explanations of error messages and notice messages from EL .NET Enveloper.
Following explanations are using default messages as examples. Messages will display according to your settings when you customized error/notice messages (please refer to page 10, 3.1.2/7. Custom Error Messages).

Device Not Found Error

The current calling device is failed to access to when the enveloped software is running. (Figure 4-1)



(Figure 4-1) Device Not Found

Invalid Device Error

When software enveloper is binding with a Serial Number, a device with different Serial Number is detected while the enveloped software is running; following error message will pop up (Figure 4-2).



(Figure 4-2) Invalid Device

Background Check Errors

When software is enveloped with 'Background check' function, the enveloped software will check device regularly while it's running. If device cannot be detected,

an error message will pop up (Figure 4-3). Click ‘Retry’, the device will be redetected, click ‘Cancel’ the software will exit. You can retry 3 times consecutively, the software will exit when failed to detect the device at the third time.
When software is enveloped with ‘Background Check’ function and ‘Binding Serial Number’ function at the same time and error message ‘Invalid device’ is pop out (Figure 4-4), click ‘Retry’, error message ‘Invalid Device’ will display if the detected device’ Serial Number not match the binding Serial Number.



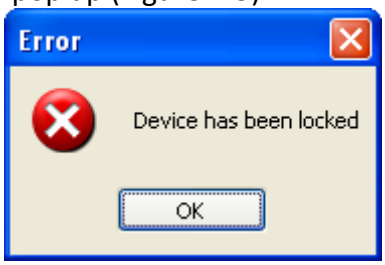
(Figure 4-3) Device Not Found



(Figure 4-4) Invalid Device

Device Locked

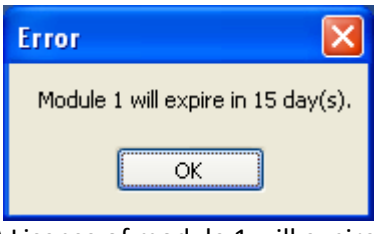
When the device is locked when the enveloped software is being debugged, following error message will pop up (Figure 4-5).



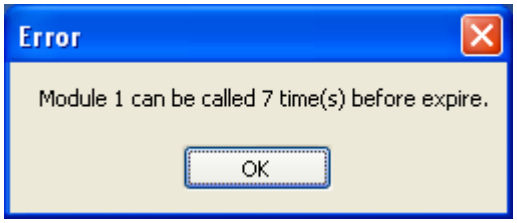
(Figure 4-5) Device Has Been Locked

Module Expiration Notice

When calling an encrypted method which is binding with a module expiring in 15 days or can be called no more than 15 times, a notice message will pup-out as following (Figure 4-6/4-7).



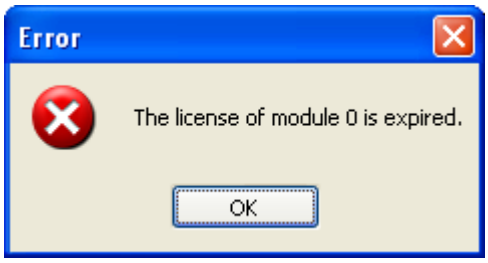
(Figure 4-6) License of module 1 will expire in 15 days.



(Figure 4-7) Module 1 can be called 7 times before license expired.

Module Expired Error

When calling an encrypted method which is binding with an expired module, an error message will pup-out as following (Figure 4-8).



(Figure 4-8) Module License Expired

Executable File Error

Collapse of the enveloped software may caused by executable file ‘EC01’ different from the executable file used to envelope, and led to decryption error. Because of the envelope code cannot judge whether the decryption is correct, this error led to software collapse directly instead of displays an error message.

Appendix 5

Supported Versions for Developer Tools

EL .NET Enveloper supports .NET assemblies of following versions: .NET 2.0, .NET 3.0, .NET 3.5 and .NET 4.0.

Supported Operation Systems

EL .NET Enveloper v1.0.7.0 supports following operation systems: Windows 2003 Server(32bit&64bit), Windows XP Professional (32bit&64bit), Windows Vista(32bit&64bit), Windows 2008(32bit&64bit), Windows7(32bit&64bit), Windows8(32bit&64bit).